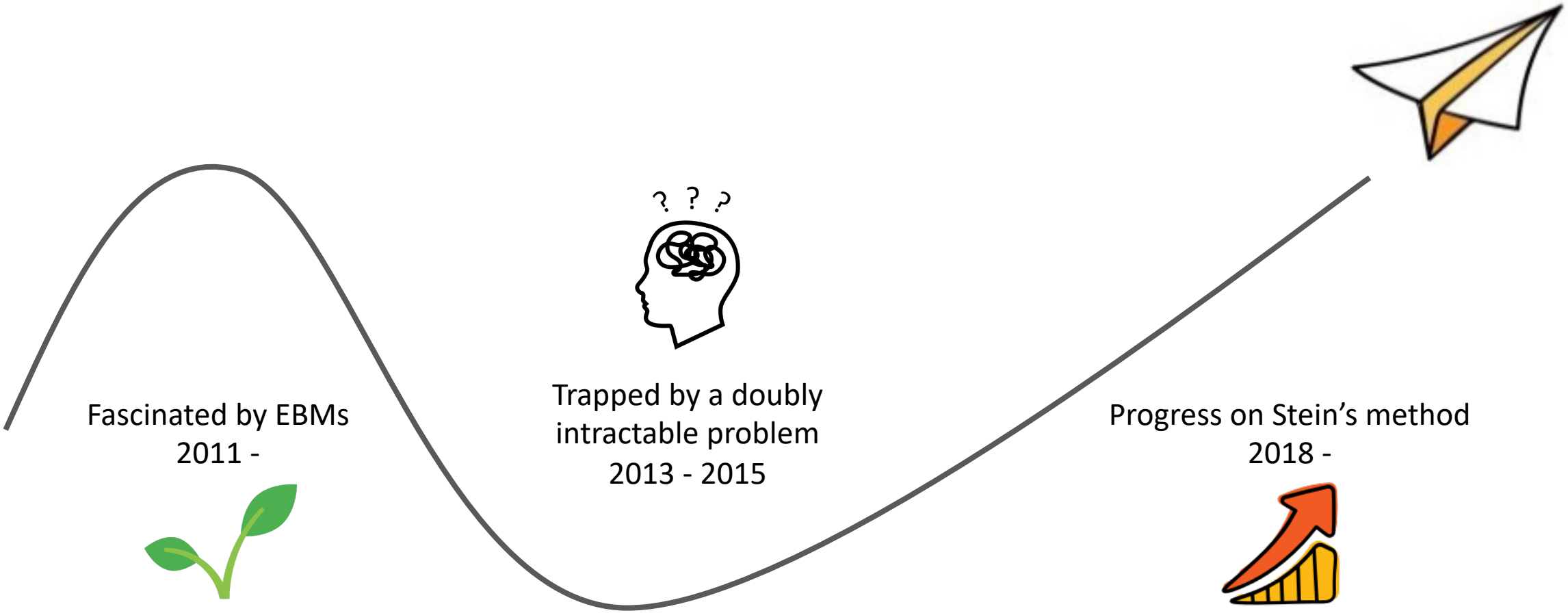
EBM Inference & Learning: A Personal Story

Yingzhen Li
yingzhen.li@imperial.ac.uk

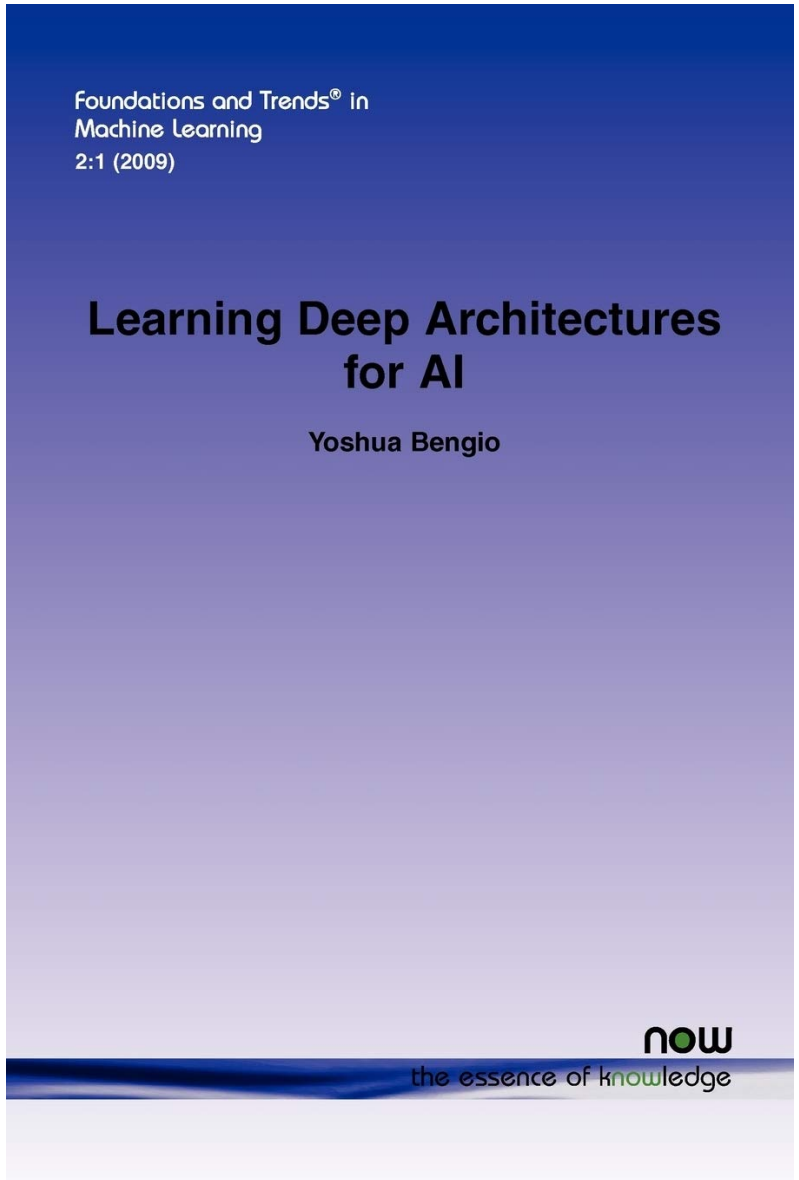
Feat. Wenbo Gong (University of Cambridge)



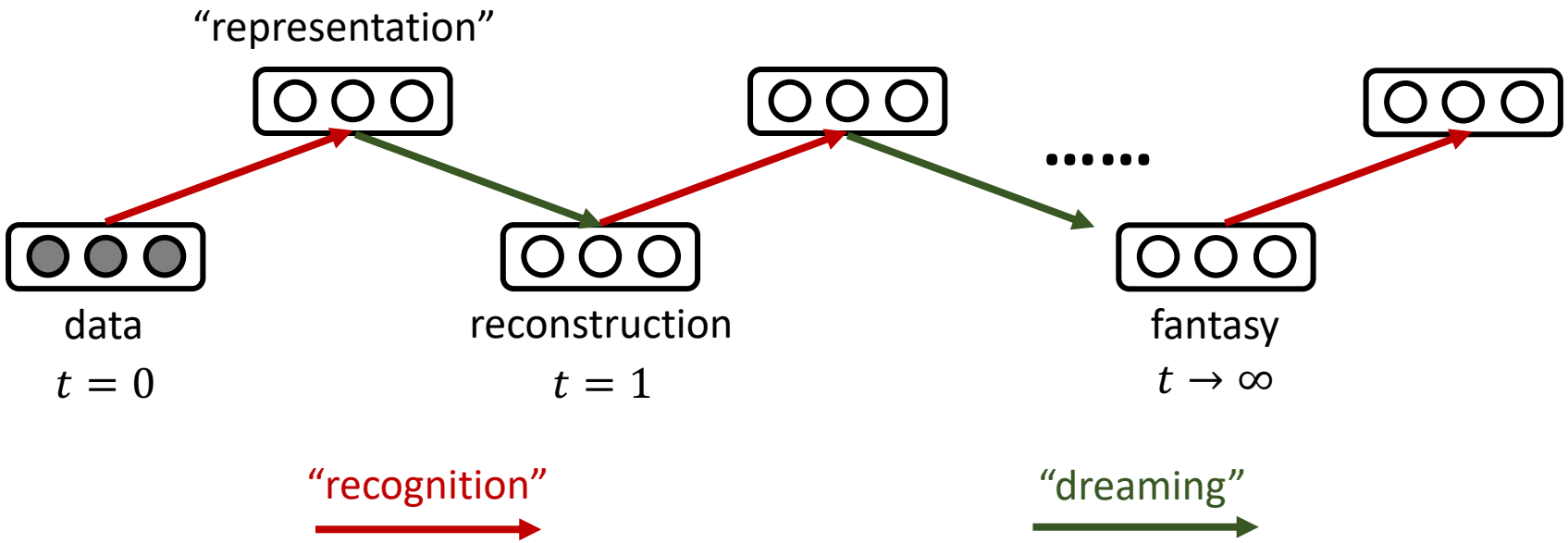
Fascinated by EBMs
2011 -

Trapped by a doubly intractable problem
2013 - 2015

Progress on Stein's method
2018 -


Foundations and Trends® in
Machine Learning
2:1 (2009)
Learning Deep Architectures
for AI
Yoshua Bengio
now
the essence of knowledge

Hinton's "fantasy" metaphor



Hinton. Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation. 2002
Hinton, Osindero and Teh. A Fast Learning Algorithm for Deep Belief Nets. Neural Computation. 2006
Bengio. Learning Deep Architectures for AI. Foundations and Trends in Machine Learning. 2009

Energy-based models

energy function

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp[-E(x;\theta)]$$

normalising constant/
partition function

$$Z(\theta) = \int \exp[-E(x;\theta)]dx$$

Energy-based models

energy function

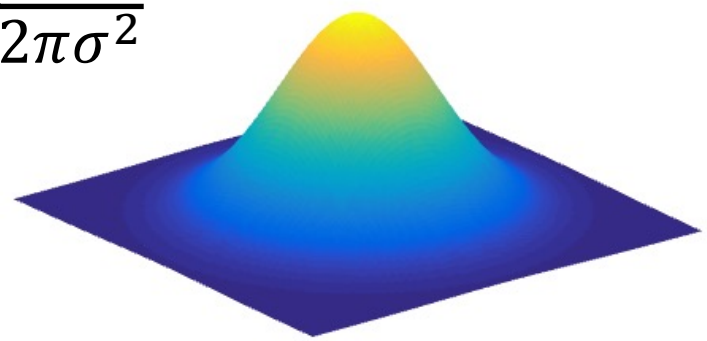
$$p(x|\theta) = \frac{1}{Z(\theta)} \exp[-E(x;\theta)]$$

normalising constant/
partition function

$$Z(\theta) = \int \exp[-E(x;\theta)]dx$$

Examples:

- Gaussian: $E(x;\theta) = \frac{1}{2\sigma^2}(x-\mu)^2, \theta = \{\mu, \sigma^2\}, Z(\theta) = \sqrt{2\pi\sigma^2}$
- Exponential family: $E(x;\theta) = -\Phi(x)^T\theta$



Restricted Boltzmann machines

hidden/latent variable

(negative) energy function

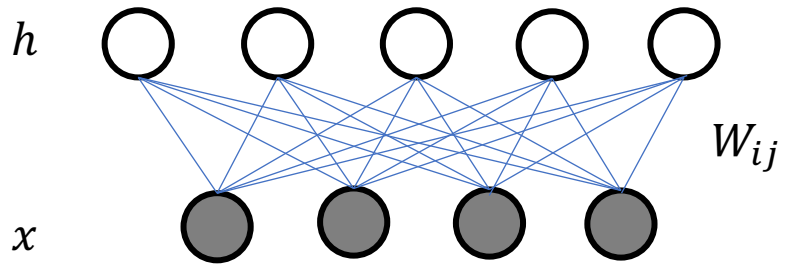
$$p(x, h|\theta) = \frac{1}{Z(\theta)} \exp[b_x^T x + b_h^T h + x^T W h]$$

normalising constant/ partition function

$$Z(\theta) = \sum_{x,h} \exp[b_x^T x + b_h^T h + x^T W h]$$

Examples: Restricted Boltzmann Machine

- $-E(x, h; \theta) = b_x^T x + b_h^T h + x^T W h$
- $\theta = \{W, b_x, b_y\}$
- $x \in \{0, 1\}^{D_x}, h \in \{0, 1\}^{D_h}$



Training EBMs

energy function

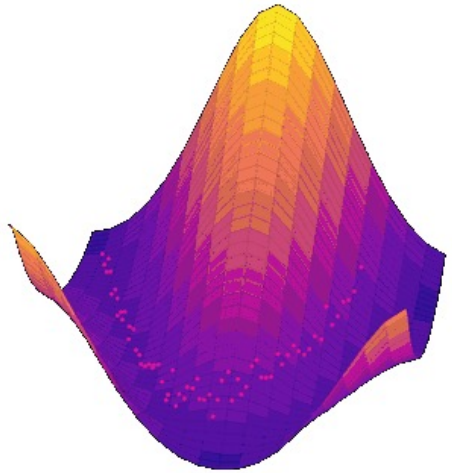
$$p(x|\theta) = \frac{1}{Z(\theta)} \exp[-E(x;\theta)]$$

normalising constant/
partition function

$$Z(\theta) = \int \exp[-E(x;\theta)]dx$$

Fitting an EBM to data:

- Low energy around observed data
- High energy anywhere else

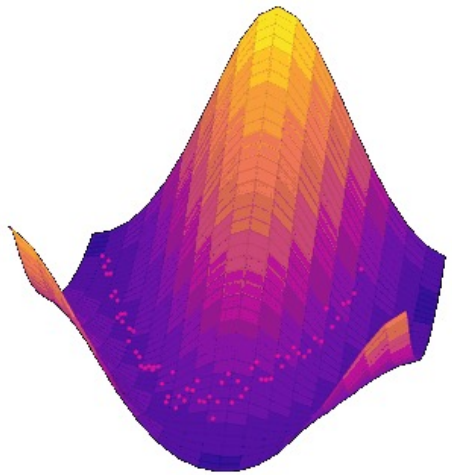


Training EBMs

Maximum Likelihood Estimation (MLE) of θ:

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp[-E(x;\theta)]$$

$$\theta^* = \arg\max_{\theta} E_{p_{data}(x)}[\log p(x|\theta)]$$

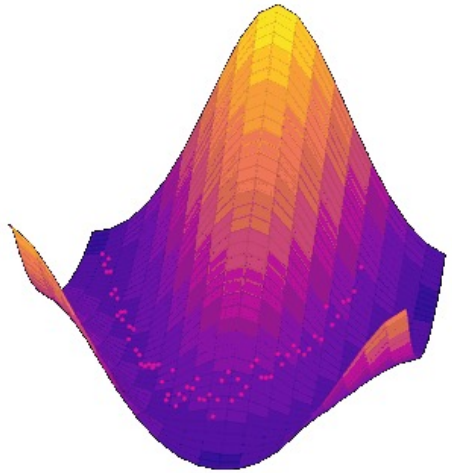


Training EBMs

Maximum Likelihood Estimation (MLE) of θ:

