Amortised learning by wake sleep environment

Kevin, Ted Moskovitz, Heishiro Kanagawa, Maneesh Sahani

Gatsby Computational Neuroscience Unit, University College London

February 25, 2020

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Caputure rich data by

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Caputure rich data by

- model structure: hierarchical, squential

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Caputure rich data by

- model structure: hierarchical, squential
- latent variables: continuous, discrete, positive, geometry of support

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Caputure rich data by

- model structure: hierarchical, squential
- latent variables: continuous, discrete, positive, geometry of support

Has wide practical use

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Caputure rich data by

- model structure: hierarchical, squential
- latent variables: continuous, discrete, positive, geometry of support

Has wide practical use

- infer $\boldsymbol{z}$

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Caputure rich data by

- model structure: hierarchical, squential
- latent variables: continuous, discrete, positive, geometry of support

Has wide practical use

- infer $\boldsymbol{z}$
- generate $\boldsymbol{x}$ (prediction, imputation)

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Caputure rich data by

- model structure: hierarchical, squential
- latent variables: continuous, discrete, positive, geometry of support

Has wide practical use

- infer $\boldsymbol{z}$
- generate $\boldsymbol{x}$ (prediction, imputation)
- compression, outlier detection, and so on

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Caputure rich data by

- model structure: hierarchical, squential
- latent variables: continuous, discrete, positive, geometry of support

Has wide practical use

- infer $\boldsymbol{z}$
- generate $\boldsymbol{x}$ (prediction, imputation)
- compression, outlier detection, and so on

Only if we find the **correct parameters** for data

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Caputure rich data by

- model structure: hierarchical, squential
- latent variables: continuous, discrete, positive, geometry of support

Has wide practical use

- infer $\boldsymbol{z}$
- generate $\boldsymbol{x}$ (prediction, imputation)
- compression, outlier detection, and so on

Only if we find the **correct parameters** for data

- objectives: maximum likelihood (KL), score matching (Fisher), Jenson-Shannon, Wasserstein

Latent variable models

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$$

Caputure rich data by

- model structure: hierarchical, squential
- latent variables: continuous, discrete, positive, geometry of support

Has wide practical use

- infer $\boldsymbol{z}$
- generate $\boldsymbol{x}$ (prediction, imputation)
- compression, outlier detection, and so on

Only if we find the **correct parameters** for data

- objectives: maximum likelihood (KL), score matching (Fisher), Jenson-Shannon, Wasserstein

This talk: maximum likelihood

Maximum likelkhood estimation

Maximum likelihood: maximise $\mathbb{E}_{p^*(\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x})\right]$

Maximum likelkhood estimation

Maximum likelihood: maximise $\mathbb{E}_{p^*(\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x})\right]$
Nice properties as $n \to \infty$:

Maximum likelkhood estimation

Maximum likelihood: maximise $\mathbb{E}_{p^*(\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x})]$
Nice properties as $n \to \infty$:

- consistency:

$$p\left(\left|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\right| \geq \epsilon\right) \to 0$$

Maximum likelkhood estimation

Maximum likelihood: maximise $\mathbb{E}_{p^*(\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x})\right]$
Nice properties as $n \to \infty$:

- consistency:

$$p\left(\left|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\right| \geq \epsilon\right) \to 0$$

- asympototic efficiency and normality (conf. bounds)

$$\sqrt{n}\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\right) \xrightarrow{d} \mathcal{N}\left(0, I^{-1}\right) \quad I = -\mathbb{E}_p\left[\nabla^2_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}^*}\right]$$

Maximum likelkhood estimation

Maximum likelihood: maximise $\mathbb{E}_{p^*(\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x})\right]$
Nice properties as $n \to \infty$:

- consistency:

$$p\left(\left|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\right| \geq \epsilon\right) \to 0$$

- asympototic efficiency and normality (conf. bounds)

$$\sqrt{n}\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\right) \xrightarrow{d} \mathcal{N}\left(0, I^{-1}\right) \quad I = -\mathbb{E}_p\left[\nabla^2_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}^*}\right]$$

Can optimise by gradient method, but requires an intractable integral

$$\Delta_{\boldsymbol{\theta}}(\boldsymbol{x}) := \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \quad \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) d\boldsymbol{z}$$

Expectation-Maximisation: variational inference for learning

Instead, use the lower bound to derive the EM algorihtm

$$\mathcal{F}(q, \boldsymbol{\theta}) := \mathbb{E}_{q(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] + \mathbb{H}\left[q\right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Expectation-Maximisation: variational inference for learning

Instead, use the lower bound to derive the EM algorihtm

$$\mathcal{F}(q, \boldsymbol{\theta}) := \mathbb{E}_{q(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] + \mathbb{H}\left[q\right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Bound is tight for $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$.

Expectation-Maximisation: variational inference for learning

Instead, use the lower bound to derive the EM algorihtm

$$\mathcal{F}(q, \boldsymbol{\theta}) := \mathbb{E}_{q(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] + \mathbb{H}\left[q\right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Bound is tight for $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$.
To keep maximising $\mathcal{F}$, when $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ at iteration t:

Expectation-Maximisation: variational inference for learning

Instead, use the lower bound to derive the EM algorihtm

$$\mathcal{F}(q, \boldsymbol{\theta}) := \mathbb{E}_{q(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] + \mathbb{H}\left[q\right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Bound is tight for $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$.
To keep maximising $\mathcal{F}$, when $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ at iteration t:

- E-step: find $q(\boldsymbol{z})$ ***for each*** $\boldsymbol{x}^* \sim p^*$:

Expectation-Maximisation: variational inference for learning

Instead, use the lower bound to derive the EM algorihtm

$$\mathcal{F}(q, \boldsymbol{\theta}) := \mathbb{E}_{q(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] + \mathbb{H}\left[q\right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Bound is tight for $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$.
To keep maximising $\mathcal{F}$, when $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ at iteration t:

- E-step: find $q(\boldsymbol{z})$ ***for each*** $\boldsymbol{x}^* \sim p^*$:

$$\arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

Expectation-Maximisation: variational inference for learning

Instead, use the lower bound to derive the EM algorihtm

$$\mathcal{F}(q, \boldsymbol{\theta}) := \mathbb{E}_{q(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] + \mathbb{H}\left[q\right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Bound is tight for $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$.
To keep maximising $\mathcal{F}$, when $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ at iteration t:

- E-step: find $q(\boldsymbol{z})$ ***for each*** $\boldsymbol{x}^* \sim p^*$:

$$\arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\mu, \sigma} \mathrm{KL}\left[\mathcal{N}(\mu, \Sigma) \| p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

Expectation-Maximisation: variational inference for learning

Instead, use the lower bound to derive the EM algorihtm

$$\mathcal{F}(q, \boldsymbol{\theta}) := \mathbb{E}_{q(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] + \mathbb{H}\left[q\right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Bound is tight for $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$.
To keep maximising $\mathcal{F}$, when $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ at iteration t:

- E-step: find $q(\boldsymbol{z})$ ***for each*** $\boldsymbol{x}^* \sim p^*$:

$$\arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\mu, \sigma} \mathrm{KL}\left[\mathcal{N}(\mu, \Sigma) \| p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

- M-step: change $\boldsymbol{\theta}$ by

$$\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{z})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right]$$

Expectation-Maximisation: variational inference for learning

Instead, use the lower bound to derive the EM algorihtm

$$\mathcal{F}(q, \boldsymbol{\theta}) := \mathbb{E}_{q(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] + \mathbb{H}\left[q\right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Bound is tight for $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$.
To keep maximising $\mathcal{F}$, when $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ at iteration t:

- E-step: find $q(\boldsymbol{z})$ ***for each*** $\boldsymbol{x}^* \sim p^*$:

$$\arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left[q(\boldsymbol{z}) \| p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\mu, \sigma} \mathrm{KL}\left[\mathcal{N}(\mu, \Sigma) \| p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

- M-step: change $\boldsymbol{\theta}$ by

$$\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{z})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right]$$

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) \right]$ by Monte Carlo – **need samples**!

