1938: Variational Inference in Robotics

Introduction to Variational Inference in Probabilistic Models

Botond Cseke

Machine Learning Research Lab, Volkswagen AG

November 14, 2022

Learning via inference

Hierarchical Bayesian model for data

- latent variable model (often analytically intractable)

$$p_\theta(x|u) = \int p_\theta(x|z)\, p_\theta(z|u) dz$$

Learning

- maximum likelihood learning from data (i.i.d.)

$$\theta^* = \operatorname*{argmax}_\theta \sum_i \log p_\theta(x_i|u_i)$$

- learning via variational approximations / Bayesian inference

$$\begin{aligned} \log p_\theta(x_i|u_i) &= \max_{q(z)} L_\theta(q(z); x_i, u_i) \\ &\geq \max_{q(z)\in\mathcal{Q}} L_\theta(q(z); x_i, u_i) \quad \text{(typically)} \end{aligned}$$

Contents

Probabilistic Graphical Models

- representation of hierarchical Bayesian models
- inference in hierarchical Bayesian models

Inference as optimisation

- divergence measures for approximating densities
- approximate inference in PGMs
- inference as learning with auto-encoders

State Space Models

- state space models and auto-encoders
- exact inference/computation in SSMs
- inference as learning with SSM auto-encoders

Probabilistic models

Probabilistic models

- variables of the model $z_1, z_2, \ldots, z_n$
- joint probability density $p(z_1, z_2, \ldots, z_n)$

Inference

- marginals

$$p(z_I) = \sum_{\{z_j : j \notin I\}} p(z_1, z_2, \ldots, z_n)$$

- conditional densities

$$p(z_I | z_J) = p(z_I, z_J)/p(z_J)$$

Issues of interest

- representation: parameters, dependencies
- inference: how to perform inference, quality of approximations
- computation: complexity, efficiency

Representation - probabilistic graphical models

(e.g. Cowell et al., 2007)

Using dependence/independence relations to provide a sparse representation

- independence

$$p(z_i, z_j) = p(z_i)p(z_j) \quad \text{or} \quad p(z_i|z_j) = p(z_i)$$

Representation - probabilistic graphical models

(e.g. Cowell et al., 2007)

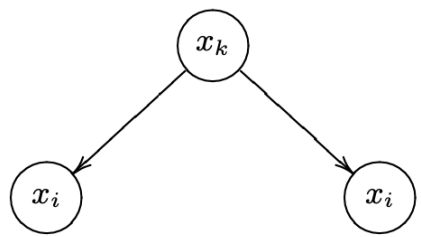
Using dependence/independence relations to provide a sparse representation

- independence

$$p(z_i, z_j) = p(z_i)p(z_j) \quad \text{or} \quad p(z_i|z_j) = p(z_i)$$

- conditional independence

$$p(z_i, z_j|z_k) = p(z_i|z_k)p(z_j|z_k) \quad \text{or} \quad p(z_i|z_j, z_k) = p(x_i|z_k)$$



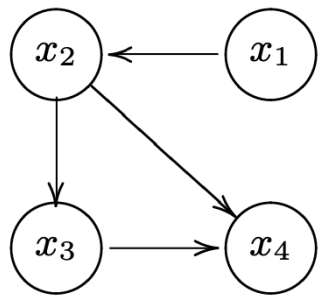
Representation - probabilistic graphical models

(e.g. Cowell et al., 2007)

Conditional independence relations between the variables $z_1, \ldots, z_n$ can be represented by separation properties on a graph.

- Directed PGM (Bayesian networks)
  - conditional independence relations (d-separation)
  - factorisation using chain rule and independence relations

$$p(z_1, \ldots, z_n) = \prod_k p(z_k | z_{\text{parents}(k)})$$



Representation - probabilistic graphical models

(e.g. Cowell et al., 2007)

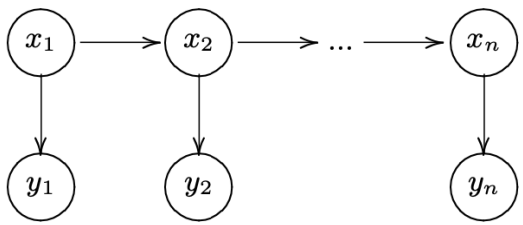
Conditional independence relations between the variables $z_1, \ldots, z_n$ can be represented by separation properties on a graph.

- Directed PGM (Bayesian networks)
  - conditional independence relations (d-separation)
  - factorisation using chain rule and independence relations

$$p(z_1, \ldots, z_n) = \prod_k p(z_k | z_{\text{parents}(k)})$$

- Example: Hidden Markov model, sequential models

$$p(z_1, \ldots, x_z) = p(x_1|z_1)p(z_1) \prod_{t=1}^{T-1} p(x_{k+1}|z_{k+1})p(z_{k+1}|z_k)$$



Contents

Probabilistic graphical models / practical example

(e.g. Cowell et al., 2007)