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp[-E(x;\theta)]$$

$$\begin{aligned}\theta^* &= \arg\max_\theta E_{p_{data}(x)}[\log p(x|\theta)] \\ &= \arg\max_\theta E_{p_{data}(x)}[-E(x;\theta) - \log Z(\theta)]\end{aligned}$$



Training EBMs

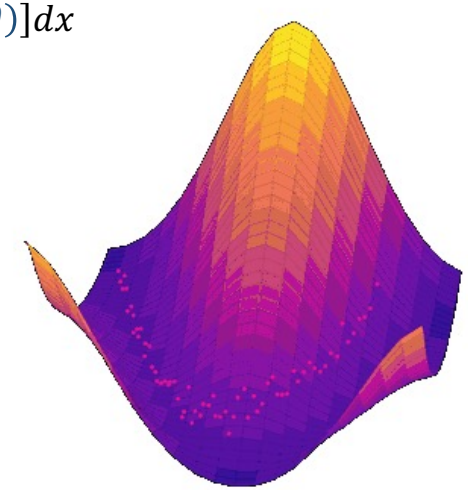
Maximum Likelihood Estimation (MLE) of θ:

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp[-E(x;\theta)]$$

$$\begin{aligned}\theta^* &= \arg\max_\theta E_{p_{data}(x)}[\log p(x|\theta)] \\ &= \arg\max_\theta E_{p_{data}(x)}[-E(x;\theta) - \log Z(\theta)]\end{aligned}$$

$$Z(\theta) = \int \exp[-E(x;\theta)]dx$$

$$-\nabla_\theta E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)}[\nabla_\theta E(x;\theta)] + \nabla_\theta \log Z(\theta)$$



Training EBMs

Maximum Likelihood Estimation (MLE) of θ:

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp[-E(x;\theta)]$$

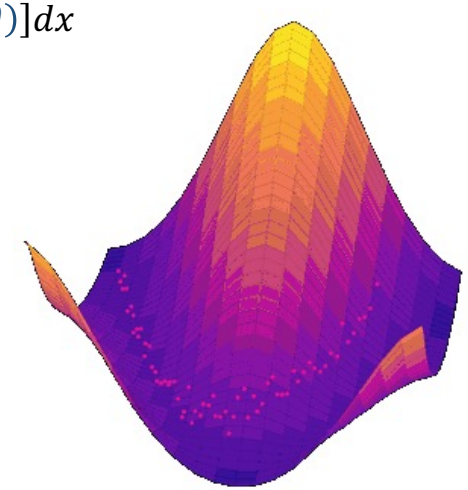
$$\begin{aligned}\theta^* &= \arg\max_\theta E_{p_{data}(x)}[\log p(x|\theta)] \\ &= \arg\max_\theta E_{p_{data}(x)}[-E(x;\theta) - \log Z(\theta)]\end{aligned}$$

$$Z(\theta) = \int \exp[-E(x;\theta)]dx$$

$$\begin{aligned}-\nabla_\theta E_{p_{data}(x)}[\log p(x|\theta)] &= E_{p_{data}(x)}[\nabla_\theta E(x;\theta)] + \nabla_\theta \log Z(\theta) \\ &= E_{p_{data}(x)}[\nabla_\theta E(x;\theta)] - E_{p(x|\theta)}[\nabla_\theta E(x;\theta)]\end{aligned}$$

decrease energy around data

increase energy around samples



Training EBMs

Maximum Likelihood Estimation (MLE) of θ:

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp[-E(x;\theta)]$$

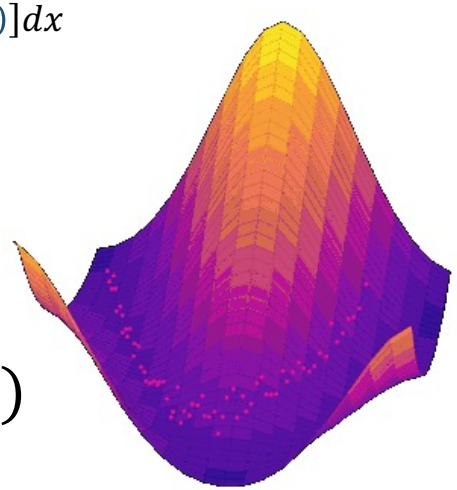
$$\begin{aligned}\theta^* &= \arg\max_\theta E_{p_{data}(x)}[\log p(x|\theta)] \\ &= \arg\max_\theta E_{p_{data}(x)}[-E(x;\theta) - \log Z(\theta)]\end{aligned}$$

$$Z(\theta) = \int \exp[-E(x;\theta)]dx$$

$$\begin{aligned}-\nabla_\theta E_{p_{data}(x)}[\log p(x|\theta)] &= E_{p_{data}(x)}[\nabla_\theta E(x;\theta)] + \nabla_\theta \log Z(\theta) \\ &= E_{p_{data}(x)}[\nabla_\theta E(x;\theta)] - E_{p(x|\theta)}[\nabla_\theta E(x;\theta)]\end{aligned}$$

decrease energy around data

increase energy around samples



Sub-routine: generate "dream/fantasy data" $x \sim p(x|\theta)$

Training RBMs

MLE gradient:

$$-\nabla_\theta \, E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)}[\nabla_\theta E(x;\theta)] - E_{p(x|\theta)}[\nabla_\theta E(x;\theta)]$$

Training RBMs

MLE gradient:

$$-\nabla_\theta \, E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)}[\nabla_\theta E(x;\theta)] - E_{p(x|\theta)}[\nabla_\theta E(x;\theta)]$$

RBM:

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \exp[b_x^T x + b_h^T h + x^T W h] \qquad := -E(x,h;\theta)$$

$$p(x|\theta) = \sum_h p(x,h|\theta) := \frac{1}{Z(\theta)} \exp[-E(x;\theta)] \qquad \Rightarrow -E(x;\theta) = \log \textstyle\sum_h \exp[-E(x,h;\theta)]$$

Training RBMs

MLE gradient:

$$-\nabla_\theta \, E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)}[\nabla_\theta E(x;\theta)] - E_{p(x|\theta)}[\nabla_\theta E(x;\theta)]$$

RBM:

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \exp[b_x^T x + b_h^T h + x^T W h] \qquad := -E(x,h;\theta)$$

$$p(x|\theta) = \sum_h p(x,h|\theta) := \frac{1}{Z(\theta)} \exp[-E(x;\theta)] \quad \Rightarrow -E(x;\theta) = \log \sum_h \exp[-E(x,h;\theta)]$$

$-E(x;\theta)$ is also the log-partition function of $p(h|x,\theta)$:

$$p(h|x,\theta) = \frac{p(x,h|\theta)}{\sum_h p(x,h|\theta)} = \frac{\frac{1}{Z(\theta)} \exp[-E(x,h;\theta)]}{\sum_h \frac{1}{Z(\theta)} \exp[-E(x,h;\theta)]} = \frac{\exp[-E(x,h;\theta)]}{\exp[-E(x;\theta)]} := \frac{1}{Z(x;\theta)} \exp[-E(x,h;\theta)]$$

Training RBMs

MLE gradient:

$$-\nabla_\theta \, E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)}[\nabla_\theta E(x;\theta)] - E_{p(x|\theta)}[\nabla_\theta E(x;\theta)]$$

RBM:

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \exp[b_x^T x + b_h^T h + x^T W h] \qquad := -E(x,h;\theta)$$

$$p(x|\theta) = \sum_h p(x,h|\theta) := \frac{1}{Z(\theta)} \exp[-E(x;\theta)] \quad \Rightarrow -E(x;\theta) = \log \textstyle\sum_h \exp[-E(x,h;\theta)]$$

Gradient of $E(x;\theta)$:

$$\nabla_\theta E(x;\theta) = \frac{1}{\sum_h \exp[-E(x,h;\theta)]} \sum_h \exp[-E(x,h;\theta)] \nabla_\theta E(x,h;\theta) \qquad \sum_h p(h|x,\theta)$$

$$= E_{p(h|x,\theta)}[\nabla_\theta E(x,h;\theta)]$$

Training RBMs

MLE gradient:

$$-\nabla_\theta \, E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)}[\nabla_\theta E(x;\theta)] - E_{p(x|\theta)}[\nabla_\theta E(x;\theta)]$$

RBM:

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \exp[b_x^T x + b_h^T h + x^T W h] \qquad := -E(x,h;\theta)$$

$$p(x|\theta) = \sum_h p(x,h|\theta) := \frac{1}{Z(\theta)} \exp[-E(x;\theta)] \quad \Rightarrow -E(x;\theta) = \log \textstyle\sum_h \exp[-E(x,h;\theta)]$$

Gradient of MLE objective for RBM:

$$-\nabla_\theta \, E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)p(h|x,\theta)}[\nabla_\theta E(x,h;\theta)] - E_{p(x,h|\theta)}[\nabla_\theta E(x,h;\theta)]$$

Sample h conditioned on data

Simulate both $x, h \sim p(x,h|\theta)$

Training RBMs

Gradient of MLE objective for RBM:

$$-\nabla_\theta \, E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)p(h|x,\theta)}[\nabla_\theta E(x,h;\theta)] - E_{p(x,h|\theta)}[\nabla_\theta E(x,h;\theta)]$$

Sample h conditioned on data

Simulate both $x, h \sim p(x,h|\theta)$

- Gibbs sampling: repeat the following 2 steps for $t \geq 1$:
  1. $h_t \sim p(h|x_{t-1},\theta)$
  2. $x_t \sim p(x|h_{t-1},\theta)$

Key advantage of RBMs to enable Gibbs sampling:
$p(h|x,\theta)$ and $p(x|h,\theta)$ are tractable!
(as 1-layer MLPs with sigmoid activations)

Hinton. Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation. 2002

Training RBMs

Gradient of MLE objective for RBM:

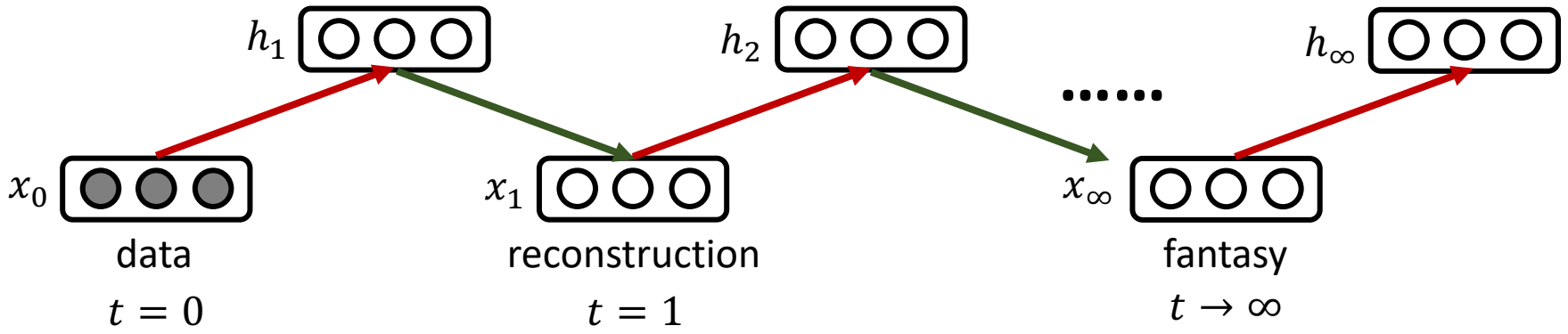
$$-\nabla_\theta \, E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)p(h|x,\theta)}[\nabla_\theta E(x,h;\theta)] - E_{p(x,h|\theta)}[\nabla_\theta E(x,h;\theta)]$$

Sample h conditioned on data

Simulate both $x, h \sim p(x,h|\theta)$

- Gibbs sampling: repeat the following 2 steps for $t \geq 1$:
  1. $h_t \sim p(h|x_{t-1},\theta)$
  2. $x_t \sim p(x|h_{t-1},\theta)$

$$\Rightarrow x_\infty, h_\infty \sim p(x,h|\theta)$$



Training RBMs

Gradient of MLE objective for RBM:

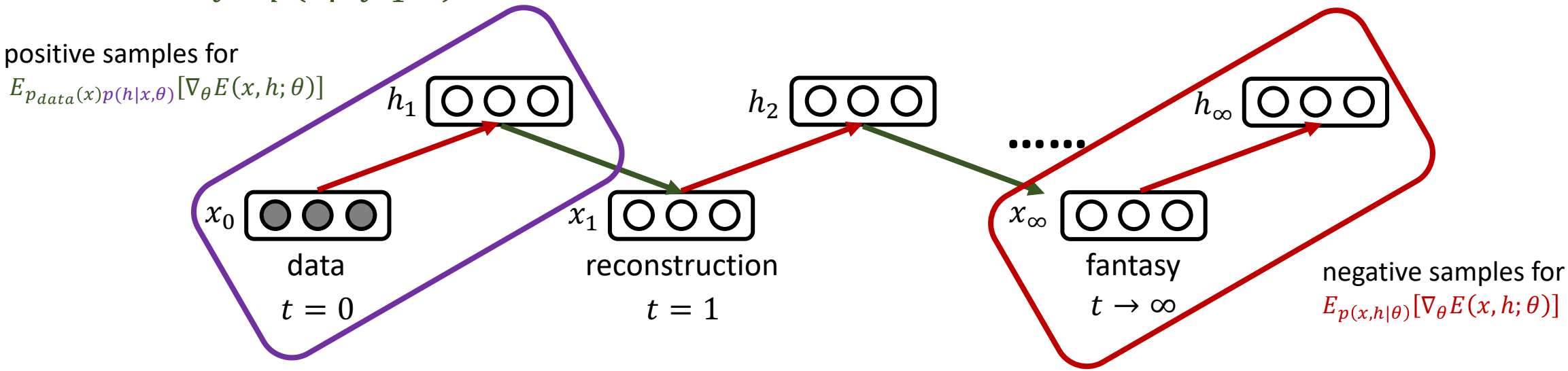
$$-\nabla_\theta E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)p(h|x,\theta)}[\nabla_\theta E(x,h;\theta)] - E_{p(x,h|\theta)}[\nabla_\theta E(x,h;\theta)]$$

Sample h conditioned on data

Simulate both $x, h \sim p(x,h|\theta)$

- Gibbs sampling: repeat the following 2 steps for $t \geq 1$:
  1. $h_t \sim p(h|x_{t-1},\theta)$
  2. $x_t \sim p(x|h_{t-1},\theta)$

$$\Rightarrow x_\infty, h_\infty \sim p(x,h|\theta)$$



Hinton. Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation. 2002

Training RBMs

Gradient of MLE objective for RBM:

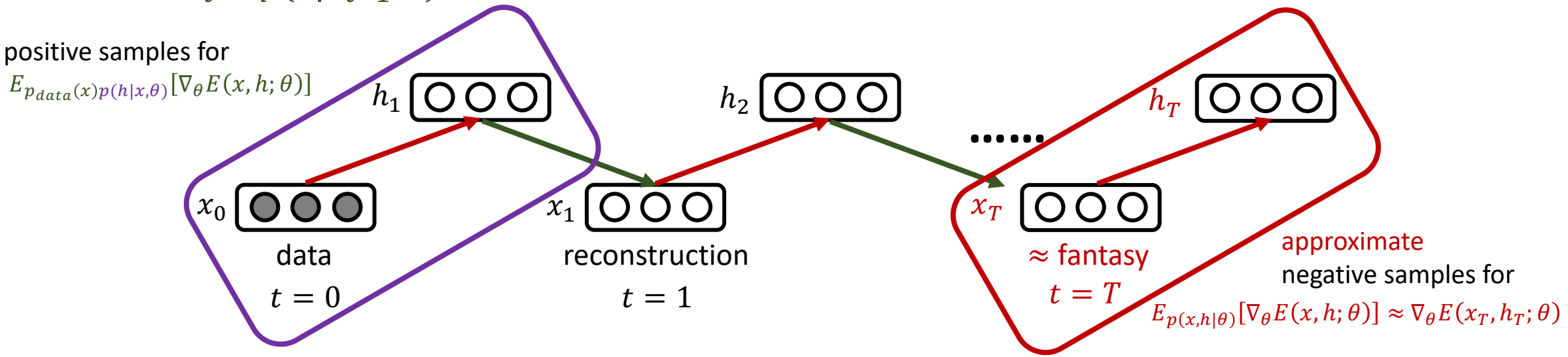
$$-\nabla_\theta E_{p_{data}(x)}[\log p(x|\theta)] = E_{p_{data}(x)p(h|x,\theta)}[\nabla_\theta E(x,h;\theta)] - E_{p(x,h|\theta)}[\nabla_\theta E(x,h;\theta)]$$

Sample h conditioned on data

Simulate both $x, h \sim p(x,h|\theta)$

- Gibbs sampling: repeat the following 2 steps for $t \geq 1$:
  1. $h_t \sim p(h|x_{t-1},\theta)$
  2. $x_t \sim p(x|h_{t-1},\theta)$

Contrastive divergence (CD): truncate the Markov Chain at time T



Applications of RBMs (2006-2012)

- Layer-wised pre-training (deep belief networks)
- Classification (discriminative RBMs)
- Collaborative filtering
- …

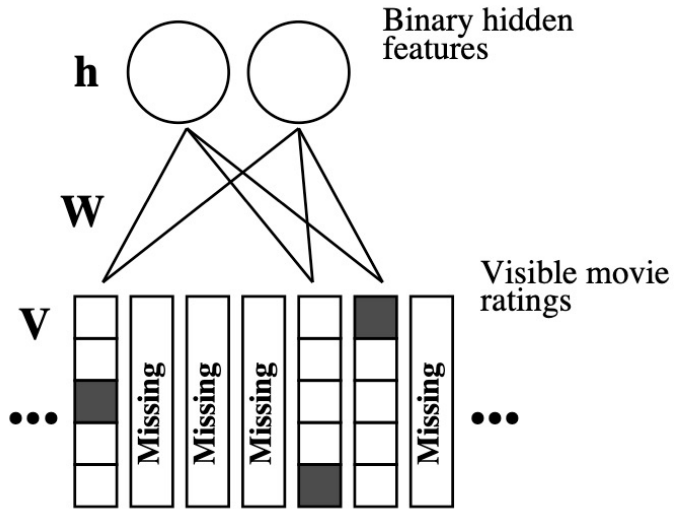


Fig: Salakhutdinov et al. (2007)

Hinton, Osindero and Teh. A Fast Learning Algorithm for Deep Belief Nets. Neural Computation. 2006
Salakhutdinov, Mnih and Hinton. Restricted Boltzmann Machines for Collaborative Filtering. ICML 2007
Larochelle and Bengio. Classification using Discriminative Restricted Boltzmann Machines. ICML 2008

Bayesian inference

Empirical approximation of MLE training:

$$\hat{\theta} = \arg\max_{\theta} \sum_{n=1}^{N} \log p(x_n|\theta)$$

$$x_n \sim p_{data}(x),\ D = \{x_n\}_{n=1}^{N}$$

Severe overfitting when N is small!

Capturing uncertainty in θ given data:

likelihood prior

$$p(\theta|D) = \frac{p(D|\theta)\, p(\theta)}{p(D)}$$

posterior model evidence

$p(D|\theta) = \prod_{n=1}^{N} p(x_n|\theta)$ (iid. assumption)

Bayesian inference

Posterior as an energy-based distribution:

likelihood prior

$$p(\theta|D) = \frac{p(D|\theta)\,p(\theta)}{p(D)}$$

posterior model evidence

iid. assumption:

$p(D|\theta) = \prod_{n=1}^{N} p(x_n|\theta) = \exp[\sum_{n=1}^{N} \log p(x_n|\theta)]$

Bayesian inference

Posterior as an energy-based distribution:

$$p(\theta|D) = \frac{\exp[\sum_n^N \log p(x_n|\theta) + \log p(\theta)]}{p(D)}$$

posterior: $p(\theta|D)$

log-likelihood: $\sum_n^N \log p(x_n|\theta)$

log-prior: $\log p(\theta)$

model evidence: $p(D)$

iid. assumption:

$p(D|\theta) = \prod_{n=1}^N p(x_n|\theta) = \exp[\sum_{n=1}^N \log p(x_n|\theta)]$

Bayesian inference

Posterior as an energy-based distribution:

$-E(\theta; D) \coloneqq$ log-likelihood + log-prior

$$p(\theta|D) = \frac{\exp[\sum_n^N \log p(x_n|\theta) + \log p(\theta)]}{p(D)}$$

posterior; log-likelihood; log-prior

model evidence = partition function of $p(\theta|D)$

$$p(D) = \int p(\theta, D) d\theta = \int \exp[-E(\theta; D)]\, d\theta$$

iid. assumption:

$$p(D|\theta) = \prod_{n=1}^N p(x_n|\theta) = \exp[\sum_{n=1}^N \log p(x_n|\theta)]$$

Bayesian inference

Posterior as an energy-based distribution:

$-E(\theta; D) \coloneqq$ log-likelihood + log-prior

$$p(\theta|D) = \frac{\exp[\sum_n^N \log p(x_n|\theta) + \log p(\theta)]}{p(D)}$$

posterior; log-likelihood; log-prior

model evidence = partition function of $p(\theta|D)$

$$p(D) = \int p(\theta, D) d\theta = \int \exp[-E(\theta; D)]\, d\theta$$

iid. assumption:

$$p(D|\theta) = \prod_{n=1}^N p(x_n|\theta) = \exp[\sum_{n=1}^N \log p(x_n|\theta)]$$

Solving EBM inference ⇔ Solving Bayesian inference!

“Let’s try to do Bayesian inference for the parameters of an RBM”

Rich Turner
(my PhD supervisor)

A doubly intractable problem

The RBM model:

$$\theta = \{W, b_x, b_y\}$$

$$-E(x, h; \theta) = b_x^T x + b_h^T h + x^T W h$$

$$p(x, h|\theta) = \frac{1}{Z(\theta)} \exp[-E(x, h; \theta)]$$

$$p(x|\theta) = \sum_h p(x, h|\theta) := \frac{1}{Z(\theta)} \exp[-E(x; \theta)] \quad \Rightarrow -E(x; \theta) = \log \sum_h \exp[-E(x, h; \theta)]$$

Bayesian inference for θ:

intractable likelihood (mainly due to $Z(\theta)$)

$$p(\theta|D) = \frac{p(D|\theta)\, p(\theta)}{p(D)} = \frac{\frac{1}{Z(\theta)^N} \exp[-\sum_{n=1}^N E(x_n; \theta)]\, p(\theta)}{\int \frac{1}{Z(\theta)^N} \exp[-\sum_{n=1}^N E(x_n; \theta)]\, p(\theta) d\theta}$$

intractable integral for model evidence

A doubly intractable problem

Bayesian inference for θ:

intractable likelihood (mainly due to $Z(\theta)$)

$$p(\theta|D) = \frac{p(D|\theta)\,p(\theta)}{p(D)} = \frac{\frac{1}{Z(\theta)^N}\exp[-\sum_n^N E(x_n;\theta)]\,p(\theta)}{\int \frac{1}{Z(\theta)^N}\exp[-\sum_n^N E(x_n;\theta)]\,p(\theta)d\theta}$$

intractable integral for model evidence

Objectives for approximate Bayesian inference:

- Construct some optimization procedure to fit $q(\theta) \approx p(\theta|D)$
- Cheap approximations for $E(x;\theta)$ and $Z(\theta)$ as sub-routines

Message passing in RBMs

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \exp[b_x^T x + b_h^T h + x^T W h]$$

Factor graph representation:

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{D_x} \phi_i(x_i) \prod_{i,j} \psi_{ij}(x_i, h_j) \prod_{j=1}^{D_h} \phi_j(h_j)$$

$$\phi_i(x_i) = \exp[x_i b_x(i)]$$
$$\phi_j(h_j) = \exp[h_j b_h(j)]$$

singleton factors

$$\psi_{ij}(x_i, h_j) = \exp[x_i W_{ij} h_j]$$

pairwise factors

Kschischang et al. Factor Graphs and the Sum-Product Algorithm. IEEE Transactions on Information Theory. 2001

Message passing in RBMs

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \exp[b_x^T x + b_h^T h + x^T W h]$$

Factor graph representation:

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{D_x} \phi_i(x_i) \prod_{i,j} \psi_{ij}(x_i, h_j) \prod_{j=1}^{D_h} \phi_j(h_j)$$

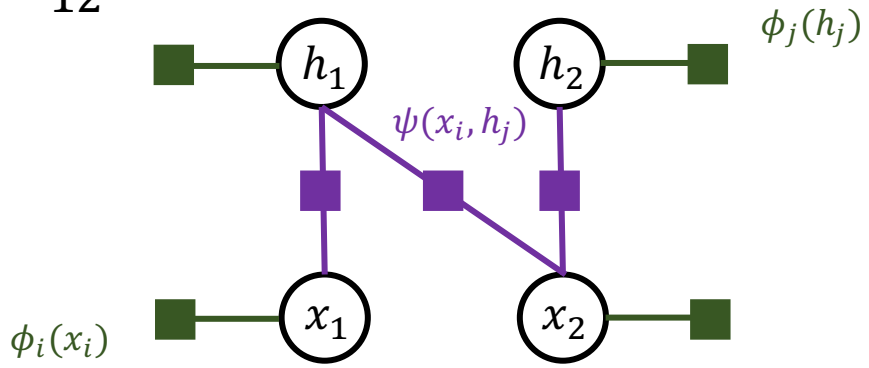
$$\phi_i(x_i) = \exp[x_i b_x(i)]$$
$$\phi_j(h_j) = \exp[h_j b_h(j)]$$

singleton factors

$$\psi_{ij}(x_i, h_j) = \exp[x_i W_{ij} h_j]$$

pairwise factors

Example: $D_x = D_h = 2, W_{12} = 0$

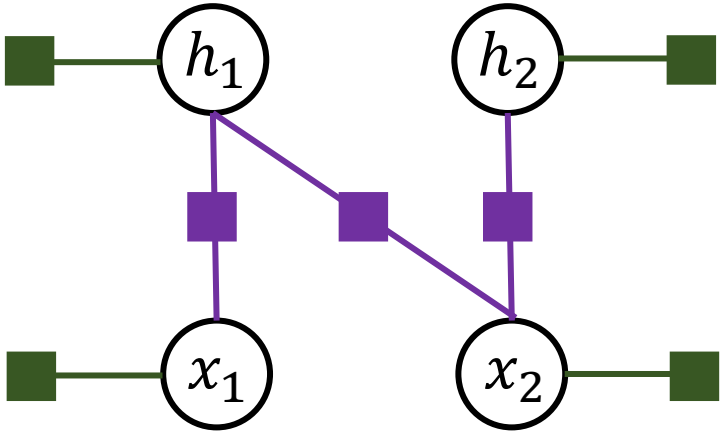


Kschischang et al. Factor Graphs and the Sum-Product Algorithm. IEEE Transactions on Information Theory. 2001

Message passing in RBMs

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{D_x} \phi_i(x_i) \prod_{i,j} \psi_{ij}(x_i,h_j) \prod_{j=1}^{D_h} \phi_j(h_j)$$

Computing (unnormalised) marginals:



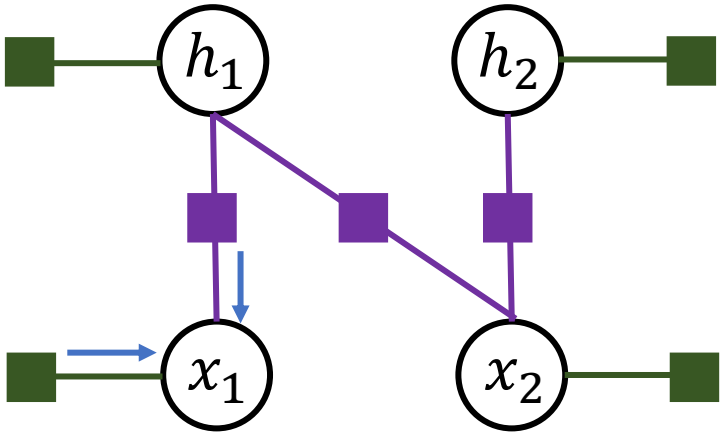
Unnormalised marginal:

$$p^*(x_1|\,\theta) = \sum_{x_2,h_1,h_2} p^*(x_1,x_2,h_1,h_2|\,\theta) = \phi_1(x_1) \sum_{x_2,h_1,h_2} \phi_2(x_2) \prod_{i,j} \psi_{ij}(x_i,h_j) \prod_{j=1}^{2} \phi_j(h_j)$$

Message passing in RBMs

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{D_x} \phi_i(x_i) \prod_{i,j} \psi_{ij}(x_i,h_j) \prod_{j=1}^{D_h} \phi_j(h_j)$$

Computing (unnormalised) marginals:



variable-to-factor message

factor-to-variable message

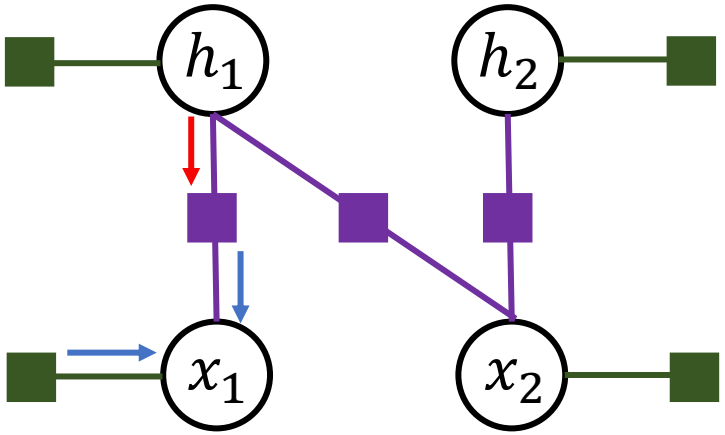
Unnormalised marginal:

$$p^*(x_1|\theta) = \sum_{x_2,h_1,h_2} p^*(x_1,x_2,h_1,h_2|\theta) = \underbrace{m_{\phi_1 \to x_1}(x_1)}_{\phi_1(x_1)} \underbrace{\sum_{x_2,h_1,h_2} \phi_2(x_2) \prod_{i,j} \psi_{ij}(x_i,h_j) \prod_{j=1}^{2} \phi_j(h_j)}_{m_{\psi_{11} \to x_1}(x_1)}$$