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) \right]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL} \left[q_{\phi}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*) \right]$$

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL} \left[q_{\phi}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*) \right] \quad \text{e.g.} \quad \arg\min_{\boldsymbol{\phi}} \mathrm{KL} \left[\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^*), \Sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*) \right]$$

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) \right]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL} \left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*) \right] \quad \text{e.g.} \quad \arg\min_{\boldsymbol{\phi}} \mathrm{KL} \left[\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^*), \Sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*) \right]$$

- Gradient w.r.t. $\boldsymbol{\phi}$: reparameterisation trick, score trick, wake-sleep

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^*), \Sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

- Gradient w.r.t. $\boldsymbol{\phi}$: reparameterisation trick, score trick, wake-sleep

- Pros:

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[q_{\phi}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^*), \Sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

- Gradient w.r.t. $\boldsymbol{\phi}$: reparameterisation trick, score trick, wake-sleep

- Pros:
  - scale to large $\boldsymbol{x}$

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^*), \Sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

- Gradient w.r.t. $\boldsymbol{\phi}$: reparameterisation trick, score trick, wake-sleep

- Pros:
  - scale to large $\boldsymbol{x}$
  - allows flexible $q(\boldsymbol{z})$

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^*), \Sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

- Gradient w.r.t. $\boldsymbol{\phi}$: reparameterisation trick, score trick, wake-sleep

- Pros:
  - scale to large $\boldsymbol{x}$
  - allows flexible $q(\boldsymbol{z})$
  - complicated $p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$ (?)

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^*), \Sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

- Gradient w.r.t. $\boldsymbol{\phi}$: reparameterisation trick, score trick, wake-sleep

- Pros:
  - scale to large $\boldsymbol{x}$
  - allows flexible $q(\boldsymbol{z})$
  - complicated $p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$ (?)
- But:

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^*), \Sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

- Gradient w.r.t. $\boldsymbol{\phi}$: reparameterisation trick, score trick, wake-sleep

- Pros:
  - scale to large $\boldsymbol{x}$
  - allows flexible $q(\boldsymbol{z})$
  - complicated $p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$ (?)

- But:
  - $q \overset{\mathrm{KL}}{\approx} p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \quad \not\Rightarrow \quad \nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) \overset{l\text{-}2}{\approx} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^*), \Sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

- Gradient w.r.t. $\boldsymbol{\phi}$: reparameterisation trick, score trick, wake-sleep

- Pros:
  - scale to large $\boldsymbol{x}$
  - allows flexible $q(\boldsymbol{z})$
  - complicated $p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$ (?)

- But:
  - $q \overset{\mathrm{KL}}{\approx} p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \quad \not\Rightarrow \quad \nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) \overset{l\text{-}2}{\approx} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$
  - reparameterisation depends on $\mathcal{Q}$

Amortised infernce for learning

- Approximate $\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{z})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})]$ by Monte Carlo – **need samples**!
- Inferential model with parameter $\boldsymbol{\phi}$: find $\boldsymbol{\phi}$ ***for all*** $\boldsymbol{x}^* \sim p^*$

$$\arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right] \quad \text{e.g.} \quad \arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^*), \Sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^*) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)\right]$$

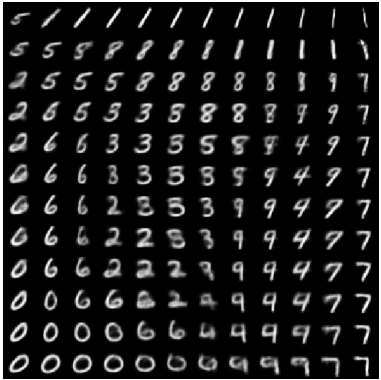
- Gradient w.r.t. $\boldsymbol{\phi}$: reparameterisation trick, score trick, wake-sleep

- Pros:
  - scale to large $\boldsymbol{x}$
  - allows flexible $q(\boldsymbol{z})$
  - complicated $p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$ (?)

- But:
  - $q \overset{\mathrm{KL}}{\approx} p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \quad \not\Rightarrow \quad \nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) \overset{l\text{-}2}{\approx} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$
  - reparameterisation depends on $\mathcal{Q}$
  - flexible $p(\boldsymbol{z}, \boldsymbol{x})$ can induce complex $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$

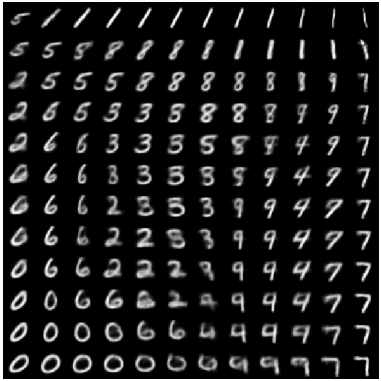
Example: complex posterior in Gaussian VAE

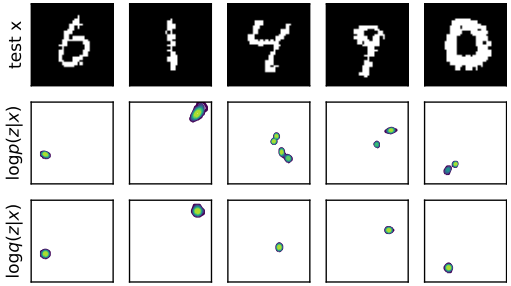
$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, where $\boldsymbol{z} \in \mathbb{R}^2$



Example: complex posterior in Gaussian VAE

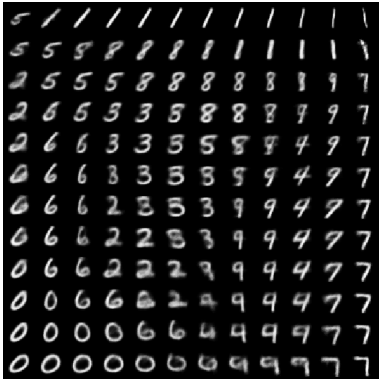
$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, where $\boldsymbol{z} \in \mathbb{R}^2$

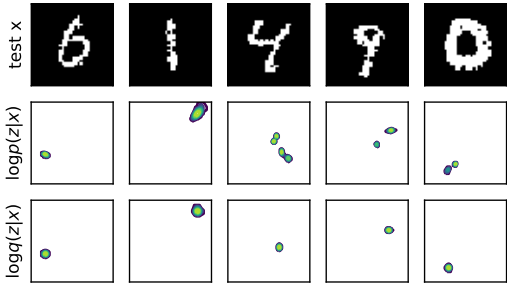




Example: complex posterior in Gaussian VAE

$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, where $\boldsymbol{z} \in \mathbb{R}^2$





This talk: maximum likelihood learning without inference

Amortised learning: focus on ML gradient

Suppose we are at iteration t with $\boldsymbol{\theta} = \boldsymbol{\theta}_t$, want

$$\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) := \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$$

Amortised learning: focus on ML gradient

Suppose we are at iteration t with $\boldsymbol{\theta} = \boldsymbol{\theta}_t$, want

$$\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) := \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$$

using an approximate q

Amortised learning: focus on ML gradient

Suppose we are at iteration t with $\boldsymbol{\theta} = \boldsymbol{\theta}_t$, want

$$\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) := \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$$

using an approximate q

$$\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta})$$

Amortised learning: focus on ML gradient

Suppose we are at iteration t with $\boldsymbol{\theta} = \boldsymbol{\theta}_t$, want

$$\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) := \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$$

using an approximate q

$$\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{z})} \left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) \right]$$

Amortised learning: focus on ML gradient

Suppose we are at iteration t with $\boldsymbol{\theta} = \boldsymbol{\theta}_t$, want

$$\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) := \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$$

using an approximate q

$$\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{z})} \left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) \right]$$

When $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})$, can show

$$\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathcal{F}(p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x}), \boldsymbol{\theta})|_{\boldsymbol{\theta}_t} = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})|_{\boldsymbol{\theta}_t} \right]$$

Amortised learning: focus on ML gradient

Suppose we are at iteration t with $\boldsymbol{\theta} = \boldsymbol{\theta}_t$, want

$$\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) := \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$$

using an approximate q

$$\nabla_{\boldsymbol{\theta}} \mathcal{F}(q, \boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{z})} \left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) \right]$$