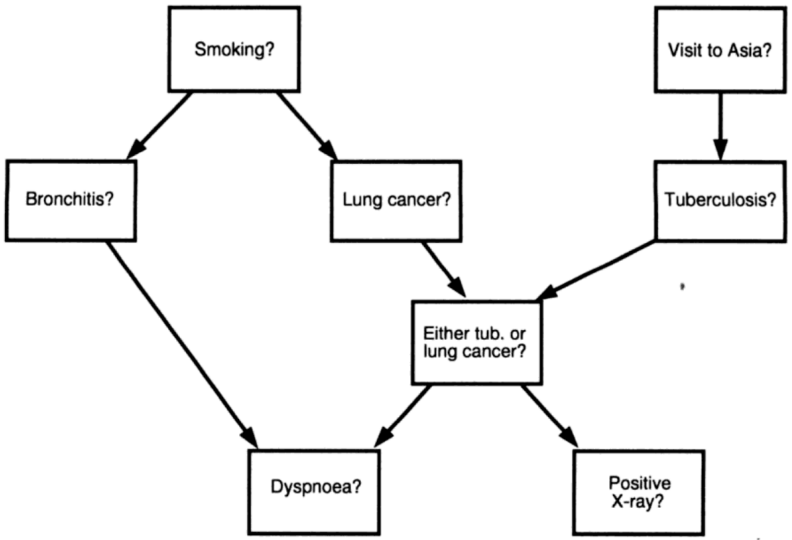


FIGURE 2.7. The ASIA network.

TABLE 2.3. Conditional probability specifications for the ASIA example.

A:	$p(a)$	$=$	0.01	L:	$p(l \mid s)$	$=$	0.1
					$p(l \mid \bar{s})$	$=$	0.01
B:	$p(b \mid s)$	$=$	0.6	S:	$p(s)$	$=$	0.5
	$p(b \mid \bar{s})$	$=$	0.3				
D:	$p(d \mid b, e)$	$=$	0.9	T:	$p(t \mid a)$	$=$	0.05
	$p(d \mid \bar{b}, e)$	$=$	0.7		$p(t \mid \bar{a})$	$=$	0.01
	$p(d \mid b, \bar{e})$	$=$	0.8				
	$p(d \mid \bar{b}, \bar{e})$	$=$	0.1				
E:	$p(e \mid l, t)$	$=$	1	X:	$p(x \mid e)$	$=$	0.98
	$p(e \mid \bar{l}, t)$	$=$	1		$p(x \mid \bar{e})$	$=$	0.05
	$p(e \mid l, \bar{t})$	$=$	1				
	$p(e \mid \bar{l}, \bar{t})$	$=$	0				

A simple practical example

- medical decision making by probabilistic inference
- learning the conditional probability tables from incomplete data
- iterative Bayes' rule organised as dynamic programming
- computational complexity exponential only in cluster size

Exact computation and sampling

(e.g. Bishop, 2006)

Exact computation

- models with (conditional) discrete variables: e.g. expert systems
- models with (conditional) Gaussians: e.g. Kalman filter/smoother

Approximation by sampling

- computations in the model $p(\bar{x}, z)$ are intractable (marginalisation, conditioning)
- sampling by Markov Chain Monte Carlo (Andrieu et al., 2003)

$$p(z|\bar{x}) = p_\infty(z)$$

$$p^{(i+1)}(z) = \int \mathcal{T}(z^{(i+1)}; z^{(i)})\, p^{(i)}(z) dz$$

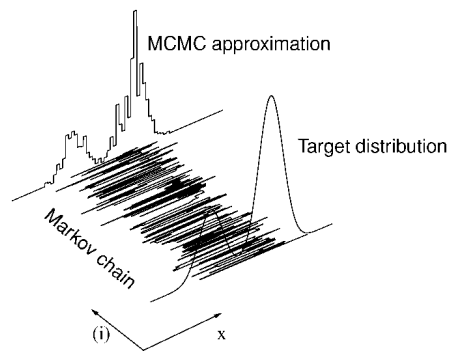
- variants: Metropolis-Hastings, Hamilton Monte Carlo, Gibbs Sampling, No-U-turn HMC, Langevin sampling

1. Initialise $x^{(0)}$.
2. For $i = 0$ to $N - 1$
   - Sample $u \sim \mathcal{U}_{[0,1]}$.
   - Sample $x^* \sim q(x^*|x^{(i)})$.
   - If $u < \mathcal{A}(x^{(i)}, x^*) = \min\left\{1, \frac{p(x^*)q(x^{(i)}|x^*)}{p(x^{(i)})q(x^*|x^{(i)})}\right\}$

     $x^{(i+1)} = x^*$

     else

     $x^{(i+1)} = x^{(i)}$



Contents

Probabilistic Graphical Models

- representation of hierarchical Bayesian models
- inference in hierarchical Bayesian models

Inference as optimisation

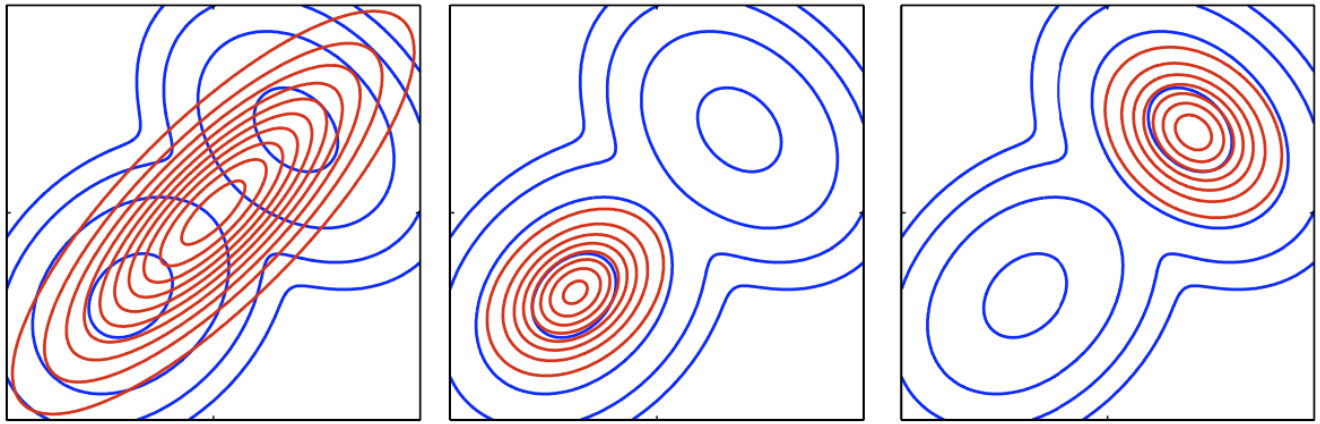
- divergence measures for approximating densities
- approximate inference in PGMs
- inference as learning with auto-encoders