Message passing in RBMs

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{D_x} \phi_i(x_i) \prod_{i,j} \psi_{ij}(x_i,h_j) \prod_{j=1}^{D_h} \phi_j(h_j)$$

Computing (unnormalised) marginals:



variable-to-factor message

factor-to-variable message

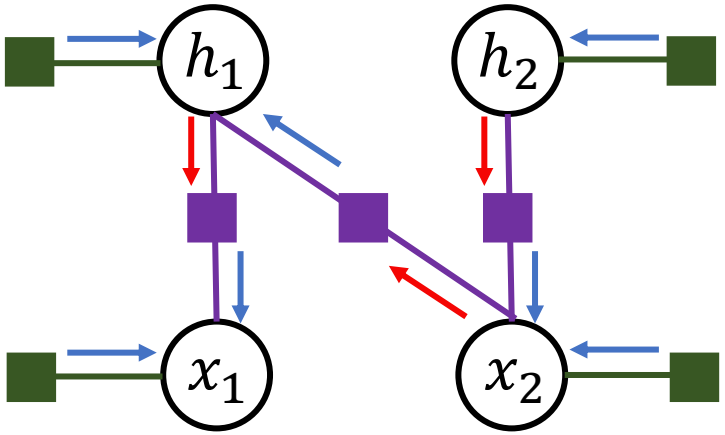
Factor-to-variable message:

$$\underbrace{\sum_{x_2,h_1,h_2} \phi_2(x_2) \prod_{i,j} \psi_{ij}(x_i,h_j) \prod_{j=1}^{2} \phi_j(h_j)}_{m_{\psi_{11}\to x_1}(x_1)} = \sum_{h_1} \psi_{11}(x_1,h_1) \underbrace{\sum_{x_2,h_2} \phi_2(x_2) \prod_{i,j\neq 1,1} \psi_{ij}(x_i,h_j) \prod_{j=1}^{2} \phi_j(h_j)}_{m_{h_1\to\psi_{11}}(h_1)}$$

Message passing in RBMs

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{D_x} \phi_i(x_i) \prod_{i,j} \psi_{ij}(x_i, h_j) \prod_{j=1}^{D_h} \phi_j(h_j)$$

Computing (unnormalised) marginals:



variable-to-factor message

factor-to-variable message

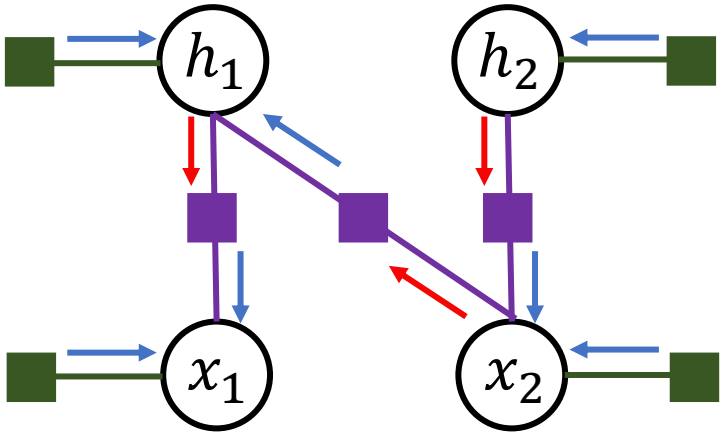
Unnormalised marginal:

$$p^*(x_1|\,\theta) = m_{\phi_1 \to x_1}(x_1) m_{\psi_{11} \to x_1}(x_1) = m_{\phi_1 \to x_1}(x_1) \sum_{h_1} \psi_{11}(x_1, h_1)\, m_{h_1 \to \psi_{11}}(h_1) = \cdots$$

Message passing in RBMs

$$p(x,h|\theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{D_x} \phi_i(x_i) \prod_{i,j} \psi_{ij}(x_i, h_j) \prod_{j=1}^{D_h} \phi_j(h_j)$$

Computing (unnormalised) marginals:



variable-to-factor message

factor-to-variable message

Unnormalised marginal:

$$p^*(x_1|\,\theta) = m_{\phi_1 \to x_1}(x_1) m_{\psi_{11} \to x_1}(x_1) = m_{\phi_1 \to x_1}(x_1) \sum_{h_1} \psi_{11}(x_1, h_1)\, m_{h_1 \to \psi_{11}}(h_1) = \cdots$$

Partition function:

$$Z(\theta) = \sum_{x_1} p^*(x_1|\theta)$$

Message passing in RBMs

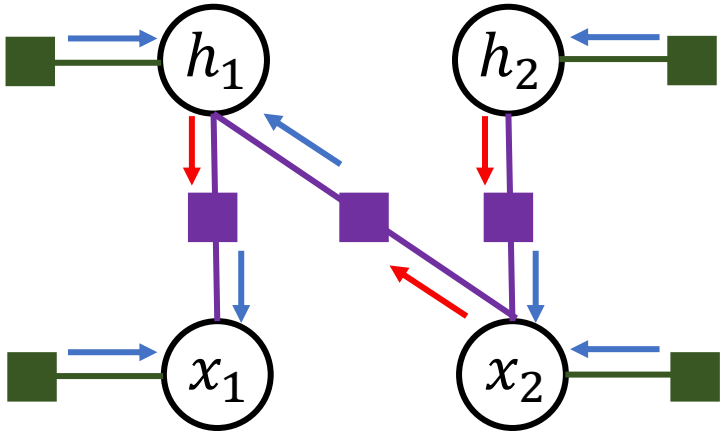
Belief propagation (BP):

Compute the following messages:

- Factor-to-variable message: $m_{f \to x}(x)$
- Variable-to-factor message: $m_{x \to f}(x)$

Approximate the (unnormalized) marginals & partition function using messages :

- $p_{BP}(x_i) \approx p^*(x_i)$
- $Z_{BP}(\theta) \approx Z(\theta)$



Exact when the graph contains no loops (determined by W)

Wainwright and Jordan. Graphical Models, Exponential Families, and Variational Inference. Foundations and Trends in Machine Learning. 2008

Message passing for posterior inference

Bayesian inference for RBM parameters θ:

$$\exp[-E(x_n;\theta)] = p^*(x_n|\theta)$$

$$p(\theta|D) = \frac{p(D|\theta)\,p(\theta)}{p(D)} = \frac{\frac{1}{Z(\theta)^N}\exp[-\sum_{n=1}^{N} E(x_n;\theta)]\,p(\theta)}{\int \frac{1}{Z(\theta)^N}\exp[-\sum_{n=1}^{N} E(x_n;\theta)]\,p(\theta)d\theta}$$

$p(D) = \int \frac{1}{Z(\theta)^N} \prod_{n=1}^{N} p^*(x_n|\theta)\ p(\theta)\ d\theta$ still intractable!

Message passing for posterior inference

Bayesian inference for RBM parameters θ:

$$p(\theta|D) \propto \frac{1}{Z(\theta)^N} \prod_{n=1}^{N} p^*(x_n|\theta)\ p(\theta)$$

$p(D) = \int \frac{1}{Z(\theta)^N} \prod_{n=1}^{N} p^*(x_n|\theta)\ p(\theta)\, d\theta$ still intractable!

Proposed solution: Expectation propagation (EP)

$$p(\theta|D) \approx q(\theta) \propto \frac{1}{\tilde{Z}(\theta)^N} \prod_{n=1}^{N} \tilde{f}_n(\theta)\, p(\theta)$$

Fit $\tilde{f}_n(\theta) \approx p^*(x_n|\theta)$, $\tilde{Z}(\theta) \approx Z(\theta)$, such that the normalising constant of q is tractable

Message passing for Bayesian RBMs

- Goal of EP approximations:

$$p(\theta|D) \propto \frac{1}{Z(\theta)^N} \prod_{n=1}^{N} p^*(x_n|\theta)\, p(\theta) \qquad \approx \qquad q(\theta) \propto \frac{1}{\tilde{Z}(\theta)^N} \prod_{n=1}^{N} \tilde{f}_n(\theta)\, p(\theta)$$

Idea 1: independent approximations:

$$\tilde{f}_n(\theta) \approx p^*(x_n|\theta), \;\; \tilde{Z}(\theta) \approx Z(\theta)$$

Problem: approximation error magnified by multiplication!

Message passing for Bayesian RBMs

- Goal of EP approximations:

$$p(\theta|D) \propto \frac{1}{Z(\theta)^N} \prod_{n=1}^{N} p^*(x_n|\theta)\, p(\theta) \qquad \approx \qquad q(\theta) \propto \frac{1}{\tilde{Z}(\theta)^N} \prod_{n=1}^{N} \tilde{f}_n(\theta)\, p(\theta)$$

Idea 2 (EP): update $\tilde{f}_i(\theta)$ such that

$$\frac{1}{\tilde{Z}(\theta)^N} \prod_{n \neq i} \tilde{f}_n(\theta)\, p(\theta) \times p^*(x_i|\theta) \approx \frac{1}{\tilde{Z}(\theta)^N} \prod_{n \neq i} \tilde{f}_n(\theta)\, p(\theta) \times \tilde{f}_i(\theta)$$

Message passing for Bayesian RBMs

- Goal of EP approximations:

$$p(\theta|D) \propto \frac{1}{Z(\theta)^N} \prod_{n=1}^{N} p^*(x_n|\theta)\, p(\theta) \qquad \approx \qquad q(\theta) \propto \frac{1}{\tilde{Z}(\theta)^N} \prod_{n=1}^{N} \tilde{f}_n(\theta)\, p(\theta)$$

Idea 2 (EP): update $\tilde{f}_i(\theta)$ such that

$$\frac{1}{\tilde{Z}(\theta)^N} \prod_{n \neq i} \tilde{f}_n(\theta)\, p(\theta) \times p^*(x_i|\theta) \approx \frac{1}{\tilde{Z}(\theta)^N} \prod_{n \neq i} \tilde{f}_n(\theta)\, p(\theta) \times \tilde{f}_i(\theta)$$

cavity distribution (treat as constant) $\propto q(\theta)/\tilde{f}_i(\theta)$ $\propto q(\theta)/\tilde{f}_i(\theta)$

Take errors in other terms in considerations

Message passing for Bayesian RBMs

- Goal of EP approximations:

$$p(\theta|D) \propto \frac{1}{Z(\theta)^N} \prod_{n=1}^{N} p^*(x_n|\theta)\, p(\theta) \quad \approx \quad q(\theta) \propto \frac{1}{\tilde{Z}(\theta)^N} \prod_{n=1}^{N} \tilde{f}_n(\theta)\, p(\theta)$$

Idea 2 (EP): update $\tilde{f}_i(\theta)$ such that

$$\frac{1}{\tilde{Z}(\theta)^N} \prod_{n \neq i} \tilde{f}_n(\theta)\, p(\theta) \times p^*(x_i|\theta) \approx \frac{1}{\tilde{Z}(\theta)^N} \prod_{n \neq i} \tilde{f}_n(\theta)\, p(\theta) \times \tilde{f}_i(\theta)$$

Intractable for RBM
⇒ approximated by BP:

$$p_{BP}(x_i|\theta) \approx p^*(x_i|\theta)$$

Message passing for Bayesian RBMs

- Goal of EP approximations:

$$p(\theta|D) \propto \frac{1}{Z(\theta)^N} \prod_{n=1}^{N} p^*(x_n|\theta)\, p(\theta) \qquad \approx \qquad q(\theta) \propto \frac{1}{\tilde{Z}(\theta)^N} \prod_{n=1}^{N} \tilde{f}_n(\theta)\, p(\theta)$$

Idea 2 (EP): update $\tilde{f}_i(\theta)$ such that

$$\frac{1}{\tilde{Z}(\theta)^N} \prod_{n \neq i} \tilde{f}_n(\theta)\, p(\theta) \times p_{BP}(x_i|\theta) \approx \frac{1}{\tilde{Z}(\theta)^N} \prod_{n \neq i} \tilde{f}_n(\theta)\, p(\theta) \times \tilde{f}_i(\theta)$$

Message passing for Bayesian RBMs

- Goal of EP approximations:

$$p(\theta|D) \propto \frac{1}{Z(\theta)^N} \prod_{n=1}^{N} p^*(x_n|\theta)\, p(\theta) \qquad \approx \qquad q(\theta) \propto \frac{1}{\tilde{Z}(\theta)^N} \prod_{n=1}^{N} \tilde{f}_n(\theta)\, p(\theta)$$

Idea 2 (EP): update $\tilde{Z}(\theta)$ such that

$$\frac{1}{\tilde{Z}(\theta)^{N+1}} \prod_{n} \tilde{f}_n(\theta)\, p(\theta) \times Z_{BP}(\theta) \approx \frac{1}{\tilde{Z}(\theta)^{N+1}} \prod_{n} \tilde{f}_n(\theta)\, p(\theta) \times \tilde{Z}(\theta)$$

cavity distribution (treat as constant) $\propto q(\theta)/\tilde{Z}(\theta)$ $\propto q(\theta)/\tilde{Z}(\theta)$

Message passing for Bayesian RBMs

- Goal of EP approximations:

$$p(\theta|D) \propto \frac{1}{Z(\theta)^N} \prod_{n=1}^{N} p^*(x_n|\theta)\, p(\theta) \qquad \approx \qquad q(\theta) \propto \frac{1}{\tilde{Z}(\theta)^N} \prod_{n=1}^{N} \tilde{f}_n(\theta)\, p(\theta)$$

Algorithm summary:

1. Pick a factor $f(\theta) \in \{\tilde{f}_1(\theta), \ldots, \tilde{f}_N(\theta), \tilde{Z}(\theta)\}$
2. Update the factor $f(\theta)$ using EP update rules
3. Repeat the above until convergence

$$\frac{1}{\tilde{Z}(\theta)^N} \prod_{n \neq i} \tilde{f}_n(\theta)\, p(\theta) \times p_{BP}(x_i|\theta) \approx \frac{1}{\tilde{Z}(\theta)^N} \prod_{n \neq i} \tilde{f}_n(\theta)\, p(\theta) \times \tilde{f}_i(\theta)$$

$$\frac{1}{\tilde{Z}(\theta)^{N+1}} \prod_{n} \tilde{f}_n(\theta)\, p(\theta) \times Z_{BP}(\theta) \approx \frac{1}{\tilde{Z}(\theta)^{N+1}} \prod_{n} \tilde{f}_n(\theta)\, p(\theta) \times \tilde{Z}(\theta)$$