When $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})$, can show

$$\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathcal{F}(p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x}), \boldsymbol{\theta})|_{\boldsymbol{\theta}_t} = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})|_{\boldsymbol{\theta}_t} \right]$$

Amortised learning: directly approximate $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x})$ above

Posterior mean from least square regression (LSR)

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})|_{\boldsymbol{\theta}_t}\right]$

Posterior mean from least square regression (LSR)

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})|_{\boldsymbol{\theta}_t}\right]$
- let $\boldsymbol{y}_n = \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)|_{\boldsymbol{\theta}_t}$

Posterior mean from least square regression (LSR)

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})|_{\boldsymbol{\theta}_t}\right]$
- let $\boldsymbol{y}_n = \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)|_{\boldsymbol{\theta}_t}$
- problem: given $p(\boldsymbol{x}, \boldsymbol{y})$, want $f : \boldsymbol{x} \mapsto \mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\boldsymbol{y}\right]$

Posterior mean from least square regression (LSR)

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})|_{\boldsymbol{\theta}_t}\right]$
- let $\boldsymbol{y}_n = \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)|_{\boldsymbol{\theta}_t}$
- problem: given $p(\boldsymbol{x}, \boldsymbol{y})$, want $f : \boldsymbol{x} \mapsto \mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}]$
- Estimate by LSR:

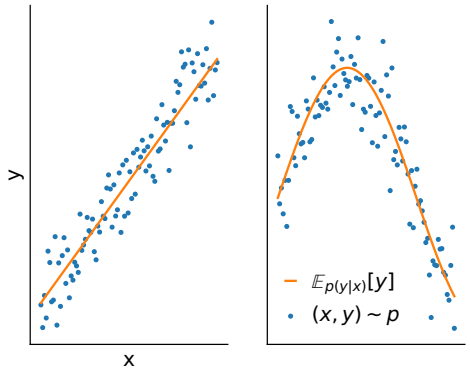
$$\min_{f \in \mathcal{H}_\gamma} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})}\left[\|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x})\|_2^2\right]$$

Posterior mean from least square regression (LSR)

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})|_{\boldsymbol{\theta}_t}\right]$
- let $\boldsymbol{y}_n = \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)|_{\boldsymbol{\theta}_t}$
- problem: given $p(\boldsymbol{x}, \boldsymbol{y})$, want $f : \boldsymbol{x} \mapsto \mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}]$
- Estimate by LSR:

$$\min_{f \in \mathcal{H}_\gamma} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})}\left[\|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x})\|_2^2\right]$$

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{y}_n - \boldsymbol{f}(\boldsymbol{x}_n)\|_2^2, \ \boldsymbol{x}_n, \boldsymbol{y}_n \sim p$$



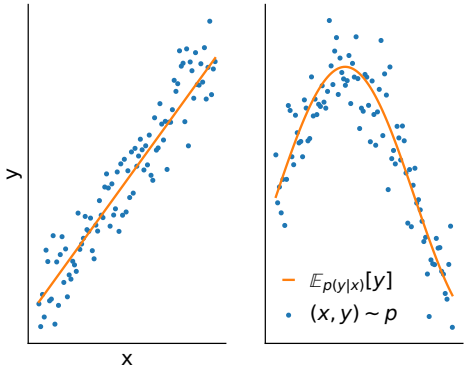
Posterior mean from least square regression (LSR)

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})|_{\boldsymbol{\theta}_t}\right]$
- let $\boldsymbol{y}_n = \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)|_{\boldsymbol{\theta}_t}$
- problem: given $p(\boldsymbol{x}, \boldsymbol{y})$, want $f : \boldsymbol{x} \mapsto \mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}]$
- Estimate by LSR:

$$\min_{f \in \mathcal{H}_{\boldsymbol{\gamma}}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})}\left[\|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x})\|_2^2\right]$$

- $\hat{\Delta}_{\boldsymbol{\theta}_t, \boldsymbol{\gamma}}(\boldsymbol{x}) = \hat{\boldsymbol{f}}_{\boldsymbol{\gamma}}(\boldsymbol{x})$

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{y}_n - \boldsymbol{f}(\boldsymbol{x}_n)\|_2^2, \ \boldsymbol{x}_n, \boldsymbol{y}_n \sim p$$



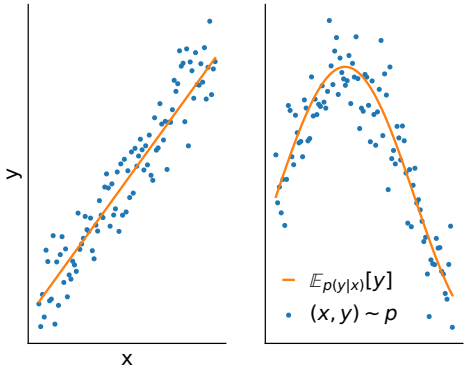
Posterior mean from least square regression (LSR)

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})|_{\boldsymbol{\theta}_t}\right]$
- let $\boldsymbol{y}_n = \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)|_{\boldsymbol{\theta}_t}$
- problem: given $p(\boldsymbol{x}, \boldsymbol{y})$, want $f : \boldsymbol{x} \mapsto \mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}]$
- Estimate by LSR:

$$\min_{f \in \mathcal{H}_{\boldsymbol{\gamma}}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})}\left[\|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x})\|_2^2\right]$$

- $\hat{\Delta}_{\boldsymbol{\theta}_t, \boldsymbol{\gamma}}(\boldsymbol{x}) = \hat{\boldsymbol{f}}_{\boldsymbol{\gamma}}(\boldsymbol{x})$
- But requires $\boldsymbol{y}_n = \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)|_{\boldsymbol{\theta}_t}$ for all n

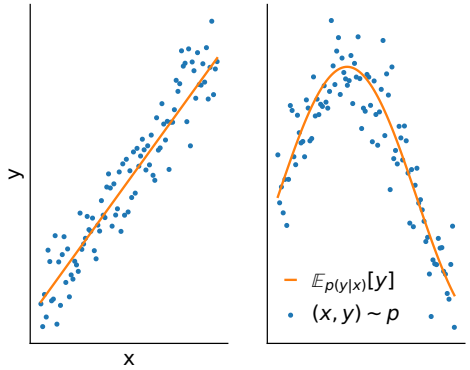
$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{y}_n - \boldsymbol{f}(\boldsymbol{x}_n)\|_2^2, \ \boldsymbol{x}_n, \boldsymbol{y}_n \sim p$$



Posterior mean from least square regression

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})|_{\boldsymbol{\theta}_t}\right]$

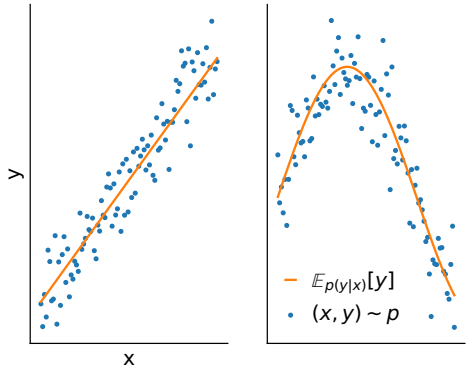
$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{y}_n - f(\boldsymbol{x}_n)\|_2^2, \ \boldsymbol{x}_n, \boldsymbol{y}_n \sim p_{\boldsymbol{\theta}_t}$$



Posterior mean from least square regression

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] |_{\boldsymbol{\theta}_t}$

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{y}_n - f(\boldsymbol{x}_n)\|_2^2 \,, \; \boldsymbol{x}_n, \boldsymbol{y}_n \sim p_{\boldsymbol{\theta}_t}$$

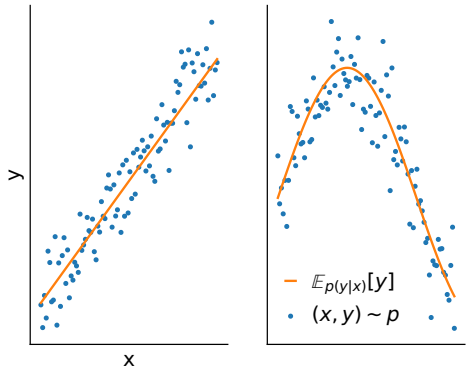


Posterior mean from least square regression

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] \big|_{\boldsymbol{\theta}_t}$
- estimate $y_{\boldsymbol{\theta},n} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$ and differentiate

$$\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(y,\boldsymbol{x})} \left[y_{\boldsymbol{\theta}}\right] \big|_{\boldsymbol{\theta}_t}$$