State Space Models

- state space models and auto-encoders
- exact inference/computation in SSMs
- inference as learning with SSM auto-encoders

Inference as optimisation

(e.g. Bishop, 2006)



Optimise a KL divergence measure over a parametric class

$$\min_{q \in \mathcal{Q}} D[q(z)||p(z)], \quad \text{s.t.} \quad \mathcal{Q} = \{\exp\{\theta \cdot f(z)\}/Z(\theta) : \theta \in \Theta_0\}$$

Common choices for $D[\cdot\|\cdot]$

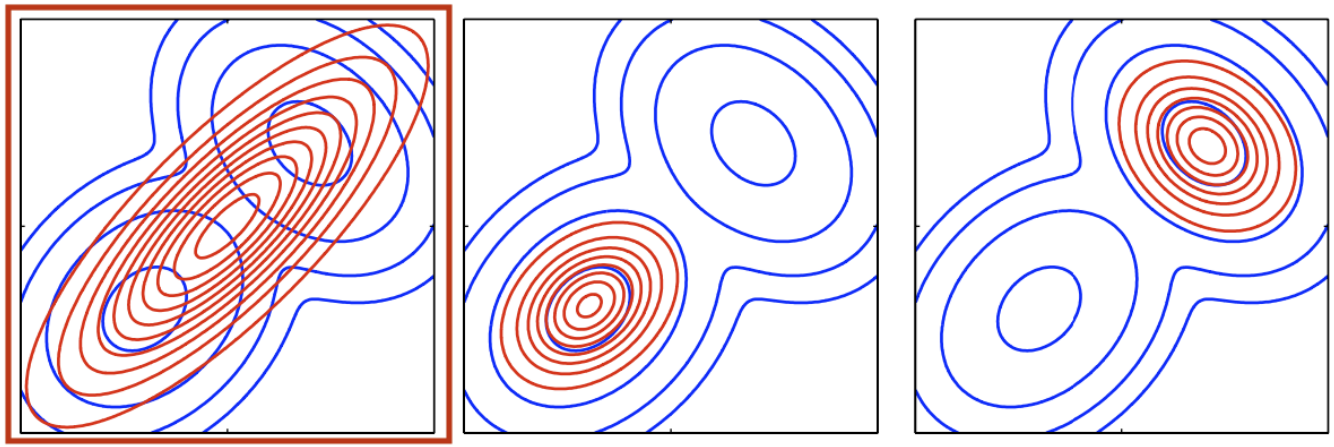
- moment matching Kullback-Leibler

$$D_{\mathrm{KL}}[p||q] = \int dz\, p(z) \log[p(z)/q(z)], \quad E_{q^*}[f(z)] = E_p[f(z)]$$

- variational Kullback-Leibler

$$D_{\mathrm{KL}}[q||p] = \int dz\, q(z) \log[q(z)/p(z)]$$

Inference as optimisation / moment matching



Optimise a KL divergence measure over a parametric class

$$\min_{\theta} D_{\mathrm{KL}}[p||q_\theta], \quad \text{s.t.} \quad q_\theta(z) = \exp\{\theta \cdot f(z)\}/Z(\theta),\ \theta \in \Theta_0$$

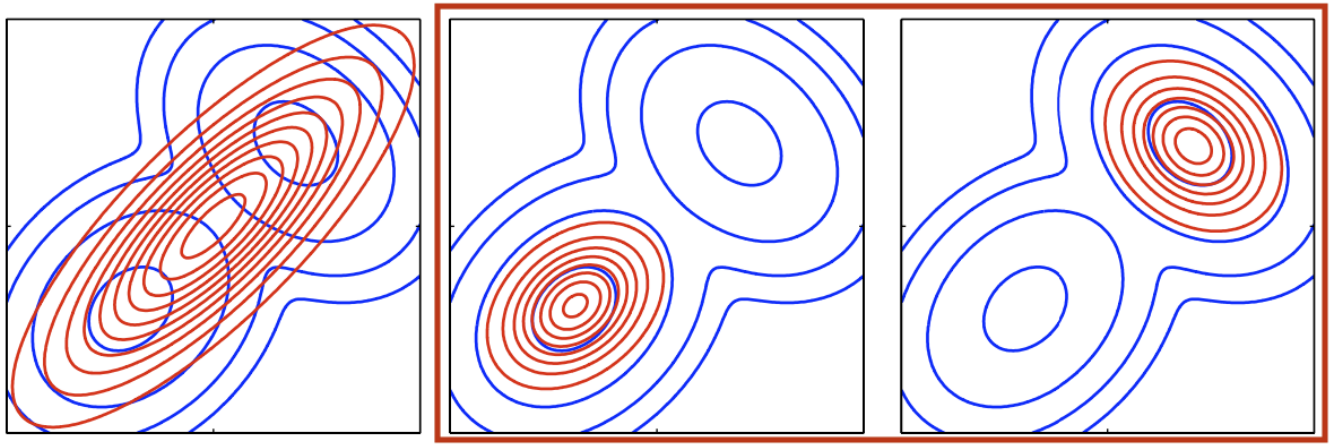
Details

$$D_{\mathrm{KL}}[p||q_\theta] = E_p[\log p(z)] - E_p[f(z)]\theta + \log Z(\theta), \quad \partial_\theta \log Z(\theta) = E_q[f(z)]$$