Synthetic experiments

Settings:

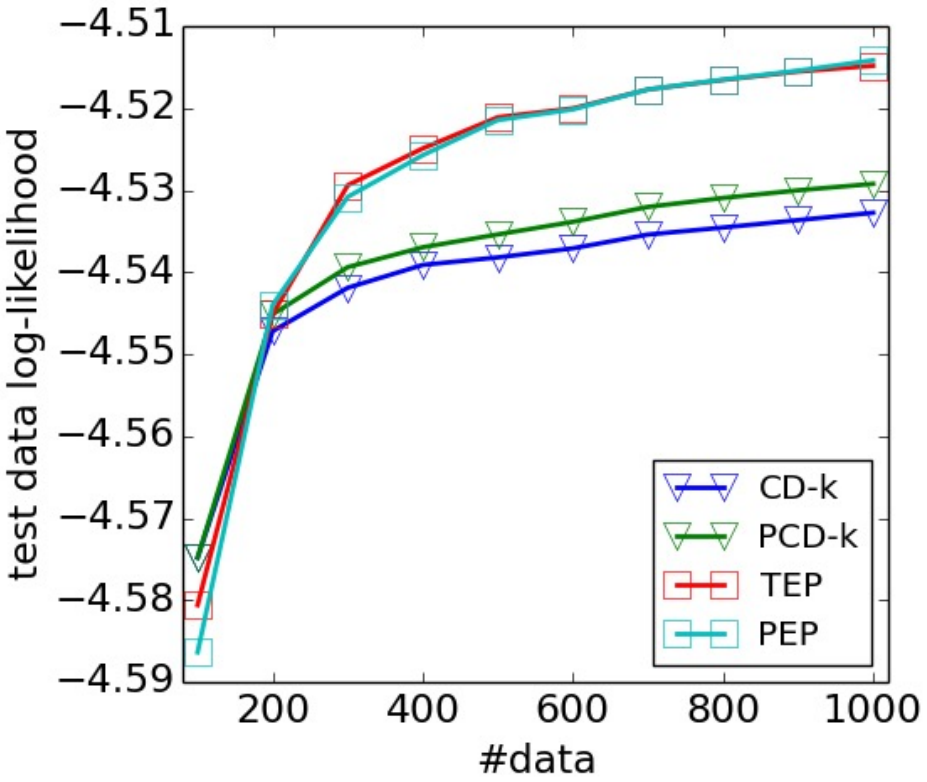
- $x_n \sim p_{data}(x)$ as an RBM (no model mismatch)
- $D_x = 10, D_h = 5$

Methods in comparison:

- Approx. MLE (CD, PCD)
- Bayesian inference (TEP, PEP)

Evaluation metric: test-LL for $x^* \sim p_{data}(x)$

- For MLE: test-LL = $\log p(x^*|\hat{\theta})$
- For EP methods: test-LL $\approx \log \int p(x^*|\theta)q(\theta)d\theta$



Small-scale real-world dataset

FAQ Dataset: predict if an input sentence is a question or an answer

- Unbalanced dataset (#answers > #questions)
- Unsupervised pre-training of RBM (either by approx. MLE or EP methods)
- Build a 2-layer MLP classifier on top of RBM hidden variables

		"answer"	"question"	
	training schemes	error (average)	error (pos. data)	error (neg. data)	TPR/FPR
Approx. MLE	CD-1	8.67 ± 1.73	4.14 ± 1.16	67.89 ± 21.83	1.57 ± 0.50
	CD-10	8.39 ± 1.95	3.84 ± 0.93	67.72 ± 23.20	1.60 ± 0.54
EP-based (Bayesian)	BPE-ADF	9.11 ± 0.80	5.19 ± 1.88	61.16 ± 13.57	1.64 ± 0.41
	BI-ADF	8.90 ± 0.75	4.98 ± 1.80	60.71 ± 14.83	1.67 ± 0.45
	BPE-PEP	8.01 ± 2.28	4.05 ± 0.84	60.29 ± 20.62	1.77 ± 0.53
	BI-PEP	7.92 ± 2.31	3.96 ± 0.80	60.19 ± 21.42	1.78 ± 0.53
	BPE-TEP	12.58 ± 1.40	7.03 ± 2.97	87.58 ± 8.20	1.06 ± 0.06
	BI-TEP	9.37 ± 1.56	4.55 ± 0.76	73.18 ± 18.79	1.39 ± 0.34

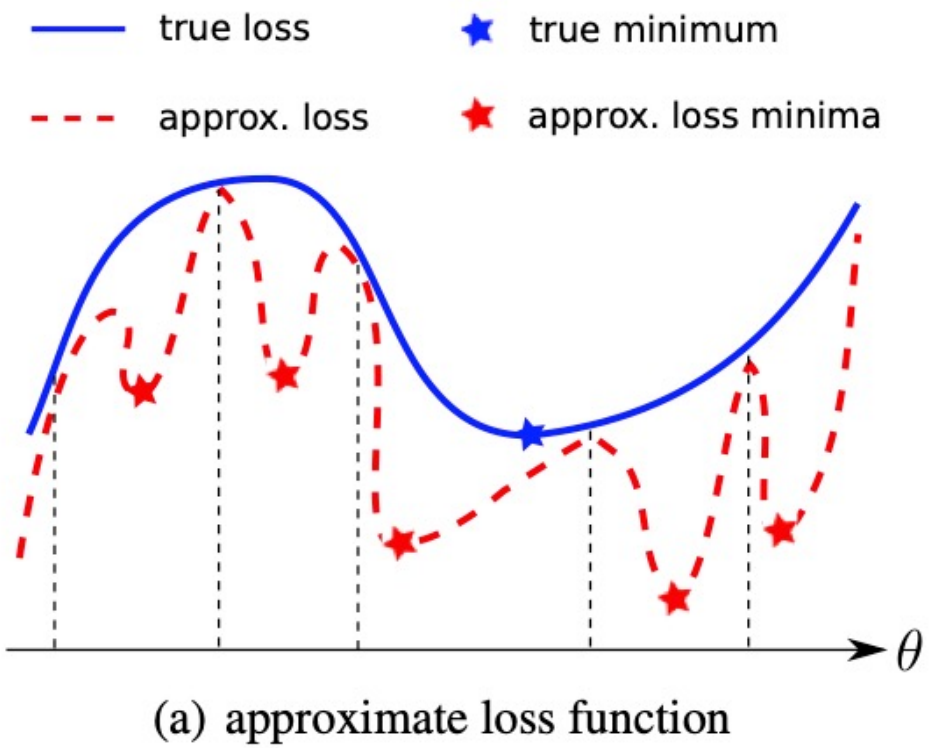
Lessons learned

- Memory consumptions way too high
  - Unnecessary repeat of approximating factors ⇐ addressed by stochastic EP
- Empirical tricks matter
  - Schedule of message passing (RBM graph is cyclic)
  - Also controlling numerical error is key
- Good inference ⇒ Good downstream performance?
  - A general question for approximate inference & model selection

Li, Hernandez-Lobato and Turner. Stochastic Expectation Propagation. NIPS 2015


YOU'RE SUPPOSED TO BE WRITING ME.
I KNOW!
DRAFT
LET ME JUST TAKE CARE OF THIS ONE THING FIRST.
FINE.
DRAFT
DRAFT
JORGE CHAM © 2015
I'M STILL HERE.
I KNOW!
DRAFT
WWW.PHDCOMICS.COM

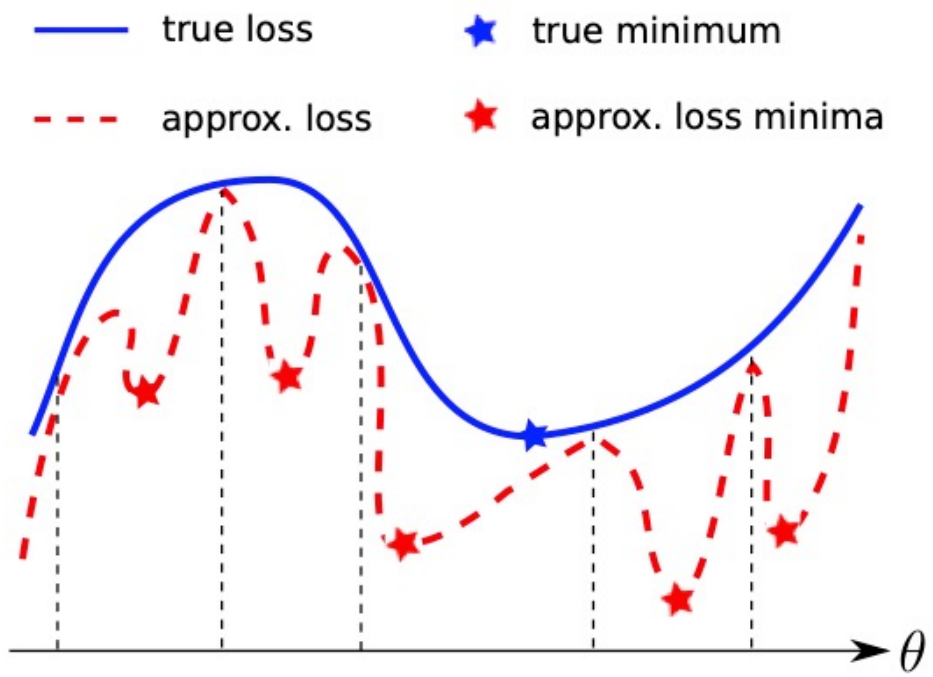
Learning models with intractable density



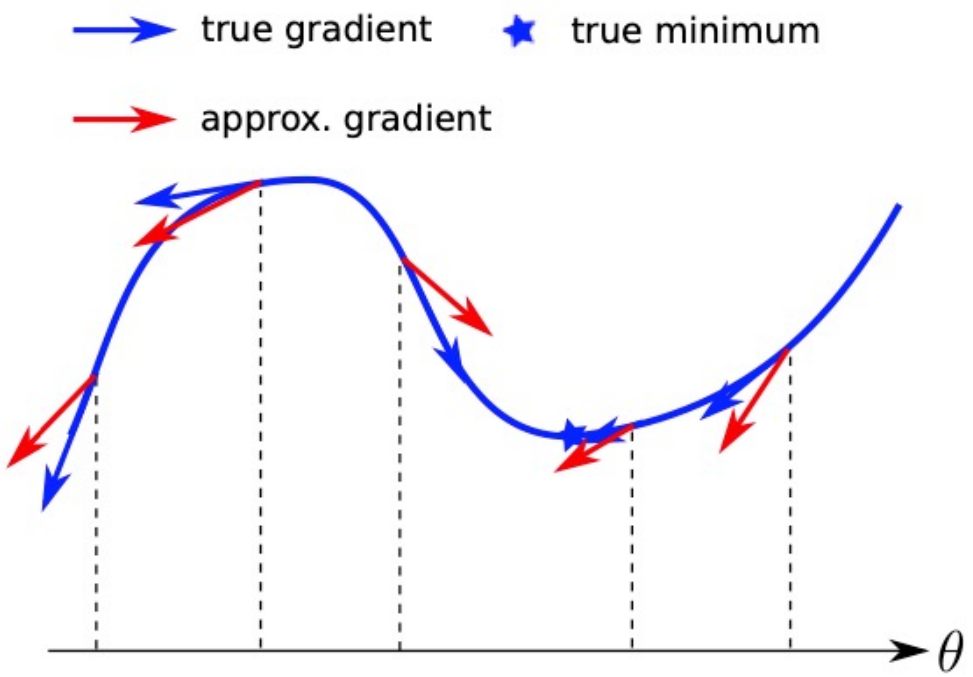
(a) approximate loss function

Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018

Learning models with intractable density



(a) approximate loss function



(b) approximate gradients

Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018

Stein's method & score matching

- Stein discrepancy:

$$S(q,p) = \sup_{f \in F} E_{q(x)}[s_p(x)^\top f(x) + \nabla^\top f(x)],$$

$$s_p(x) = \nabla_x \log p(x)$$

- Estimation requires:
  - Samples $x \sim q(x)$
  - Score function $s_p(x)$ evaluated at x

Gorham and Mackey. Measuring Sample Quality with Stein's Method. NIPS 2015
Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018

Stein's method & score matching

- Stein discrepancy:

$$S(q,p) = \sup_{f \in F} E_{q(x)}[s_p(x)^\top f(x) + \nabla^\top f(x)],$$

$$s_p(x) = \nabla_x \log p(x)$$

- Estimation requires:
  - Samples $x \sim q(x)$ ← enables estimation of $\nabla_x \log q(x)$ & posterior approximations
  - Score function $s_p(x)$ evaluated at x ← enables learning of $p(x)$ as an EBM

$$p(x) = \frac{1}{Z_\theta} \exp[-E_\theta(x)]$$

$$\nabla_x \log p(x) = -\nabla_x E_\theta(x) - \boxed{\nabla_x \log Z_\theta}$$

$$= 0$$

Gorham and Mackey. Measuring Sample Quality with Stein's Method. NIPS 2015
Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018

Stein's method & score matching

- $q = p \Leftrightarrow \nabla_x \log q(x) = \nabla_x \log p(x) \ \forall x \in R^d$
- Fisher divergence: a way to measure score difference:

$$F(p,q) = \frac{1}{2} E_q[\|\nabla_x \log q(x) - \nabla_x \log p(x)\|_2^2]$$

Unknown score (only have $x \sim q(x)$)

Hyvarinen. Estimation of Non-Normalized Statistical Models by Score Matching. JMLR 2005

Stein's method & score matching

- $q = p \Leftrightarrow \nabla_x \log q(x) = \nabla_x \log p(x) \ \forall x \in R^d$
- Fisher divergence: a way to measure score difference:

$$F(p,q) = \frac{1}{2} E_q[\|\nabla_x \log q(x) - \nabla_x \log p(x)\|_2^2]$$

- Ways to (approximately) compute/minimise $F(p,q)$:
  - Alternative formula based on integration by parts
  - Denoising auto-encoder with infinitesimal injected data noise

Stein discrepancy

Hyvarinen. Estimation of Non-Normalized Statistical Models by Score Matching. JMLR 2005
Vincent. A connection between score matching and denoising autoencoders. Neural Computation. 2011

Stein's method & score matching

- Stein discrepancy:

$$S(q,p) = \sup_{f \in F} E_{q(x)}[s_p(x)^\top f(x) + \nabla^\top f(x)],$$

$$s_p(x) = \nabla_x \log p(x)$$

- Selecting test functions $f \in F$:
  - As L_2 integrable functions:
    - $f^*(x) \propto s_p(x) - s_q(x)$ (intractable)
    - Stein discrepancy ⇔ Fisher divergence (integration by parts)
    - Stein discrepancy approximated by optimizing a neural network $f_\phi(x)$: $R^D \to R^D$