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{y}_n - f(\boldsymbol{x}_n)\|_2^2, \ \boldsymbol{x}_n, \boldsymbol{y}_n \sim p_{\boldsymbol{\theta}_t}$$

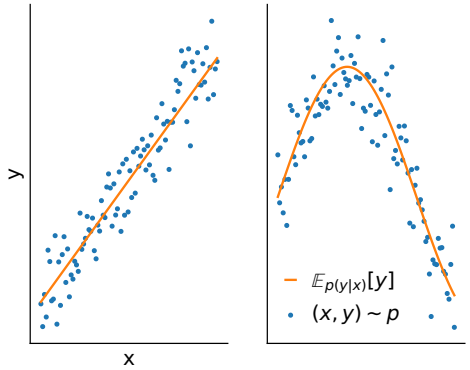


Posterior mean from least square regression

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] \big|_{\boldsymbol{\theta}_t}$
- estimate $y_{\boldsymbol{\theta},n} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$ and differentiate

$$\begin{aligned}\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(y,\boldsymbol{x})} \left[y_{\boldsymbol{\theta}}\right] \big|_{\boldsymbol{\theta}_t} \\ &= \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\boldsymbol{x}) \big|_{\boldsymbol{\theta}_t}\end{aligned}$$

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{y}_n - f(\boldsymbol{x}_n)\|_2^2, \ \boldsymbol{x}_n, \boldsymbol{y}_n \sim p_{\boldsymbol{\theta}_t}$$



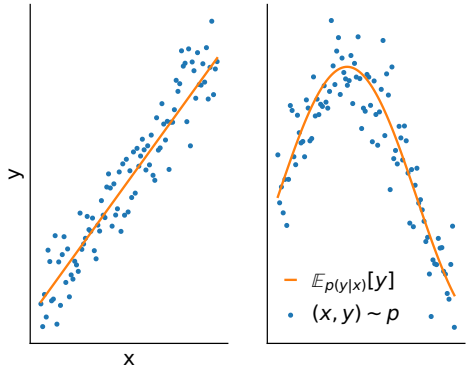
Posterior mean from least square regression

- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] \big|_{\boldsymbol{\theta}_t}$
- estimate $y_{\boldsymbol{\theta},n} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$ and differentiate

$$\begin{aligned}\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(y,\boldsymbol{x})} \left[y_{\boldsymbol{\theta}}\right] \big|_{\boldsymbol{\theta}_t} \\ &= \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\boldsymbol{x}) \big|_{\boldsymbol{\theta}_t}\end{aligned}$$

where $J_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right]$

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{y}_n - f(\boldsymbol{x}_n)\|_2^2, \ \boldsymbol{x}_n, \boldsymbol{y}_n \sim p_{\boldsymbol{\theta}_t}$$



Posterior mean from least square regression

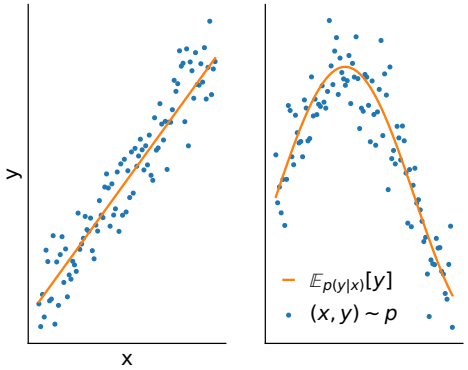
- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] \big|_{\boldsymbol{\theta}_t}$
- estimate $y_{\boldsymbol{\theta},n} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$ and differentiate

$$\begin{aligned}\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(y,\boldsymbol{x})} \left[y_{\boldsymbol{\theta}}\right] \big|_{\boldsymbol{\theta}_t} \\ &= \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\boldsymbol{x}) \big|_{\boldsymbol{\theta}_t}\end{aligned}$$

where $J_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right]$

- If estimator $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})$:
  - depends on $\boldsymbol{\theta}$ and t
  - differentiable w.r..t $\boldsymbol{\theta}$

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{y}_n - f(\boldsymbol{x}_n)\|_2^2, \ \boldsymbol{x}_n, \boldsymbol{y}_n \sim p_{\boldsymbol{\theta}_t}$$



Posterior mean from least square regression

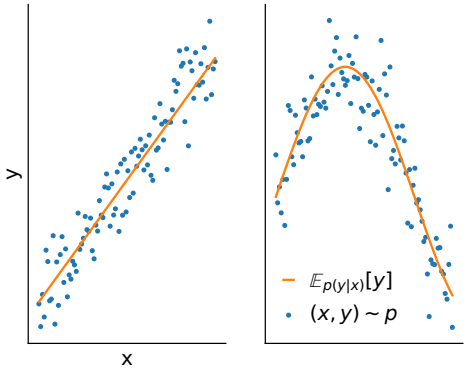
- want $\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right] |_{\boldsymbol{\theta}_t}$
- estimate $y_{\boldsymbol{\theta},n} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$ and differentiate

$$\begin{aligned}\Delta_{\boldsymbol{\theta}_t}(\boldsymbol{x}) &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(y,\boldsymbol{x})} \left[y_{\boldsymbol{\theta}}\right] \big|_{\boldsymbol{\theta}_t} \\ &= \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\boldsymbol{x}) \big|_{\boldsymbol{\theta}_t}\end{aligned}$$

where $J_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})\right]$

- If estimator $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})$:
  - depends on $\boldsymbol{\theta}$ and t
  - differentiable w.r..t $\boldsymbol{\theta}$
- Then $\boxed{\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}}$

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{y}_n - f(\boldsymbol{x}_n)\|_2^2, \ \boldsymbol{x}_n, \boldsymbol{y}_n \sim p_{\boldsymbol{\theta}_t}$$



Amortised learning

Find an estimator for $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}[y_\theta]$, $y_{\boldsymbol{\theta}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, so that $\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$

Amortised learning

Find an estimator for $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}[y_\theta]$, $y_{\boldsymbol{\theta}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, so that $\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$

- NN regression: $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \mathrm{NN}_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(\boldsymbol{x})$

Amortised learning

Find an estimator for $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}[y_\theta]$, $y_{\boldsymbol{\theta}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, so that $\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$

- NN regression: $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \mathrm{NN}_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(\boldsymbol{x})$

$$\boldsymbol{\gamma}_t(\boldsymbol{\theta}) \leftarrow \boldsymbol{\gamma}_{t-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left\| y_{\boldsymbol{\theta},n} - \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}_n) \right\|_2^2$$

Amortised learning

Find an estimator for $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}[y_\theta]$, $y_{\boldsymbol{\theta}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, so that $\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$

- NN regression: $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \mathrm{NN}_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(\boldsymbol{x})$

$$\boldsymbol{\gamma}_t(\boldsymbol{\theta}) \leftarrow \boldsymbol{\gamma}_{t-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left\| y_{\boldsymbol{\theta},n} - \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}_n) \right\|_2^2$$

- Assume a flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \sum_k \alpha_k \psi_i(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x})$

Amortised learning

Find an estimator for $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}[y_\theta]$, $y_{\boldsymbol{\theta}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, so that $\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$

- NN regression: $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \mathrm{NN}_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(\boldsymbol{x})$

$$\boldsymbol{\gamma}_t(\boldsymbol{\theta}) \leftarrow \boldsymbol{\gamma}_{t-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left\| y_{\boldsymbol{\theta},n} - \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}_n) \right\|_2^2$$

- Assume a flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \sum_k \alpha_k \psi_i(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x})$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \left\| y_{\boldsymbol{\theta},n} - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha} \right\|_{\mathcal{H}}^2$$

Amortised learning

Find an estimator for $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}[y_\theta]$, $y_{\boldsymbol{\theta}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, so that $\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$

- NN regression: $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \mathrm{NN}_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(\boldsymbol{x})$

$$\boldsymbol{\gamma}_t(\boldsymbol{\theta}) \leftarrow \boldsymbol{\gamma}_{t-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left\| y_{\boldsymbol{\theta},n} - \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}_n) \right\|_2^2$$