Maximum likelihood as divergence optimisation

$$D_{\mathrm{KL}}[\hat{p}||q_\theta] = E_{\hat{p}}[\hat{p}(z)] - \underbrace{E_{\hat{p}} \log q_\theta(z)]}_{\text{neg. log likelihood}}$$

Inference as optimisation / variational inference



Optimise a KL divergence measure over a parametric class

$$\min_{\theta} D_{\mathrm{KL}}[q_\theta||p], \quad \text{s.t.} \quad q_\theta(z) = \exp\{\theta \cdot f(z)\}/Z(\theta),\ \theta \in \Theta_0$$

Details

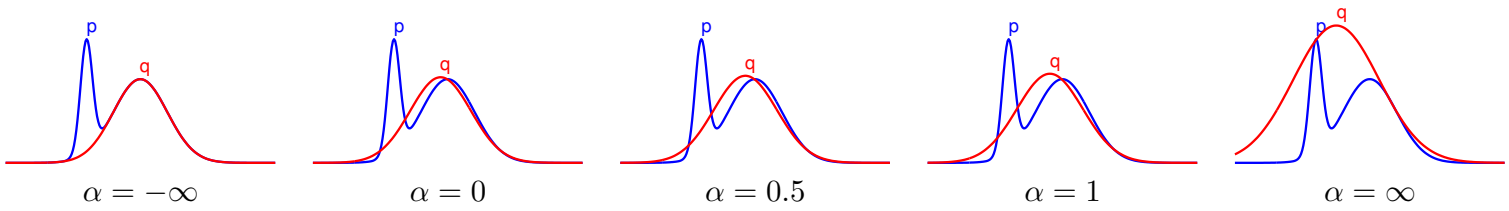
$$\begin{aligned} D_{\mathrm{KL}}[q_\theta||p] &= E_{q_\theta}[\log q_\theta(z)] - E_{q_\theta}[\log p(z)] \\ &= E_{q_\theta}[\log q_\theta(z)] - \sum_i E_{q_\theta}[\log p(z_i|z_{\mathrm{parents}(i)})] \quad (\geq -\log p(\bar{z})) \end{aligned}$$

Advantages

- the neg. entropy $E_{q_\theta}[\log q_\theta(z)]$ often comes in closed form
- $E_{q_\theta}[\log p(z)]$ allows for distributed computation

Inference as optimisation / α-divergences

(Minka, 2005)



α-divergence

$$D_\alpha[p||q] = \frac{1}{\alpha(1-\alpha)} \int dz \; \left[(1 - p(z)^\alpha q(z)^{1-\alpha}) + (\alpha p(z) + (1-\alpha)q(z))\right]$$

Special cases

$$D_{\alpha=0}[p||q] = D_{\mathrm{KL}}[q||p] \quad \text{and} \quad D_{\alpha=1}[p||q] = D_{\mathrm{KL}}[p||q]$$

The properties observed in this example are general, and can be derived from the formula for α-divergence. Start with the mode-seeking property for $\alpha \ll 0$. It happens because the valleys of p force the approximation downward. Looking at (3,4) for example, we see that $\alpha \leq 0$ emphasizes q to be small whenever p is small. These divergences are **zero-forcing** because $p(\mathbf{x}) = 0$ forces $q(\mathbf{x}) = 0$. In other words, they avoid "false positives," to an increasing degree as α gets more negative. This causes some parts of p to be excluded. The cost of excluding an $\mathbf{x}$, i.e. setting $q(\mathbf{x}) = 0$, is $p(\mathbf{x})/(1-\alpha)$. Therefore q will keep the areas of largest total mass, and exclude areas with small total mass.

When $\alpha \geq 1$, a different tendency happens. These divergences want to cover as much of p as possible. Following the terminology of Frey et al. (2000), these divergences are **inclusive** ($\alpha < 1$ are is **exclusive**). Inclusive divergences require $q > 0$ whenever $p > 0$, thus avoiding "false negatives." If two identical Gaussians are separated enough, an exclusive divergence prefers to represent only one of them, while an inclusive divergence prefers to stretch across both.

Inference as optimisation / sample based divergence minimisation

Information, Divergence and Risk for Binary Experiments

Mark D. Reid MARK.REID@ANU.EDU.AU
Robert C. Williamson BOB.WILLIAMSON@ANU.EDU.AU
Australian National University and NICTA
Canberra ACT 0200, Australia

Optimal Bounds between f-Divergences and Integral Probability Metrics

Rohit Agrawal [*1] **Thibaut Horel** [*2]

Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization

XuanLong Nguyen
SAMSI & Duke University

Martin J. Wainwright
UC Berkeley

Michael I. Jordan
UC Berkeley

Minimax Estimation of Maximum Mean Discrepancy with Radial Kernels

Ilya Tolstikhin
Department of Empirical Inference
MPI for Intelligent Systems
Tübingen 72076, Germany
ilya@tuebingen.mpg.de