Ranganath et al. Operator Variational Inference. NIPS 2016
Grathwohl et al. Learning the Stein Discrepancy for Training and Evaluating Energy-Based Models without Sampling. ICML 2020

Stein's method & score matching

- Stein discrepancy:

$$S(q,p) = \sup_{f \in F} E_{q(x)}[s_p(x)^\top f(x) + \nabla^\top f(x)],$$

$$s_p(x) = \nabla_x \log p(x)$$

- Selecting test functions $f \in F$:
  - As functions in a unit ball of an RKHS:
    - Closed-form optimal test function as "smoothed" score difference
    - Optimal test function can be computed (integration by parts)
    - Test power depends on the choice of kernel

Liu et al. A Kernelized Stein Discrepancy for Goodness-of-fit Tests. ICML 2016
Chwialkowski et al. A kernel test of goodness of fit. ICML 2016

Stein's method & score matching

- Stein discrepancy:

$$S(q,p) = \sup_{f \in F} E_{q(x)}[s_p(x)^\top f(x) + \nabla^\top f(x)],$$

$$s_p(x) = \nabla_x \log p(x)$$

- Curse-of-dimensionality problem:
  - $f \in F$ as L_2 integrable functions:
    - Optimizing $f_\phi(x)$: $R^D \to R^D$ is challenging in high dimensions
  - $f \in F$ as functions in a unit ball of an RKHS:
    - Open question of kernel choice in high dimensions

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Intuition of Slicing

Stein Operator:

$$\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^{\boldsymbol{T}} \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^{T} \boldsymbol{f}(\boldsymbol{x})$$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Intuition of Slicing

Stein Operator:

$$\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^T \boldsymbol{f}(\boldsymbol{x})$$

- Score function of p

$$\boldsymbol{s_p}(\boldsymbol{x}): \mathbb{R}^D \to \mathbb{R}^D$$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Intuition of Slicing

Stein Operator:

$$\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^T \boldsymbol{f}(\boldsymbol{x})$$

- Score function of p

$$\boldsymbol{s_p}(\boldsymbol{x}): \mathbb{R}^D \to \mathbb{R}^D$$

- Solution: Project it by directions $\boldsymbol{r} \in \mathbb{S}^{D-1}$

$$s_p^r(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{r} \in \mathbb{R}$$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Intuition of Slicing

Stein Operator:

$$\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^T \boldsymbol{f}(\boldsymbol{x})$$

- Score function of p

$$\boldsymbol{s_p}(\boldsymbol{x}): \mathbb{R}^D \to \mathbb{R}^D$$

- Solution: Project it by directions $\boldsymbol{r} \in \mathbb{S}^{D-1}$

$$s_p^r(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{r} \in \mathbb{R}$$

- Equivalence: $\boldsymbol{s_p}(\boldsymbol{x}) = \boldsymbol{s_q}(\boldsymbol{x}) \Leftrightarrow s_p^r(\boldsymbol{x}) = s_q^r(\boldsymbol{x})$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Intuition of Slicing

Stein Operator:

$$\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^T \boldsymbol{f}(\boldsymbol{x})$$

- Test function $\boldsymbol{f}(\boldsymbol{x})$

$$\boldsymbol{f}(\boldsymbol{x}): \mathbb{R}^D \to \mathbb{R}^D$$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Intuition of Slicing

Stein Operator:

$$\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^T \boldsymbol{f}(\boldsymbol{x})$$

- Test function $\boldsymbol{f}(\boldsymbol{x})$

$$\boldsymbol{f}(\boldsymbol{x}): \mathbb{R}^D \to \mathbb{R}^D$$

- Need to match dimension $s_p^r(\boldsymbol{x}) \in \mathbb{R}$

$$\boldsymbol{f}(\boldsymbol{x}) \Rightarrow \{f_r(\boldsymbol{x}): \mathbb{R}^D \to \mathbb{R}\}_{r \in \mathbb{S}^{D-1}}$$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Intuition of Slicing

Stein Operator:

$$\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^T \boldsymbol{f}(\boldsymbol{x})$$

- Test function input $\boldsymbol{x}$

$$\boldsymbol{x} \in \mathbb{R}^D$$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Intuition of Slicing

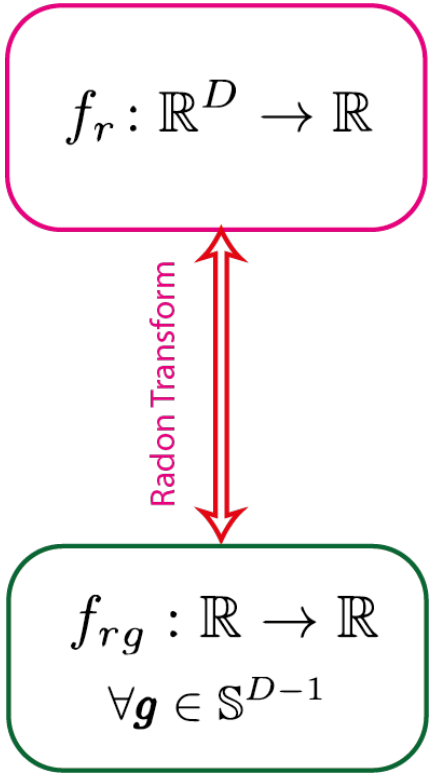
Stein Operator:

$$\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^T \boldsymbol{f}(\boldsymbol{x})$$

- Test function input $\boldsymbol{x}$

$$\boldsymbol{x} \in \mathbb{R}^D$$

- Solution: Radon Transform on $f_r(\boldsymbol{x})$ by direction $\boldsymbol{g}$



Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Intuition of Slicing

Stein Operator:

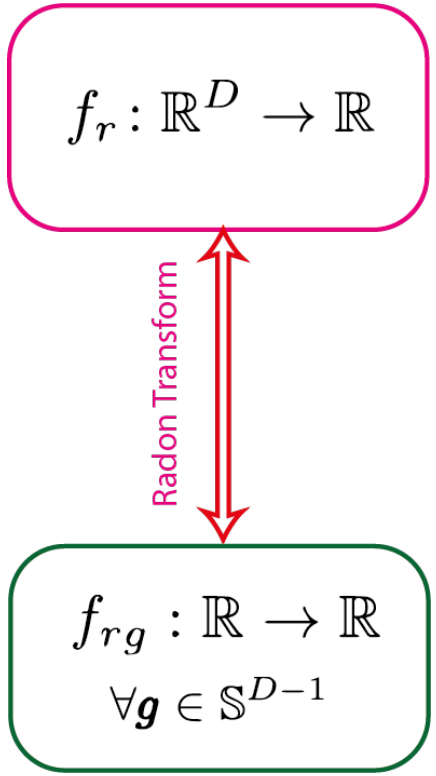
$$\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^T \boldsymbol{f}(\boldsymbol{x})$$

- Test function input $\boldsymbol{x}$

$$\boldsymbol{x} \in \mathbb{R}^D$$

- Solution: Radon Transform on $f_r(\boldsymbol{x})$ by direction $\boldsymbol{g}$

$$f_r(\boldsymbol{x}) \Rightarrow \{f_{rg}(\boldsymbol{x}^T \boldsymbol{g})\}_{\boldsymbol{g} \in \mathbb{S}^{D-1}}$$



Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Intuition of Slicing

Stein Operator:

$$\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{s_p}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^T \boldsymbol{f}(\boldsymbol{x})$$

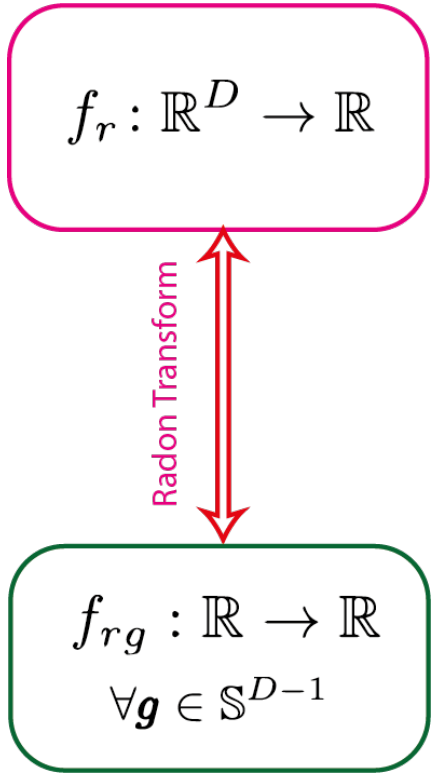
- Test function input $\boldsymbol{x}$

$$\boldsymbol{x} \in \mathbb{R}^D$$

- Solution: Radon Transform on $f_r(\boldsymbol{x})$ by direction $\boldsymbol{g}$

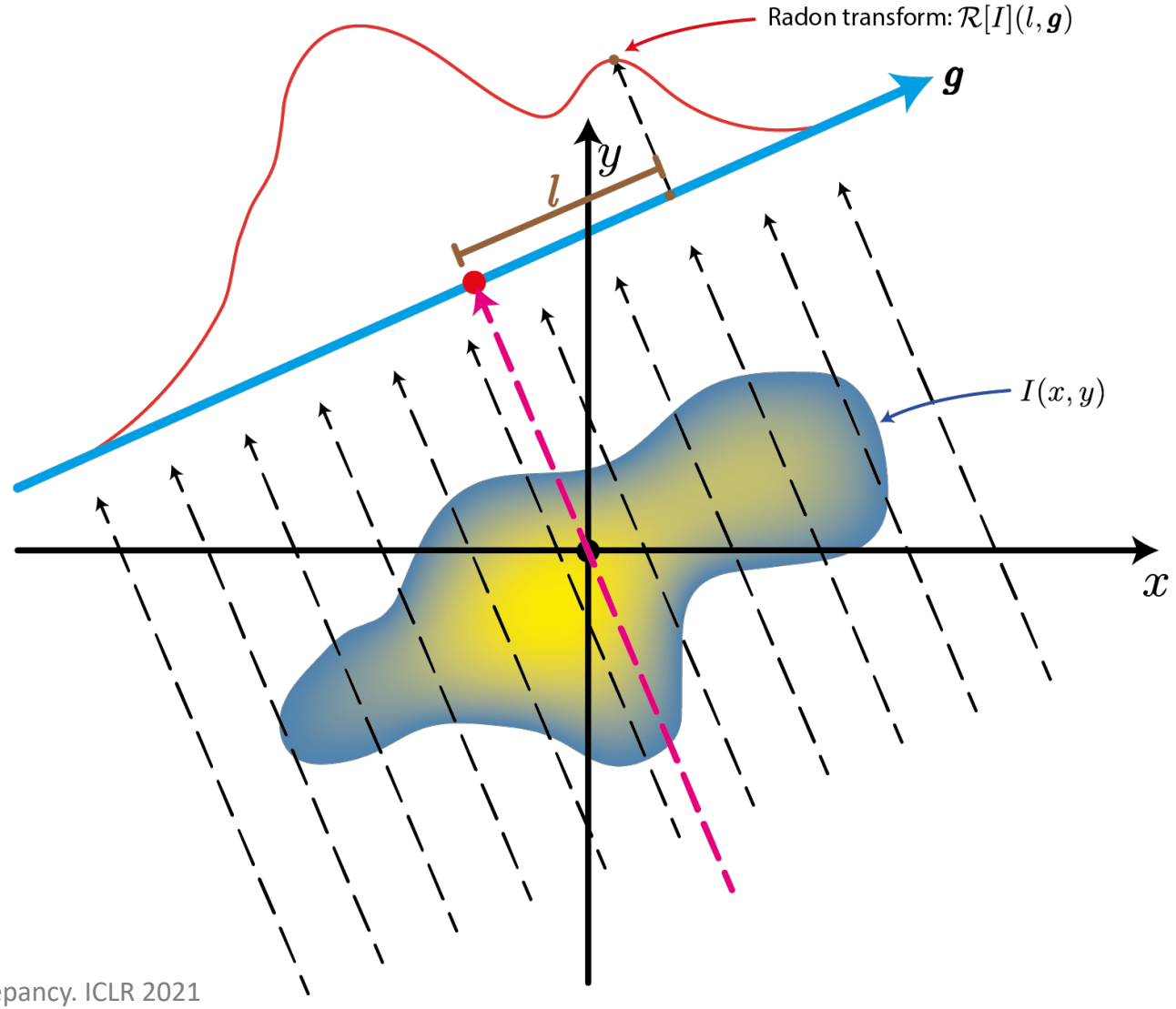
$$f_r(\boldsymbol{x}) \Rightarrow \{f_{rg}(\boldsymbol{x}^T \boldsymbol{g})\}_{\boldsymbol{g} \in \mathbb{S}^{D-1}}$$

- Property: Radon Transform is **invertible!**



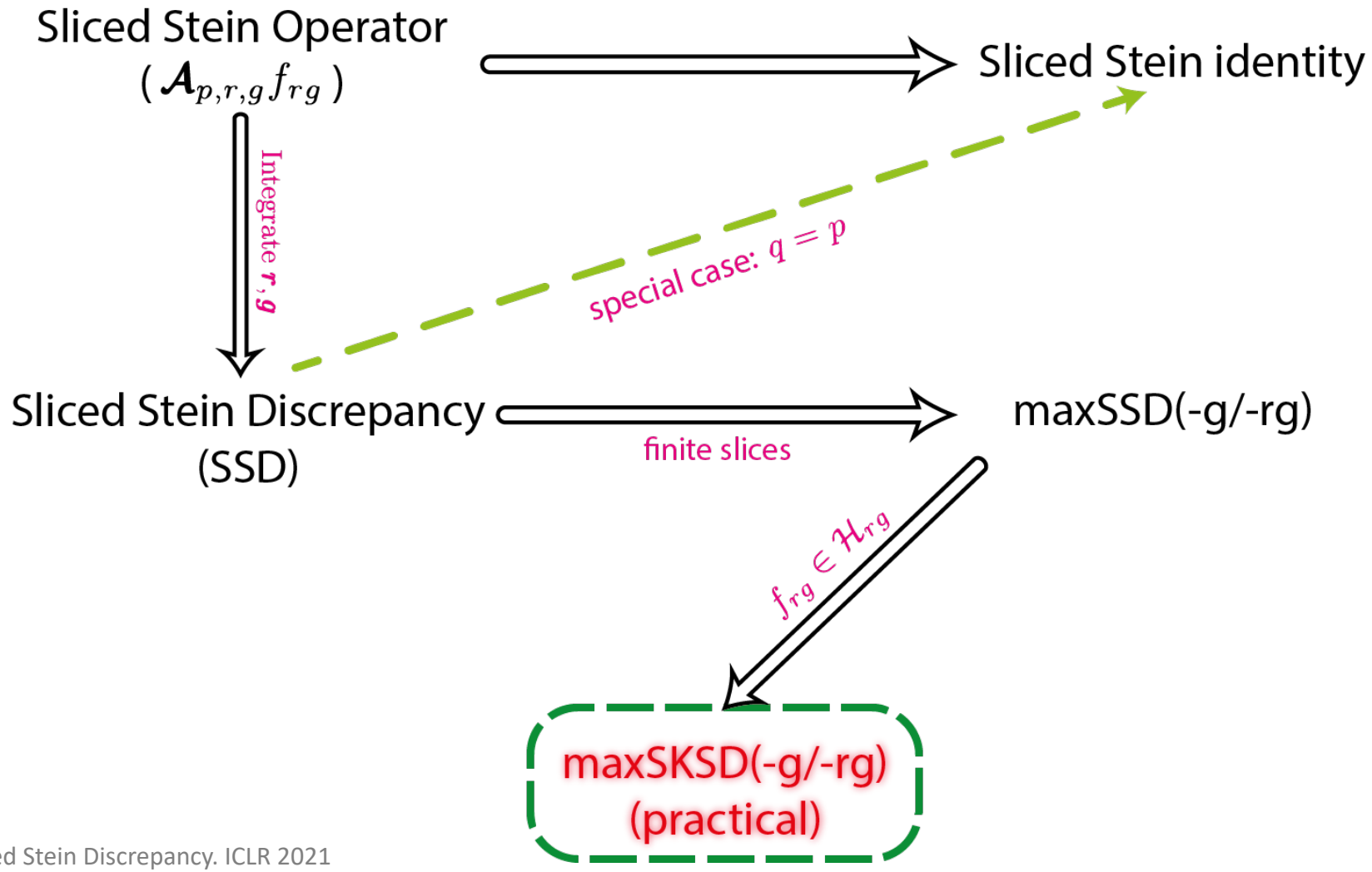
Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Radon Transform