- Assume a flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \sum_k \alpha_k \psi_i(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x})$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \|y_{\boldsymbol{\theta},n} - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x})\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_{\mathcal{H}}^2$$

- Particles:

Amortised learning

Find an estimator for $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}[y_\theta]$, $y_{\boldsymbol{\theta}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, so that $\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$

- NN regression: $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \mathrm{NN}_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(\boldsymbol{x})$

$$\boldsymbol{\gamma}_t(\boldsymbol{\theta}) \leftarrow \boldsymbol{\gamma}_{t-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left\| y_{\boldsymbol{\theta},n} - \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}_n) \right\|_2^2$$

- Assume a flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \sum_k \alpha_k \psi_i(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x})$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \| y_{\boldsymbol{\theta},n} - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \|_2^2 + \lambda \| \boldsymbol{\alpha} \|_{\mathcal{H}}^2$$

- Particles:
  - Let $\boldsymbol{z}' = S_{\boldsymbol{\gamma}}(\boldsymbol{x}, \epsilon)$, where ϵ is a noise source

Amortised learning

Find an estimator for $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}[y_\theta]$, $y_{\boldsymbol{\theta}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, so that $\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$

- NN regression: $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \mathrm{NN}_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(\boldsymbol{x})$

$$\boldsymbol{\gamma}_t(\boldsymbol{\theta}) \leftarrow \boldsymbol{\gamma}_{t-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left\| y_{\boldsymbol{\theta},n} - \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}_n) \right\|_2^2$$

- Assume a flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \sum_k \alpha_k \psi_i(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x})$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \| y_{\boldsymbol{\theta},n} - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \|_2^2 + \lambda \| \boldsymbol{\alpha} \|_{\mathcal{H}}^2$$

- Particles:
  - Let $\boldsymbol{z}' = S_{\boldsymbol{\gamma}}(\boldsymbol{x}, \epsilon)$, where ϵ is a noise source
  - train $\boldsymbol{\gamma}$ so that $\mathbb{E}_{\boldsymbol{z}'|\boldsymbol{x}}[\log p_{\boldsymbol{\theta}_t}(\boldsymbol{z}', \boldsymbol{x})] = \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})$

Amortised learning

Find an estimator for $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}[y_\theta]$, $y_{\boldsymbol{\theta}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, so that $\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$

- NN regression: $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \mathrm{NN}_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(\boldsymbol{x})$

$$\boldsymbol{\gamma}_t(\boldsymbol{\theta}) \leftarrow \boldsymbol{\gamma}_{t-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left\| y_{\boldsymbol{\theta},n} - \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}_n) \right\|_2^2$$

- Assume a flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \sum_k \alpha_k \psi_i(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x})$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \|y_{\boldsymbol{\theta},n} - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x})\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_{\mathcal{H}}^2$$

- Particles:
  - Let $\boldsymbol{z}' = S_{\boldsymbol{\gamma}}(\boldsymbol{x}, \epsilon)$, where ϵ is a noise source
  - train $\boldsymbol{\gamma}$ so that $\mathbb{E}_{\boldsymbol{z}'|\boldsymbol{x}}[\log p_{\boldsymbol{\theta}_t}(\boldsymbol{z}', \boldsymbol{x})] = \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})$

$$\min_{\boldsymbol{\gamma}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z},\boldsymbol{x})} \left[y_{\boldsymbol{\theta},n} - \mathbb{E}_{\boldsymbol{z}'|\boldsymbol{x}} \left[\log p_{\boldsymbol{\theta}_t}(\boldsymbol{z}', \boldsymbol{x}) \right] \right]$$

Amortised learning

Find an estimator for $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}[y_\theta]$, $y_{\boldsymbol{\theta}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, so that $\hat{\Delta}_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$

- NN regression: $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \mathrm{NN}_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(\boldsymbol{x})$

$$\boldsymbol{\gamma}_t(\boldsymbol{\theta}) \leftarrow \boldsymbol{\gamma}_{t-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left\| y_{\boldsymbol{\theta},n} - \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}_n) \right\|_2^2$$

- Assume a flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \sum_k \alpha_k \psi_i(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x})$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \|y_{\boldsymbol{\theta},n} - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x})\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_{\mathcal{H}}^2$$

- Particles:
  - Let $\boldsymbol{z}' = S_{\boldsymbol{\gamma}}(\boldsymbol{x}, \epsilon)$, where ϵ is a noise source
  - train $\boldsymbol{\gamma}$ so that $\mathbb{E}_{\boldsymbol{z}'|\boldsymbol{x}}[\log p_{\boldsymbol{\theta}_t}(\boldsymbol{z}', \boldsymbol{x})] = \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})$

$$\min_{\boldsymbol{\gamma}} \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z},\boldsymbol{x})} \left[y_{\boldsymbol{\theta},n} - \mathbb{E}_{\boldsymbol{z}'|\boldsymbol{x}} \left[\log p_{\boldsymbol{\theta}_t}(\boldsymbol{z}', \boldsymbol{x}) \right] \right]$$

  - But no guarantee with $\nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$ as trained on $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x})$

Amortised learning with nonlinear regression

Flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[y_\theta\right]$

Amortised learning with nonlinear regression

Flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[y_\theta\right]$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \left\| \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha} \right\|_2^2, \quad \boldsymbol{x}_n, \boldsymbol{z}_n \sim p_{\boldsymbol{\theta}_t}$$

Amortised learning with nonlinear regression

Flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[y_\theta\right]$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \left\| \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha} \right\|_2^2, \quad \boldsymbol{x}_n, \boldsymbol{z}_n \sim p_{\boldsymbol{\theta}_t}$$

- Given $\boldsymbol{x}^* \in p^*$, the estimator outputs

$$\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} \boldsymbol{\psi}(\boldsymbol{x}^*)$$

Amortised learning with nonlinear regression

Flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[y_{\boldsymbol{\theta}}\right]$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \left\| \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha} \right\|_2^2, \quad \boldsymbol{x}_n, \boldsymbol{z}_n \sim p_{\boldsymbol{\theta}_t}$$

- Given $\boldsymbol{x}^* \in p^*$, the estimator outputs

$$\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} \boldsymbol{\psi}(\boldsymbol{x}^*)$$

where

$$\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}} \boldsymbol{\Psi}^{\intercal} \left(\boldsymbol{\Psi} \boldsymbol{\Psi}^{\intercal} + \lambda \mathbf{I} \right)^{-1} \quad \mathbf{Y}_{\boldsymbol{\theta}} = \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_1, \boldsymbol{x}_1), \ldots, \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n) \right]^{\intercal} \quad \boldsymbol{\Psi} = \left[\boldsymbol{\psi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\psi}(\boldsymbol{x}_n) \right]$$

Amortised learning with nonlinear regression

Flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[y_{\boldsymbol{\theta}}\right]$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \left\| \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha} \right\|_2^2, \quad \boldsymbol{x}_n, \boldsymbol{z}_n \sim p_{\boldsymbol{\theta}_t}$$

- Given $\boldsymbol{x}^* \in p^*$, the estimator outputs

  $$\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} \boldsymbol{\psi}(\boldsymbol{x}^*)$$

  where

  $$\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}} \boldsymbol{\Psi}^{\intercal} \left(\boldsymbol{\Psi} \boldsymbol{\Psi}^{\intercal} + \lambda \mathbf{I} \right)^{-1} \quad \mathbf{Y}_{\boldsymbol{\theta}} = \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_1, \boldsymbol{x}_1), \ldots, \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n) \right]^{\intercal} \quad \boldsymbol{\Psi} = \left[\boldsymbol{\psi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\psi}(\boldsymbol{x}_n) \right]$$

- $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathbf{Y}_{\boldsymbol{\theta}} \boldsymbol{\Psi}^{\intercal} \left(\boldsymbol{\Psi} \boldsymbol{\Psi}^{\intercal} + \lambda \mathbf{I} \right)^{-1} \psi(\boldsymbol{x}^*)$

Amortised learning with nonlinear regression

Flexible $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[y_{\boldsymbol{\theta}}\right]$