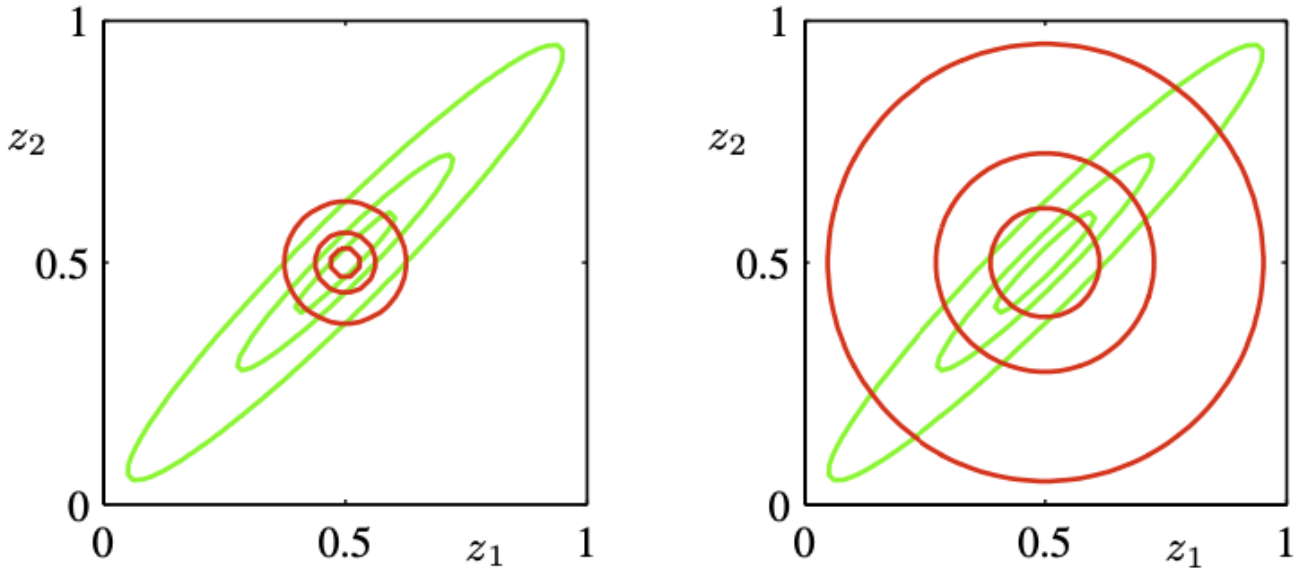
Bharath K. Sriperumbudur
Department of Statistics
Pennsylvania State University
University Park, PA 16802, USA
bks18@psu.edu

Bernhard Schölkopf
Department of Empirical Inference
MPI for Intelligent Systems
Tübingen 72076, Germany
bs@tuebingen.mpg.de

Contents

Factorised approximations

(Bishop, 2006)



Simplify approximation to differentiate groups variables

$$q_\theta(z) = \prod_i q_{\theta_i}(z_i)$$

Classical: fixed point coordinate descent KL-var

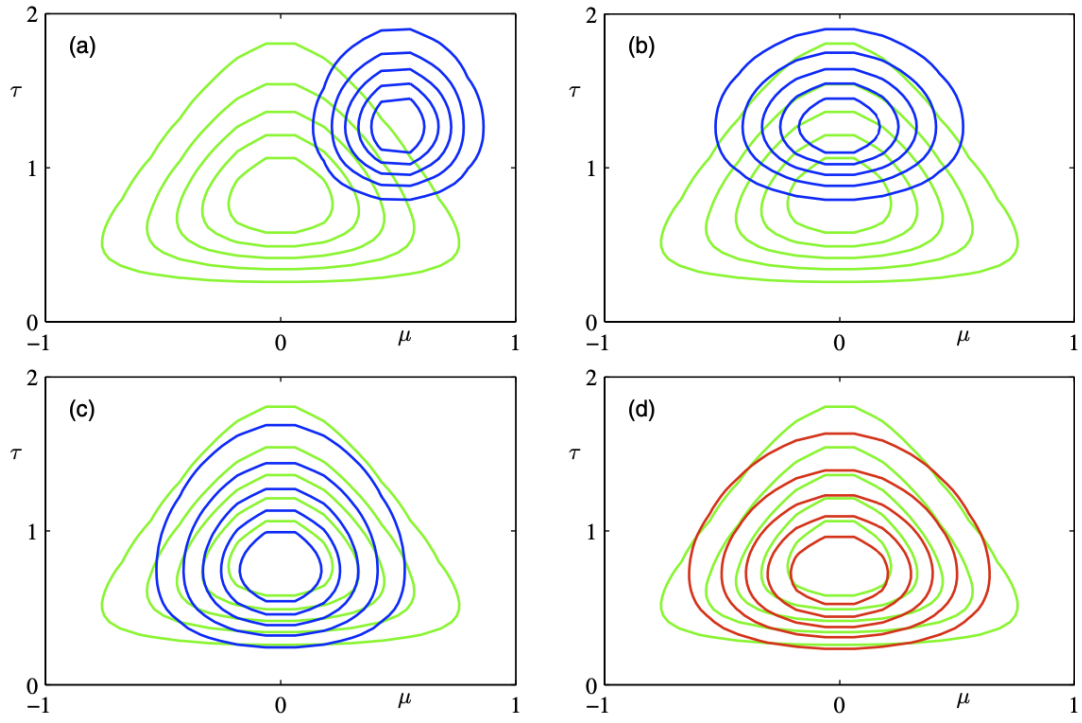
$$q_{\theta_i}(z_i)^{new} \propto \exp\{E_{\prod_{j\neq i} q_{\theta_j}}[\log p(z_i, z_{\setminus i})]\}$$