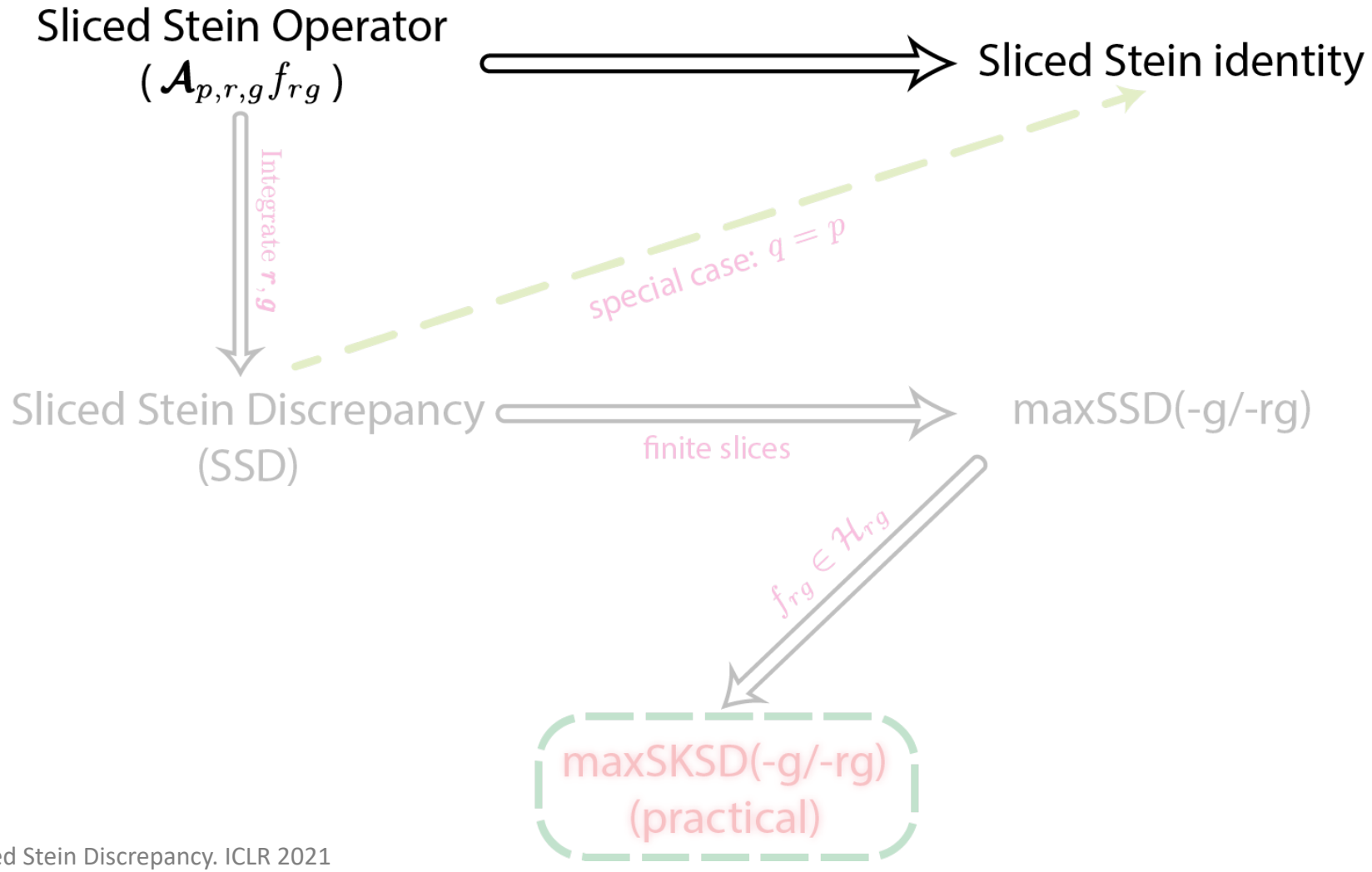
Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Construction of SKSD



Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Construction of SKSD



Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Sliced Stein Operator

Stein Operator:

$$\mathcal{A}_p \boldsymbol{f} = \boldsymbol{s_p}(\boldsymbol{x})^{\boldsymbol{T}} \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^{T} \boldsymbol{f}(\boldsymbol{x})$$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Sliced Stein Operator

Stein Operator:

$$\mathcal{A}_p \boldsymbol{f} = \boldsymbol{s_p}(\boldsymbol{x})^{\boldsymbol{T}} \boldsymbol{f}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}^T \boldsymbol{f}(\boldsymbol{x})$$

Sliced Stein Operator:

$$\mathcal{A}_{p,r,g} f_{rg} = s_p^r(\boldsymbol{x}) f_{rg}\left(\boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{g}\right) + \boldsymbol{r}^{\boldsymbol{T}} \boldsymbol{g} \nabla_{\boldsymbol{x}^T \boldsymbol{g}} f_{rg}(\boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{g})$$

1. Score function $\boldsymbol{s_p}(\boldsymbol{x}) \Rightarrow s_p^r(\boldsymbol{x})$

2. Test function $\boldsymbol{f}(\boldsymbol{x}) \Rightarrow f_{rg}(\boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{g})$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Sliced Stein Identity

Stein Identity:

$$\mathbb{E}_p\left[\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x})\right] = 0$$

If smooth $\boldsymbol{f}$ satisfies this, then $\boldsymbol{f}$ belongs to Stein class of $p(\boldsymbol{x})$

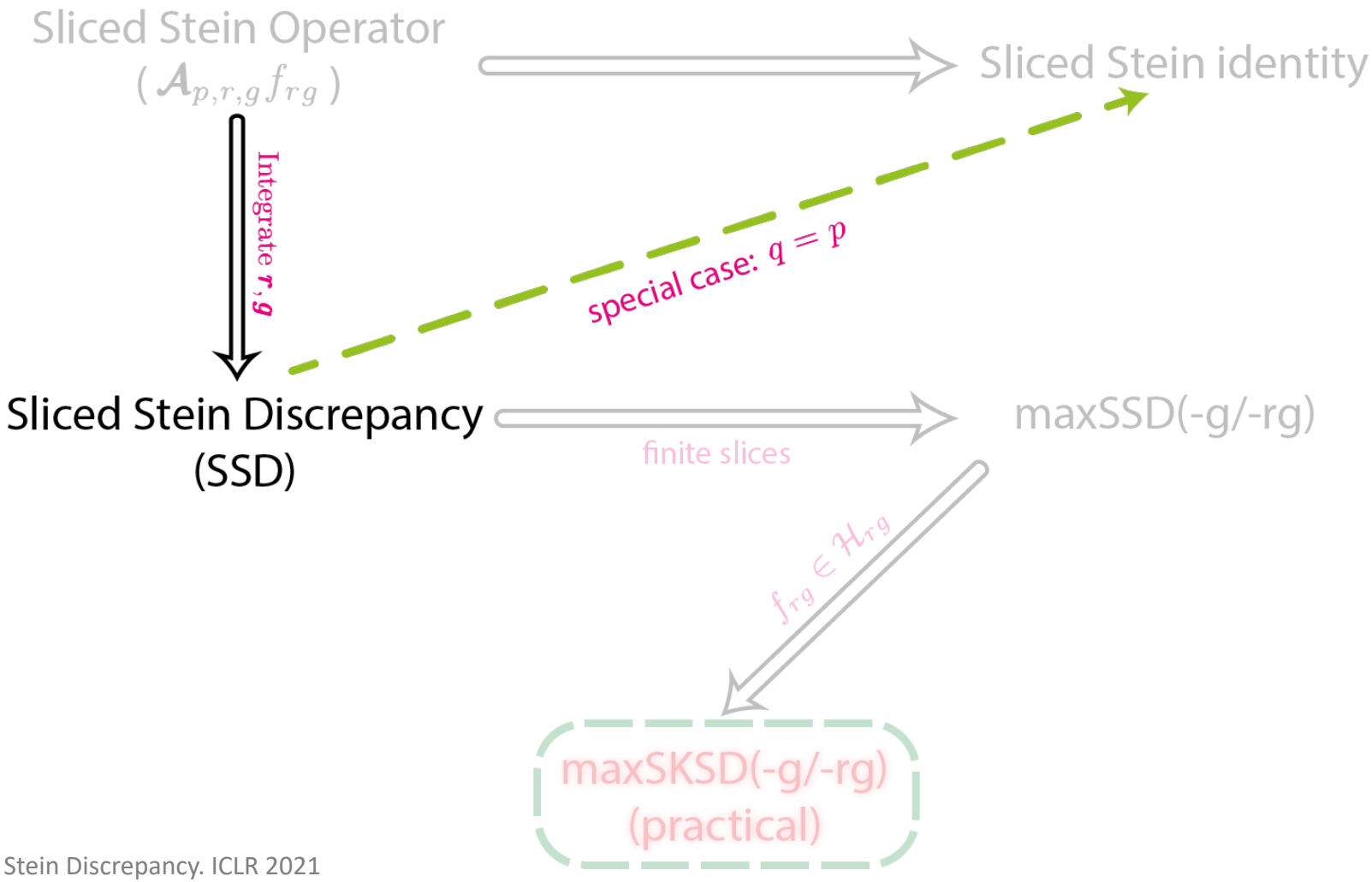
Sliced Stein Identity (with direction pair $\boldsymbol{r}, \boldsymbol{g}$):

$$\mathbb{E}_p\left[\mathcal{A}_{p,r,g} f_{rg}\left(\boldsymbol{x}^T \boldsymbol{g}\right)\right] = 0$$

with f_{rg} in Stein class of $p(\boldsymbol{x})$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Construction of SKSD



Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Sliced Stein Discrepancy

Stein Discrepancy:

$$D(q,p) = \sup_{\boldsymbol{f} \in \mathcal{F}_q} \mathbb{E}_q[\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x})]$$

Sliced Stein Discrepancy:

$$S(q,p) = \mathbb{E}_{p_r,p_g}[\sup_{f_{rg} \in \mathcal{F}_q} \mathbb{E}_q[\mathcal{A}_{p,r,g} f_{rg}(\boldsymbol{x}^T \boldsymbol{g})]]$$

1. Similar inner part, $\mathcal{A}_p \Rightarrow \mathcal{A}_{p,r,g}$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Sliced Stein Discrepancy

Stein Discrepancy:

$$D(q,p) = \sup_{\boldsymbol{f} \in \mathcal{F}_q} \mathbb{E}_q[\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x})]$$

Sliced Stein Discrepancy:

$$S(q,p) = \mathbb{E}_{p_r,p_g}[\sup_{f_{rg} \in \mathcal{F}_q} \mathbb{E}_q[\mathcal{A}_{p,r,g} f_{rg}(\boldsymbol{x}^T \boldsymbol{g})]]$$

1. Similar inner part, $\mathcal{A}_p \Rightarrow \mathcal{A}_{p,r,g}$
2. Consider all $\boldsymbol{r}, \boldsymbol{g} \in \mathbb{S}^{D-1}$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Sliced Stein Discrepancy

Stein Discrepancy:

$$D(q,p) = \sup_{\boldsymbol{f} \in \mathcal{F}_q} \mathbb{E}_q[\mathcal{A}_p \boldsymbol{f}(\boldsymbol{x})]$$

Sliced Stein Discrepancy:

$$S(q,p) = \mathbb{E}_{p_r,p_g}[\sup_{f_{rg} \in \mathcal{F}_q} \mathbb{E}_q[\mathcal{A}_{p,r,g} f_{rg}(\boldsymbol{x}^T \boldsymbol{g})]]$$