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{n=1}^{N} \left\| \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}) - \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha} \right\|_2^2, \quad \boldsymbol{x}_n, \boldsymbol{z}_n \sim p_{\boldsymbol{\theta}_t}$$

- Given $\boldsymbol{x}^* \in p^*$, the estimator outputs

  $$\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} \boldsymbol{\psi}(\boldsymbol{x}^*)$$

  where

  $$\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}} \boldsymbol{\Psi}^{\intercal} \left(\boldsymbol{\Psi} \boldsymbol{\Psi}^{\intercal} + \lambda \mathbf{I} \right)^{-1} \quad \mathbf{Y}_{\boldsymbol{\theta}} = \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_1, \boldsymbol{x}_1), \ldots, \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n) \right]^{\intercal} \quad \boldsymbol{\Psi} = \left[\boldsymbol{\psi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\psi}(\boldsymbol{x}_n) \right]$$

- $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathbf{Y}_{\boldsymbol{\theta}} \boldsymbol{\Psi}^{\intercal} \left(\boldsymbol{\Psi} \boldsymbol{\Psi}^{\intercal} + \lambda \mathbf{I} \right)^{-1} \boldsymbol{\psi}(\boldsymbol{x}^*)$
  same as if we regressed to $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$ instead of $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}} \boldsymbol{\Psi}^{\intercal} \left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda \mathbf{I}\right)^{-1}$

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}}\boldsymbol{\Psi}^{\intercal}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda\mathbf{I}\right)^{-1}$
- Wake phase: update $\boldsymbol{\theta}$ according to $\nabla\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}}\boldsymbol{\Psi}^{\intercal}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda\mathbf{I}\right)^{-1}$
- Wake phase: update $\boldsymbol{\theta}$ according to $\nabla\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) = \nabla_{\boldsymbol{\theta}}\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)$

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}}\boldsymbol{\Psi}^{\intercal}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda\mathbf{I}\right)^{-1}$
- Wake phase: update $\boldsymbol{\theta}$ according to $\nabla\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) = \nabla_{\boldsymbol{\theta}}\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)$
  - $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathrm{Adam}\left[\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)\right]$

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}}\boldsymbol{\Psi}^{\intercal}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda\mathbf{I}\right)^{-1}$
- Wake phase: update $\boldsymbol{\theta}$ according to $\nabla\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) = \nabla_{\boldsymbol{\theta}}\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)$
  - $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathrm{Adam}\left[\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)\right]$

Assumptions:

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}}\boldsymbol{\Psi}^{\intercal}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda\mathbf{I}\right)^{-1}$
- Wake phase: update $\boldsymbol{\theta}$ according to $\nabla\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) = \nabla_{\boldsymbol{\theta}}\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)$
  - $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathrm{Adam}\left[\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)\right]$

Assumptions:

- easy to sample from $p_{\boldsymbol{\theta}_t}(\boldsymbol{z},\boldsymbol{x})$

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}}\boldsymbol{\Psi}^{\intercal}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda\mathbf{I}\right)^{-1}$
- Wake phase: update $\boldsymbol{\theta}$ according to $\nabla\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) = \nabla_{\boldsymbol{\theta}}\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)$
  - $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathrm{Adam}\left[\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)\right]$

Assumptions:

- easy to sample from $p_{\boldsymbol{\theta}_t}(\boldsymbol{z},\boldsymbol{x})$
- can evaluate $\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})$

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}}\boldsymbol{\Psi}^{\intercal}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda\mathbf{I}\right)^{-1}$
- Wake phase: update $\boldsymbol{\theta}$ according to $\nabla\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) = \nabla_{\boldsymbol{\theta}}\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)$
  - $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathrm{Adam}\left[\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)\right]$

Assumptions:

- easy to sample from $p_{\boldsymbol{\theta}_t}(\boldsymbol{z},\boldsymbol{x})$
- can evaluate $\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})$

Non-assumptions:

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}}\boldsymbol{\Psi}^{\intercal}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda\mathbf{I}\right)^{-1}$
- Wake phase: update $\boldsymbol{\theta}$ according to $\nabla\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) = \nabla_{\boldsymbol{\theta}}\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)$
  - $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathrm{Adam}\left[\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)\right]$

Assumptions:

- easy to sample from $p_{\boldsymbol{\theta}_t}(\boldsymbol{z},\boldsymbol{x})$
- can evaluate $\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})$

Non-assumptions:

- **posterior**

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}}\boldsymbol{\Psi}^{\intercal}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda\mathbf{I}\right)^{-1}$
- Wake phase: update $\boldsymbol{\theta}$ according to $\nabla\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) = \nabla_{\boldsymbol{\theta}}\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)$
  - $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathrm{Adam}\left[\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)\right]$

Assumptions:

- easy to sample from $p_{\boldsymbol{\theta}_t}(\boldsymbol{z},\boldsymbol{x})$
- can evaluate $\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})$

Non-assumptions:

- **posterior**
- **support or domain** of latent

Amortimsed learning by wake sleep (ALWS)

- Sleep phase, regression to find $\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}_t}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\boldsymbol{z}_n, \boldsymbol{x}_n \sim p_{\boldsymbol{\theta}_t}$
  - evaluate $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_n, \boldsymbol{x}_n)$
  - $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\gamma}} = \mathbf{Y}_{\boldsymbol{\theta}}\boldsymbol{\Psi}^{\intercal}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{\intercal} + \lambda\mathbf{I}\right)^{-1}$
- Wake phase: update $\boldsymbol{\theta}$ according to $\nabla\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^*)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})\right]$
  - $\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*) = \nabla_{\boldsymbol{\theta}}\hat{J}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)$
  - $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathrm{Adam}\left[\hat{\Delta}_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\boldsymbol{x}^*)\right]$

Assumptions:

- easy to sample from $p_{\boldsymbol{\theta}_t}(\boldsymbol{z},\boldsymbol{x})$
- can evaluate $\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})$

Non-assumptions:

- **posterior**
- **support or domain** of latent
- **structure** of $p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x})$

Experiment I: gradient estimation

Generative model

$$z_1, z_2 \sim \mathcal{N}(0, 1),\ x|\boldsymbol{z} \sim \mathcal{N}(\text{softplus}(\boldsymbol{b} \cdot \boldsymbol{z}) - \|\boldsymbol{b}\|_2^2, \sigma_x^2)$$

Experiment I: gradient estimation

Generative model

$$z_1, z_2 \sim \mathcal{N}(0, 1),\ x|\boldsymbol{z} \sim \mathcal{N}(\text{softplus}(\boldsymbol{b} \cdot \boldsymbol{z}) - \|\boldsymbol{b}\|_2^2, \sigma_x^2)$$

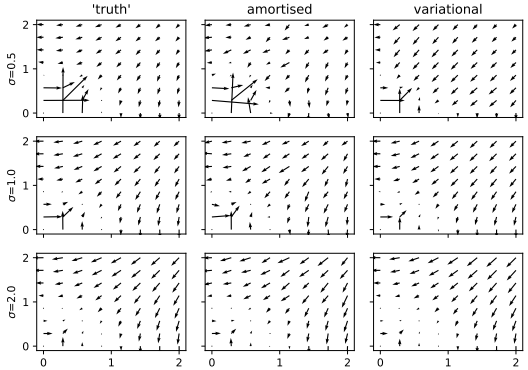
Task: estimate $\nabla_{\boldsymbol{b}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ for different $\boldsymbol{b}$ and σ

Experiment I: gradient estimation

Generative model

$$z_1, z_2 \sim \mathcal{N}(0, 1),\ x|\boldsymbol{z} \sim \mathcal{N}(\text{softplus}(\boldsymbol{b} \cdot \boldsymbol{z}) - \|\boldsymbol{b}\|_2^2, \sigma_x^2)$$

Task: estimate $\nabla_{\boldsymbol{b}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ for different $\boldsymbol{b}$ and σ



Experiment II: non-Euclidean $\boldsymbol{z}$

Model:

$$\boldsymbol{z} = [\cos(a), \sin(a)], \quad p(a) = \mathcal{U}(a|(-\pi, \pi)), \quad p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\mathrm{NN}_{\boldsymbol{w}}(\boldsymbol{z}), \sigma_x^2 \mathbf{I})$$