Modern: joint descent KL-var

$$\min_{\theta_1,\ldots,\theta_I} \sum_i E_{q_{\theta_i}}[\log q_{\theta_i}] - E_{\prod_i q_{\theta_i}}[\log p(z_1, \ldots, z_I)]$$

Factorised approximations / mean-variance example

(Bishop, 2006)



Hierarchical model for mean variance

$$p(x, \mu, \sigma^2) = \mathcal{N}(\mu; \mu_0, \sigma_0^2)\, \mathrm{InvGam}(\sigma^2; \alpha_0, \beta_0) \prod_i \mathcal{N}(x_i; \mu, \sigma^2)$$

Approximation

$$q(\mu, \sigma^2) = \mathrm{N}(\mu; \hat{\mu}, \hat{\sigma}^2)\, \mathrm{InvGam}(\sigma^2; \hat{\alpha}, \hat{\beta})$$

Black-box variational inference

Black Box Variational Inference

Rajesh Ranganath **Sean Gerrish** **David M. Blei**
Princeton University, 35 Olden St., Princeton, NJ 08540
{rajeshr,sgerrish,blei} AT cs.princeton.edu

General graphical models with structures approx.

$$p(x, z) = p(x|z)p(z), \quad p(z) = \prod_i p(z_i|z_{\text{parents}(i)})$$

Inference via optimisation with sampling and stochastic gradient

$$L(\theta) = -E_{q_\theta}[\log p(x, z) - \log q_\theta(z)]$$
$$\partial_\theta L(\theta) = E_{q_\theta}[\partial_\theta \log q_\theta(z)(\log p(x, z) - \log q_\theta(z))]$$

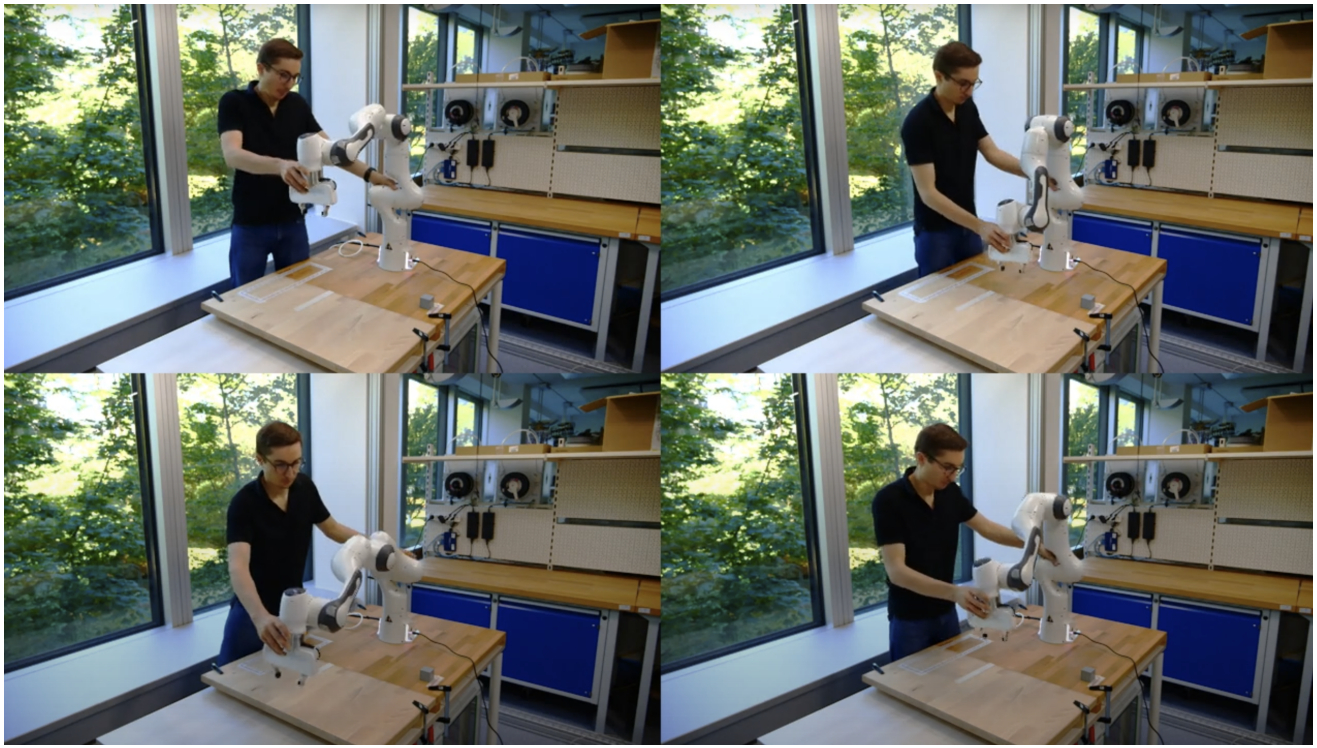
Advantages

- generic method for all type of graphical models
- scales well with data as it can deal with batch gradient
- allows for both classical and neural modelling

Application of VI in robot trajectory optimisation

Probabilistic Movement Primitives

(Paraschos et al., 2013)