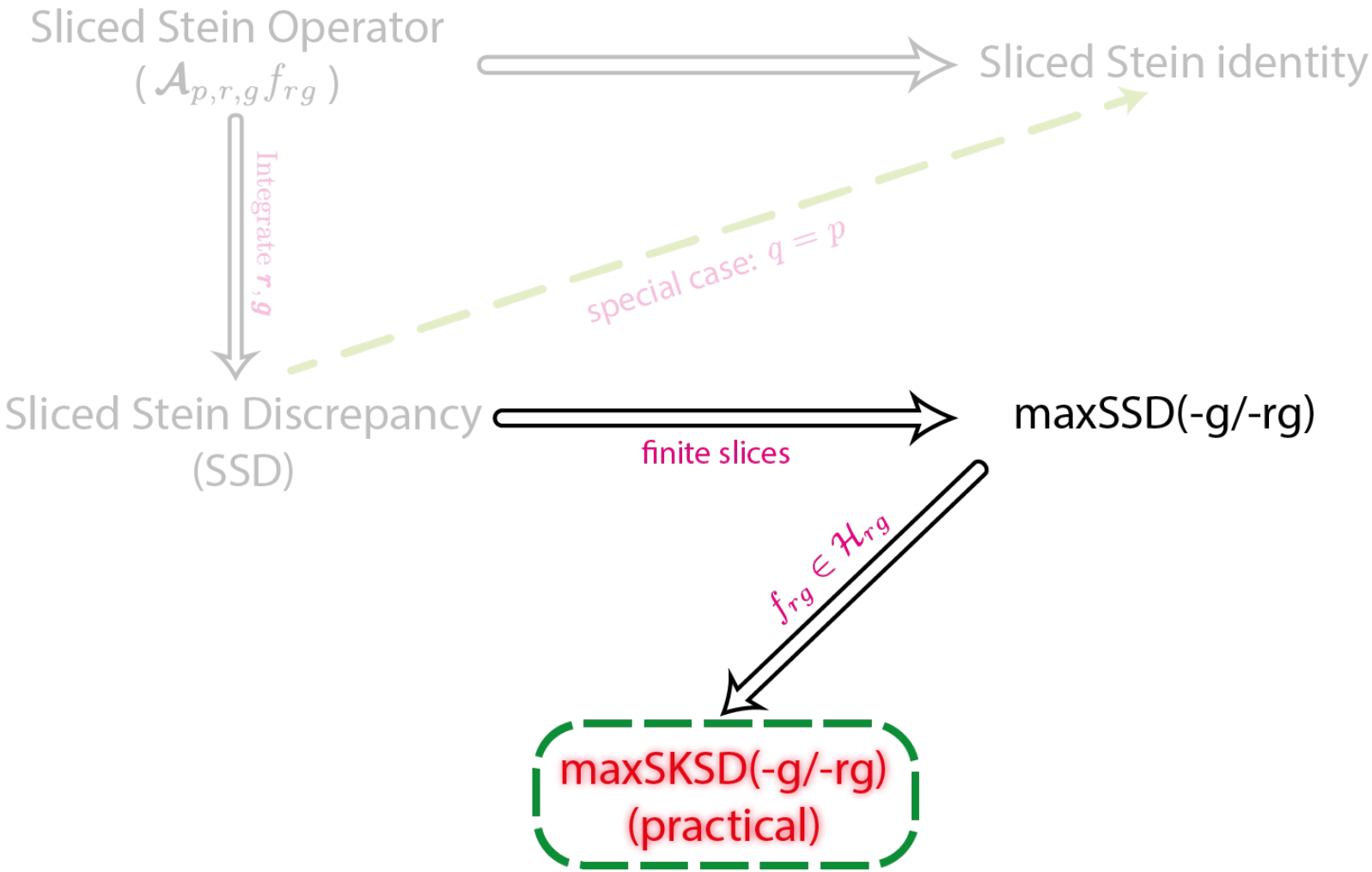
1. Similar inner part, $\mathcal{A}_p \Rightarrow \mathcal{A}_{p,r,g}$

2. Consider all $\boldsymbol{r}, \boldsymbol{g} \in \mathbb{S}^{D-1}$

Intractable due to (1) $\mathbb{E}_{p_r,p_g}$ (2) $\sup_{f_{rg} \in \mathcal{F}_q}$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Construction of SKSD



Finite Slices

Slice direction $\boldsymbol{r}$:

Key insight:

$$\boldsymbol{s_p} = \boldsymbol{s_q} \Leftrightarrow s_p^r = s_q^r \text{ for } \boldsymbol{r} = [1,0,\dots,0],\dots$$

Replace $\mathbb{E}_{p\_r}[\dots]$ by $\sum_{\boldsymbol{r}\in O_r} \dots$ where O_r is set of orthogonal basis

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Finite Slices

Slice direction $\boldsymbol{g}$:

Key insight:

Only need 1 difference to judge $p \neq q$

Replace $\mathbb{E}_{p_g}[\dots]$ by $\sup_{\boldsymbol{g} \in \mathbb{S}^{D-1}} \dots$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Finite Slice: *maxSSD-g/-rg*

Sliced Stein Discrepancy:

$$S(q,p) = \mathbb{E}_{p_r,p_g}\left[\sup_{f_{rg}\in\mathcal{F}_q} \mathbb{E}_q\left[\mathcal{A}_{p,r,g} f_{rg}\left(\boldsymbol{x}^T\boldsymbol{g}\right)\right]\right]$$

maxSSD-g:

$$S_{\max-\mathrm{g}}(\mathrm{q},\mathrm{p}) = \sum_{\boldsymbol{r}\in O_r} \sup_{\substack{f_{rg}\in\mathcal{F}_q \\ \boldsymbol{g}\in\mathbb{S}^{D-1}}} \mathbb{E}_q\left[\mathcal{A}_{p,r,g} f_{rg}\left(\boldsymbol{x}^T\boldsymbol{g}\right)\right]$$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Finite Slice: *maxSSD-g/-rg*

Sliced Stein Discrepancy:

$$S(q,p) = \mathbb{E}_{p_r,p_g}\left[\sup_{f_{rg}\in\mathcal{F}_q} \mathbb{E}_q\left[\mathcal{A}_{p,r,g} f_{rg}\left(\boldsymbol{x}^T \boldsymbol{g}\right)\right]\right]$$

maxSSD-rg:

$$S_{\max-\mathrm{rg}}(\mathrm{q},\mathrm{p}) = \sup_{\substack{f_{rg}\in\mathcal{F}_q \\ \boldsymbol{r},\boldsymbol{g}\in\mathbb{S}^{D-1}}} \mathbb{E}_q\left[\mathcal{A}_{p,r,g} f_{rg}\left(\boldsymbol{x}^T \boldsymbol{g}\right)\right]$$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Kernel Trick

Kernelized Stein Discrepancy (KSD):

$$D_k(q,p) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim q}[u_p(\boldsymbol{x},\boldsymbol{x}')]$$

with tractable $u_p(\boldsymbol{x},\boldsymbol{x}')$.

maxSKSD-g:

$$SK_{\max-\mathrm{g}}(\mathrm{q},\mathrm{p}) = \sum_{\boldsymbol{r}\in O_r} \sup_{\boldsymbol{g}\in\mathbb{S}^{D-1}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim q}[h_{p,r,g}(\boldsymbol{x},\boldsymbol{x}')]$$

with tractable $h_{p,r,g}(\boldsymbol{x},\boldsymbol{x}')$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Kernel Trick

Kernelized Stein Discrepancy (KSD):

$$D_k(q,p) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim q}[u_p(\boldsymbol{x},\boldsymbol{x}')]$$

with **tractable** $u_p(\boldsymbol{x},\boldsymbol{x}')$**.**

maxSKSD-rg:

$$SK_{\text{max-rg}}(\text{q},\text{p}) = \sup_{\boldsymbol{r},\boldsymbol{g}\in\mathbb{S}^{D-1}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim q}\left[h_{p,r,g}(\boldsymbol{x},\boldsymbol{x}')\right]$$

with **tractable** $h_{p,r,g}(\boldsymbol{x},\boldsymbol{x}')$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Kernel Trick

Kernelized Stein Discrepancy (KSD):

$$D_k(q,p) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim q}[u_p(\boldsymbol{x},\boldsymbol{x}')]$$

with **tractable** $u_p(\boldsymbol{x},\boldsymbol{x}')$**.**

maxSKSD-rg:

$$SK_{\text{max-rg}}(\text{q},\text{p}) = \sup_{\boldsymbol{r},\boldsymbol{g}\in\mathbb{S}^{D-1}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim q}\left[h_{p,r,g}(\boldsymbol{x},\boldsymbol{x}')\right]$$

with **tractable** $h_{p,r,g}(\boldsymbol{x},\boldsymbol{x}')$

Optimal test function for **KSD** is kernel-smoothed $\boldsymbol{s_p(x)} - \boldsymbol{s_q(x)}$**.** For maxSKSD-g/-rg, it is kernel-smoothed, Radon transformed ($\boldsymbol{g}$), $s_p^r(\boldsymbol{x}) - s_q^r(\boldsymbol{x})$

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

ICA Model Learning

ICA generative process:

$$\boldsymbol{z} \sim Lap(0,1),\ \boldsymbol{x} = \boldsymbol{W}\boldsymbol{z}, \quad \log p(\boldsymbol{x}) = \log p_z\left(\boldsymbol{W}^{-1}\boldsymbol{x}\right) + C$$

with unknown normalizing constant C.

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

ICA Model Learning

ICA generative process:

$$\boldsymbol{z} \sim Lap(0,1),\ \boldsymbol{x} = \boldsymbol{W}\boldsymbol{z}, \quad \log p(\boldsymbol{x}) = \log p_z\left(\boldsymbol{W}^{-1}\boldsymbol{x}\right) + C$$

With normalizing constant C.

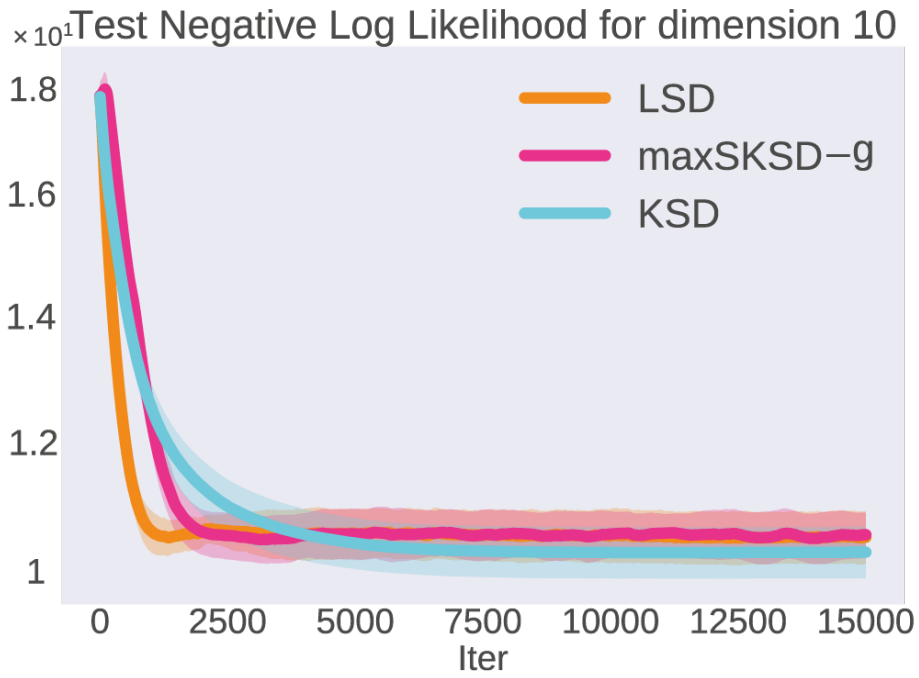
Training/evaluation setup:

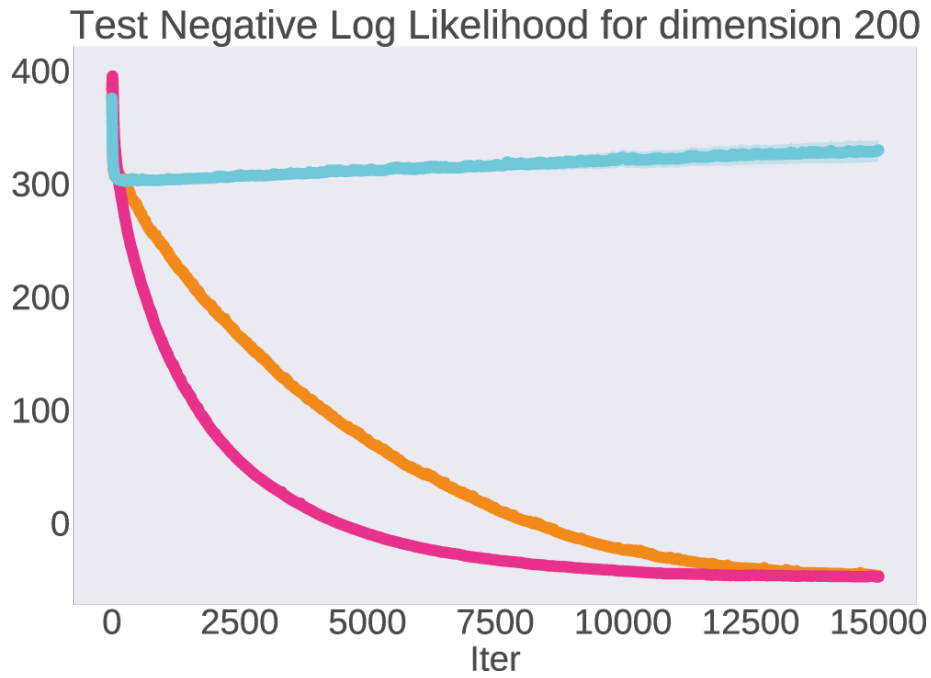
- Generate dataset $\boldsymbol{x}$ by ICA with ground trutch $\boldsymbol{W}_t$
- Model $p(\boldsymbol{x})$ is an ICA with random initialized $\boldsymbol{W}$.
- Train $p(\boldsymbol{x})$ to match dataset $\boldsymbol{x}$.
- Evaluate with **negative log likelihood (NLL.)** on test data**.**

Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

ICA Model Learning

Method	Dimension						
	$D = 10$	$D = 20$	$D = 40$	$D = 60$	$D = 80$	$D = 100$	$D = 200$
KSD	**-10.23**	-15.98	-34.50	-56.87	-86.09	-116.51	-329.49
LSD	-10.42	-14.54	**-17.16**	**-15.05**	-12.39	-5.49	46.63
maxSKSD−g	-10.45	**-14.50**	-17.28	-15.70	**-11.91**	**-4.21**	**47.72**





Gong et al. Sliced Kernelized Stein Discrepancy. ICLR 2021

Summary

Contributions:

- Propose tractable *maxSKSD-g/-rg* to address the curse-of-dimensionality of *KSD.*

Summary

Contributions:

- Propose tractable *maxSKSD-g/-rg* to address the curse-of-dimensionality of *KSD.*

- Superior performance as training objective in high dimensions.

Summary

Contributions:

- Propose tractable *maxSKSD-g/-rg* to address the curse-of-dimensionality of *KSD.*

- Superior performance as training objective in high dimensions.

- Advantages of closed-form kernel test function.

Summary

Extension:

- Intractable optimal slice directions $\boldsymbol{r}, \boldsymbol{g} \in \mathbb{S}^{D-1}$
  - Extension work: **Active Slices for Sliced Stein Discrepancy**[1]

Gong et al. Active Slices for Sliced Stein Discrepancy. arXiv:2102.03159

Summary

Extension:

- Intractable optimal slice directions $\boldsymbol{r}, \boldsymbol{g} \in \mathbb{S}^{D-1}$
  - Extension work: **Active Slices for Sliced Stein Discrepancy**[1]

- Applications in large model:
  - Deep Kernel extension to *maxSKSD-g/-rg*

Gong et al. Active Slices for Sliced Stein Discrepancy. arXiv:2102.03159

Looking forward

- Sampling from EBMs: far from being addressed
  - Better tempering/smoothing strategies as key
  - EBMs with discrete/mixed-type variables?

Looking forward

- Sampling from EBMs: far from being addressed
  - Better tempering/smoothing strategies as key
  - EBMs with discrete/mixed-type variables?
- Worth revisiting:
  - Message passing techniques (combine with advances in GNNs)
  - EBM inspired unsupervised/self-supervised pre-training

Looking forward

- Sampling from EBMs: far from being addressed
  - Better tempering/smoothing strategies as key
  - EBMs with discrete/mixed-type variables?
- Worth revisiting:
  - Message passing techniques (combine with advances in GNNs)
  - EBM inspired unsupervised/self-supervised pre-training

Do we need accurate EBM inference as a sub-routine for learning?

THANK YOU FOR LISTENING!