Experiment II: non-Euclidean $\boldsymbol{z}$

Model:

$$\boldsymbol{z} = [\cos(a), \sin(a)], \quad p(a) = \mathcal{U}(a|(-\pi, \pi)), \quad p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\mathrm{NN}_{\boldsymbol{w}}(\boldsymbol{z}), \sigma_x^2 \mathbf{I})$$

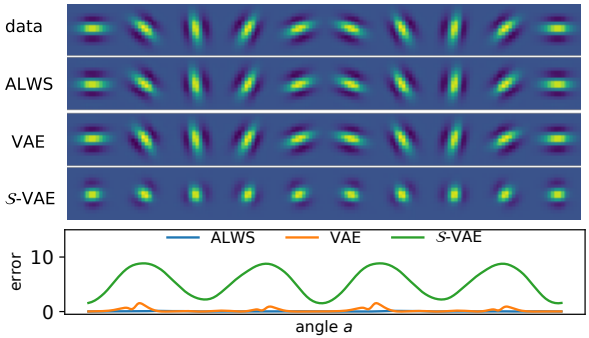
Task: generate Gabor filters of uniformly distributed orientations (no special reparameterisation)

Experiment II: non-Euclidean $\boldsymbol{z}$

Model:

$$\boldsymbol{z} = [\cos(a), \sin(a)], \quad p(a) = \mathcal{U}(a|(-\pi, \pi)), \quad p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\mathrm{NN}_{\boldsymbol{w}}(\boldsymbol{z}), \sigma_x^2 \mathbf{I})$$

Task: generate Gabor filters of uniformly distributed orientations (no special reparameterisation)



Experiment III: hierarchical models

Model:

$$
\begin{aligned}
p(\boldsymbol{z}_1) &= \mathrm{Cat}(\boldsymbol{z}_1|\boldsymbol{m}) \\
p(\boldsymbol{z}_2|\boldsymbol{z}_1 = k) &= \mathcal{N}(\boldsymbol{z}_2|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
p(\boldsymbol{x}|\boldsymbol{z}_2) &= \mathcal{N}(\boldsymbol{x}|\mathsf{NN}_{\boldsymbol{w}}(\boldsymbol{z}_2), \boldsymbol{\Sigma}_x)
\end{aligned}
$$

Experiment III: hierarchical models

Model:

$$
\begin{aligned}
p(\boldsymbol{z}_1) &= \mathrm{Cat}(\boldsymbol{z}_1|\boldsymbol{m}) \\
p(\boldsymbol{z}_2|\boldsymbol{z}_1 = k) &= \mathcal{N}(\boldsymbol{z}_2|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
p(\boldsymbol{x}|\boldsymbol{z}_2) &= \mathcal{N}(\boldsymbol{x}|\mathsf{NN}_{\boldsymbol{w}}(\boldsymbol{z}_2), \boldsymbol{\Sigma}_x)
\end{aligned}
$$

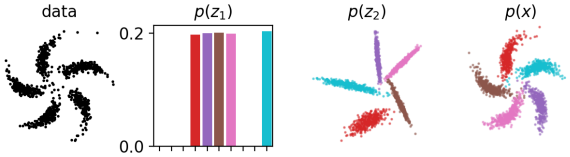
Parameters are penalised according to suitable priors

Experiment III: hierarchical models

Model:

$$
\begin{aligned}
p(\boldsymbol{z}_1) &= \mathrm{Cat}(\boldsymbol{z}_1|\boldsymbol{m}) \\
p(\boldsymbol{z}_2|\boldsymbol{z}_1 = k) &= \mathcal{N}(\boldsymbol{z}_2|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
p(\boldsymbol{x}|\boldsymbol{z}_2) &= \mathcal{N}(\boldsymbol{x}|\mathsf{NN}_{\boldsymbol{w}}(\boldsymbol{z}_2), \boldsymbol{\Sigma}_x)
\end{aligned}
$$

Parameters are penalised according to suitable priors

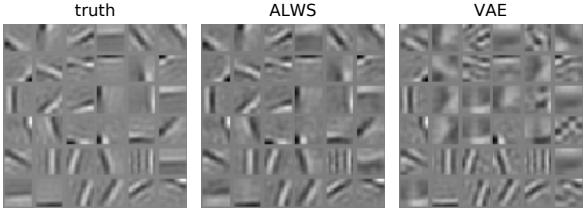


Experiment IV: feature identification

Linear noisy ICA Model:

$$p(z_i) = \mathrm{Lap}(z_i|0,1)$$
$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\mathbf{W}\boldsymbol{z}, \sigma^2 I)$$

Set up true $\mathbf{W}$,recover from data, compare with Laplace-VAE



Experiment V.1: noisy Hodgkin-Huxley model

$$
\begin{aligned}
C_m \dot{V}(t) &= -g_l[V(t) - E_l] - \bar{g}_N m^3(t) h(t)[V(t) - E_N] - \bar{g}_K n^4(t)[V(t) - E_K] + I_{in}(t) + \epsilon(t) \\
&= \alpha_e(V(t))[1 - e(t)] - \beta_e(V(t)) e(t), \quad e \in \{m, h, n\}
\end{aligned}
$$

Experiment V.1: noisy Hodgkin-Huxley model

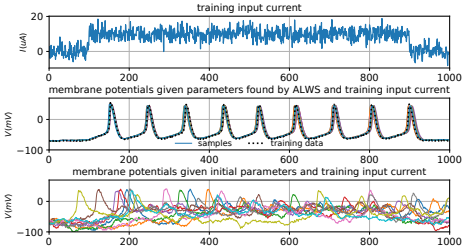
$$
\begin{aligned}
C_m \dot{V}(t) &= -g_l[V(t) - E_l] - \bar{g}_N m^3(t) h(t)[V(t) - E_N] - \bar{g}_K n^4(t)[V(t) - E_K] + I_{in}(t) + \epsilon(t) \\
&= \alpha_e(V(t))[1 - e(t)] - \beta_e(V(t)) e(t), \quad e \in \{m, h, n\}
\end{aligned}
$$

Discretised in time with $\delta t = 0.05ms$

Experiment V.1: noisy Hodgkin-Huxley model

$$C_m \dot{V}(t) = -g_l[V(t) - E_l] - \bar{g}_N m^3(t) h(t)[V(t) - E_N] - \bar{g}_K n^4(t)[V(t) - E_K] + I_{in}(t) + \epsilon(t)$$
$$= \alpha_e(V(t))[1 - e(t)] - \beta_e(V(t))e(t), \quad e \in \{m, h, n\}$$

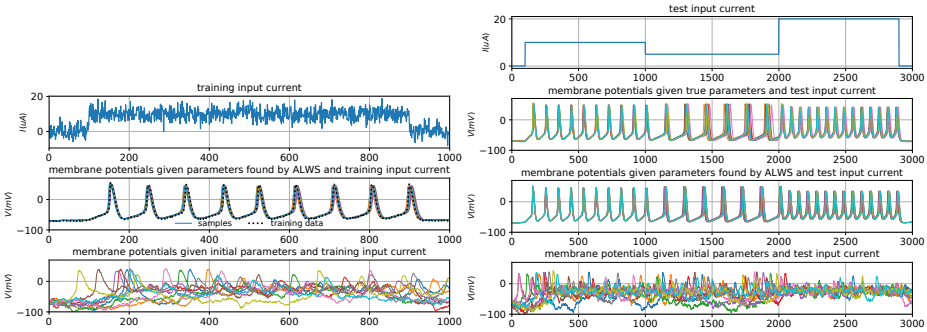
Discretised in time with $\delta t = 0.05ms$



Experiment V.1: noisy Hodgkin-Huxley model

$$C_m\dot{V}(t) = -g_l[V(t) - E_l] - \bar{g}_N m^3(t)h(t)[V(t) - E_N] - \bar{g}_K n^4(t)[V(t) - E_K] + I_{in}(t) + \epsilon(t)$$
$$= \alpha_e(V(t))[1 - e(t)] - \beta_e(V(t))e(t), \quad e \in \{m, h, n\}$$