Learning from examples

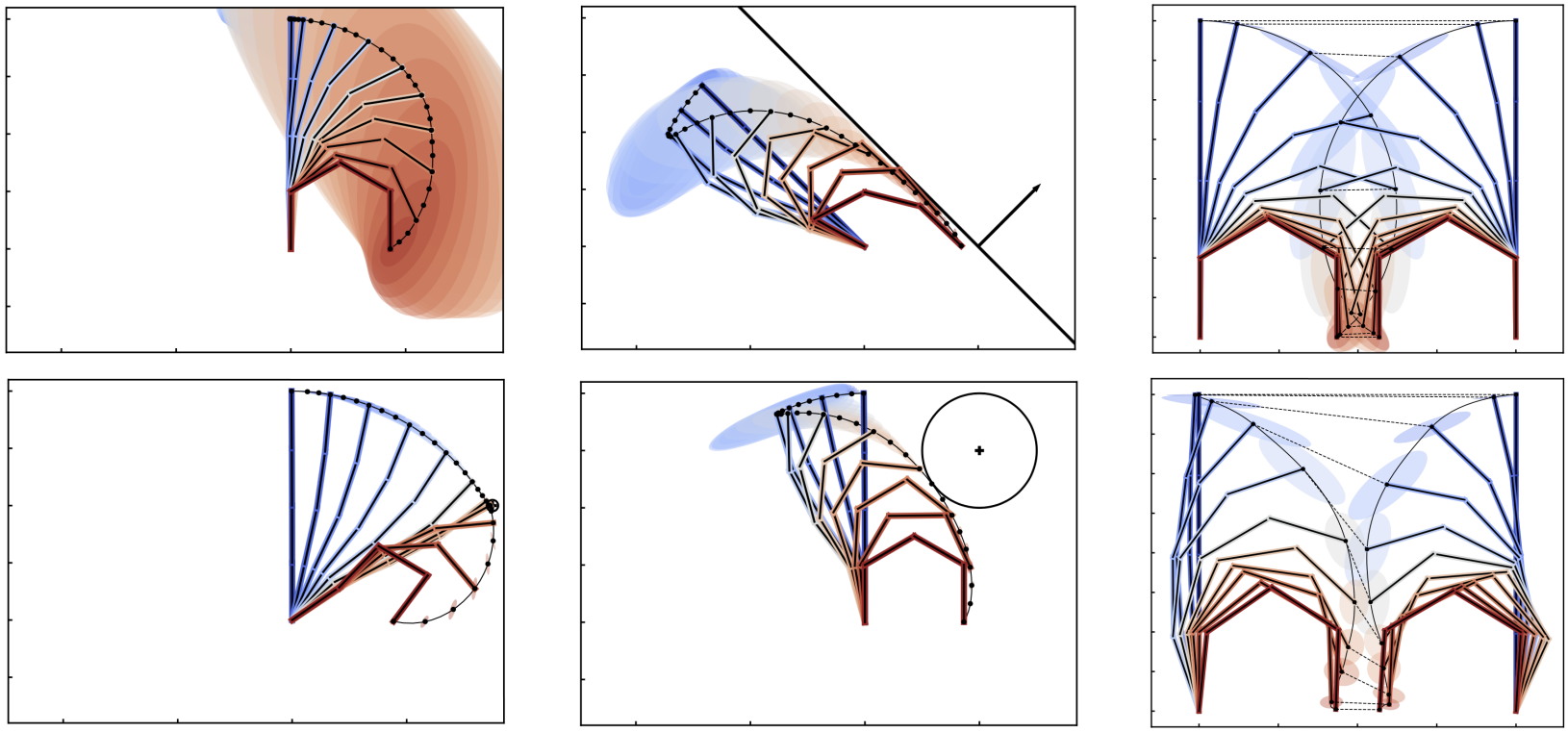
$$y(t) = w\phi(t) + \epsilon_t, \quad \phi(t)^i = \exp(-(t - t_i)^2/\sigma^2)$$

Inferring trajectory distribution / maximum likelihood

$$\max_\theta \sum_k \log p_\theta(y^k), \qquad p_\theta(y^k) = \int dw\, p_\theta(y^k|w)\, p_\theta(w)$$

Trajectory adaptation / Bayes hits the wall

(Frank et al., 2022)



Adaptation as VI with constraints instead of observations / Bayes updates

$$\min_{p} \; D_{\mathrm{KL}}[\,p(\boldsymbol{w}) \| \, p_0(\boldsymbol{w})]$$

$$\text{s.t} \;\; P_{\boldsymbol{w}}\big(c_{k,t}(\boldsymbol{w}) \leq d_{k,t}\big) \geq \alpha_{k,t} \quad \forall\, k,t \in \mathcal{T}_k,$$

Constraints for obstacle avoidance

$$c_t(w) = n^T x_t(w) - b, \qquad c_t(w) = ||x_t(w) - x_0||^2$$

[Show video: https://www.youtube.com/watch?v=ErdP7bA11v8]

Contents

VAE / Unsupervised Bayesian learning through amortisation

(Kingma and Welling, 2014)

Auto-Encoding Variational Bayes

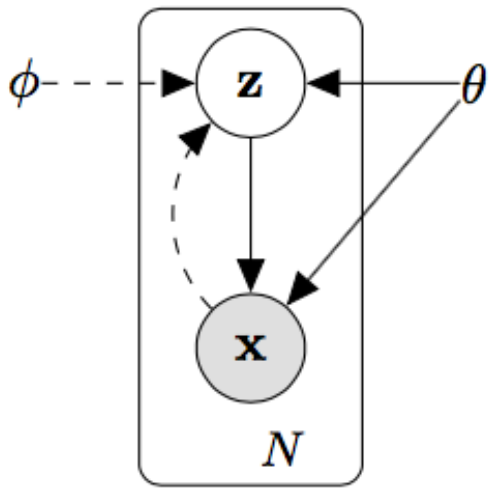
Diederik P. Kingma
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