Discretised in time with $\delta t = 0.05ms$



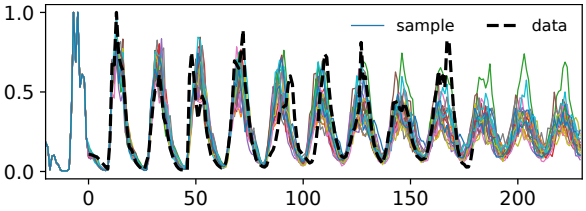
Experiment V.2: ecology data for blowfly population size

Model

$$\tau \sim \text{Categorical}(\boldsymbol{m}), \tau \in \{1, \ldots, 20\}, \quad e_t \sim \text{Gamma}(\frac{1}{\sigma_p^2}, \sigma_p^2), \quad \epsilon_t \sim \text{Gamma}(\frac{1}{\sigma_d^2}, \sigma_d^2),$$

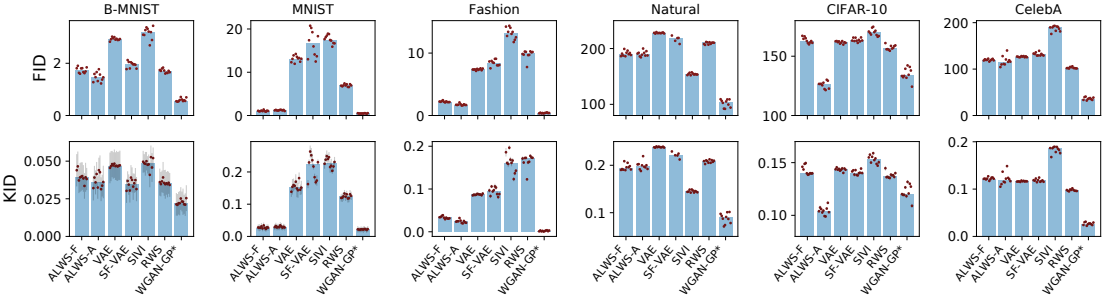
$$z_t = P x_{t-\tau} \exp(-\frac{x_{t-\tau}}{N_0}) + x_t \exp(-\delta \epsilon_t), \quad p(x_t | z_t) = \text{LogNormal}(\log(z_t), \sigma_n^2)$$

Categorical is a relaxation of a discrete parameter for time lag, $x_{-20:0}$ modelled as parameters



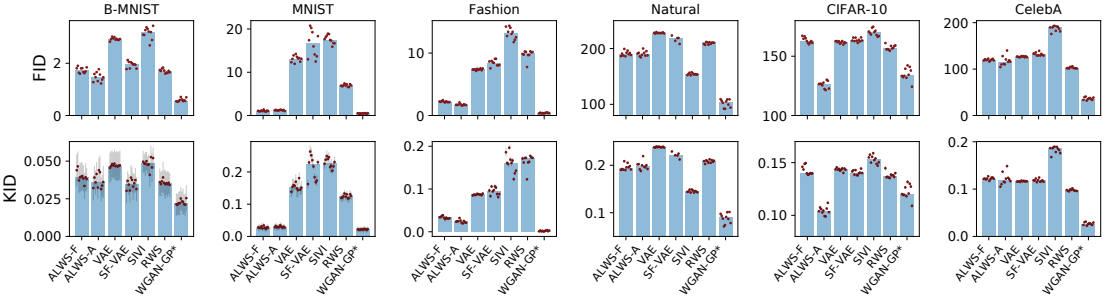
Experiment VI: image quality

Standard benchmark for learning generative models. Quality measured as distributional discrepancies



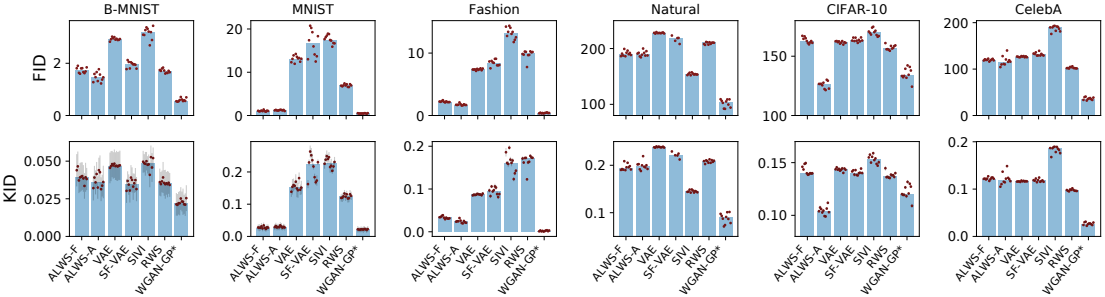
Experiment VI: image quality

Standard benchmark for learning generative models. Quality measured as distributional discrepancies
Compare with: VAE, VAE+flow, VAE+implicit posterior, reweighted wake sleep



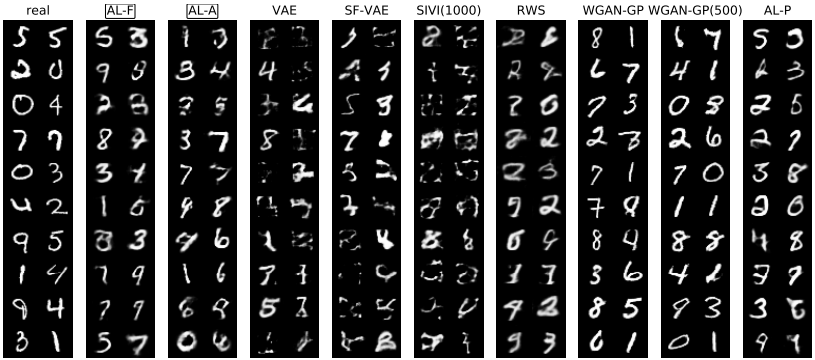
Experiment VI: image quality

Standard benchmark for learning generative models. Quality measured as distributional discrepancies
Compare with: VAE, VAE+flow, VAE+implicit posterior, reweighted wake sleep
On datasets : B-MNIST, MNIST, F-MNIST, Natural, CIFAR-10, CelebA

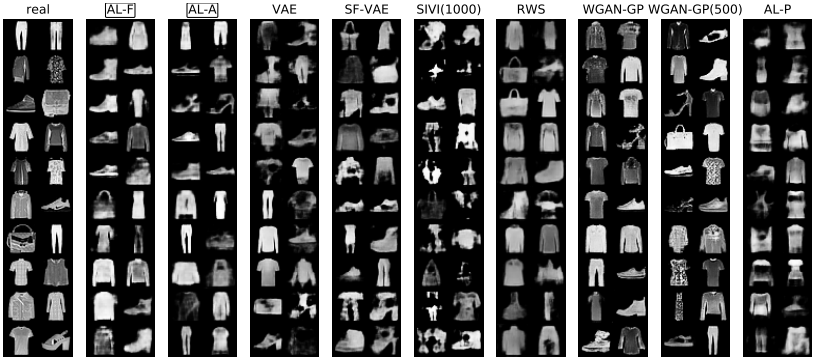


Samples: binarised MNSIT

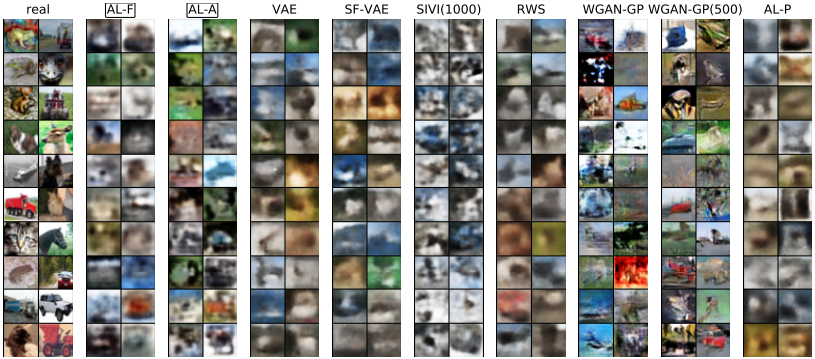
Samples: (original) MNSIT



Samples: Fashion



Samples: CIFAR-10 (bad)



Samples: Celeb-A (bad)