- Bayesian unsupervised learning

$$p_\theta(x) = \int dz\, p_\theta(x|z)\, p_0(z)$$

- variational Bayes approximations for likelihood optimisation and posterior inference
- unifies VB with auto-encoders to simplify inference $\rightarrow$ scaling it to large data sets



VAE / bounding the likelihood function

Variational bound

$$\begin{aligned}
\log p_\theta(x_i) &= \log \int dz_i \; p(x_i|z_i)\, p_0(z_i) \\
&= \log \int dz_i \; p(x_i|z_i) \; \frac{p_0(z_i)}{q_i(z_i)} q_i(z_i) \\
&\geq \int dz_i \; q_i(z_i) \; \log \frac{p(x_i|z_i) p_0(z_i)}{q_i(z_i)} \\
&= \mathbb{E}_{q_i}[\log p_\theta(x_i|z_i)] - \mathbb{D}[q_i(z_i) \parallel p_0(z_i)] \quad \equiv L(\theta, q_i : x_i)
\end{aligned}$$

Properties

- for each x_i we have

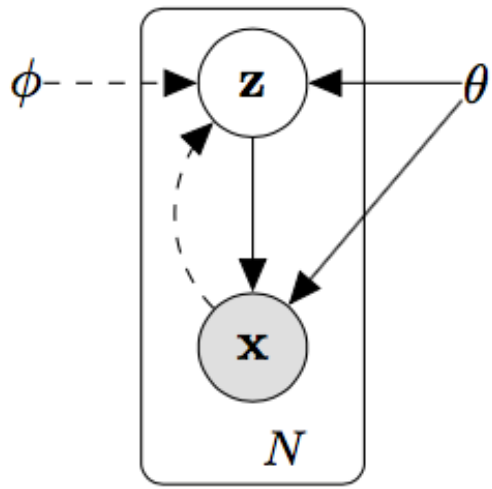
$$q_i^*(z_i) = \frac{p(x_i|z_i)\, p(z_i)}{p_\theta(x_i)} \quad \rightarrow \quad q_i^*(z_i) = p(z_i|x_i)$$

- tight bound

$$L(\theta, q_i^*; x_i) = \log p_\theta(x_i)$$

- quality of approximation

$$\log p_\theta(x_i) - L(\theta, q_i; x_i) = D[q_i(z_i) \parallel p(z_i|x_i)]$$



VAE / large scale data

Divergence bound

$$\log p_\theta(x_i) \geq L(\theta, q_i; x_i) \quad \left(= \mathbb{E}_{q_i}[\log p_\theta(x_i|z_i)] - \mathbb{D}[q_i(z_i) \parallel p_0(z_i)] \right)$$

VAE approach

- overall likelihood bound

$$\sum_i \log p_\theta(x_i) \geq \sum_i L(\theta, q_i; x_i)$$

- tight bound (many independent optimisation problems)

$$\max_\theta \sum_i \log p_\theta(x_i) \geq \max_\theta \sum_i \max_{q_i} L(\theta, q_i; x_i)$$

- amortisation and parameter sharing

$$q_i(z_i) \equiv q_\phi(z_i|x_i) \quad \text{e.g.} \quad q_\phi(z_i|x_i) = \mathcal{N}(z_i; \mu_\phi(x_i), \sigma_\phi(x_i)^2)$$

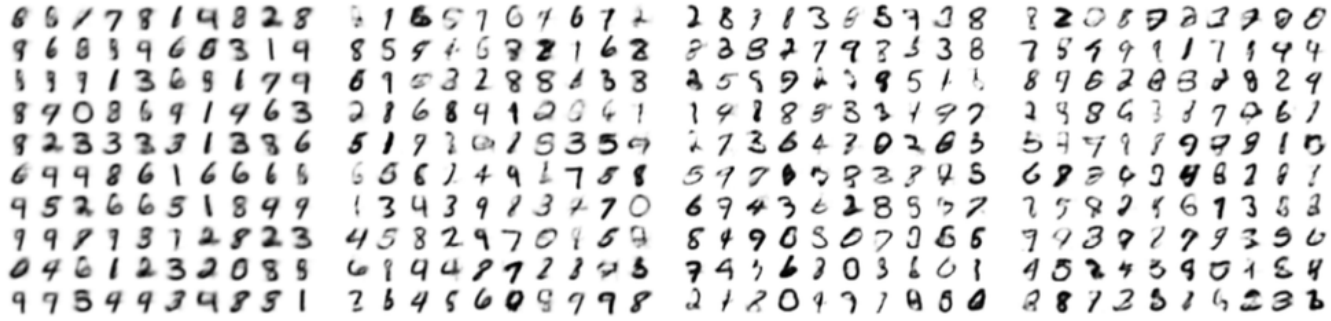
- SGD for

$$\max_\theta \max_\phi L(\theta, q_\phi(z_i; x_i); x_i)$$

- neural network parameterisation of Gaussians / reparameterisation

$$[\, \mu_\phi(x_i), \log \sigma_\phi(x_i)) \,] = \mathsf{NN}_\phi(x_i) \quad \text{and} \quad z_i^{(s)} = \mu_\phi(x_i) + \sigma_\phi(x_i)\, \epsilon_i^{(s)}$$

VAE / Experiments I



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Unsupervised learning of digits

- details of the model

$$p_0(z) = \mathcal{N}(z; 0, I_d), \quad \text{and} \quad p_\theta(x|z) = \mathcal{N}(x|\mu_\theta(z), \sigma_\phi(z)^2)$$

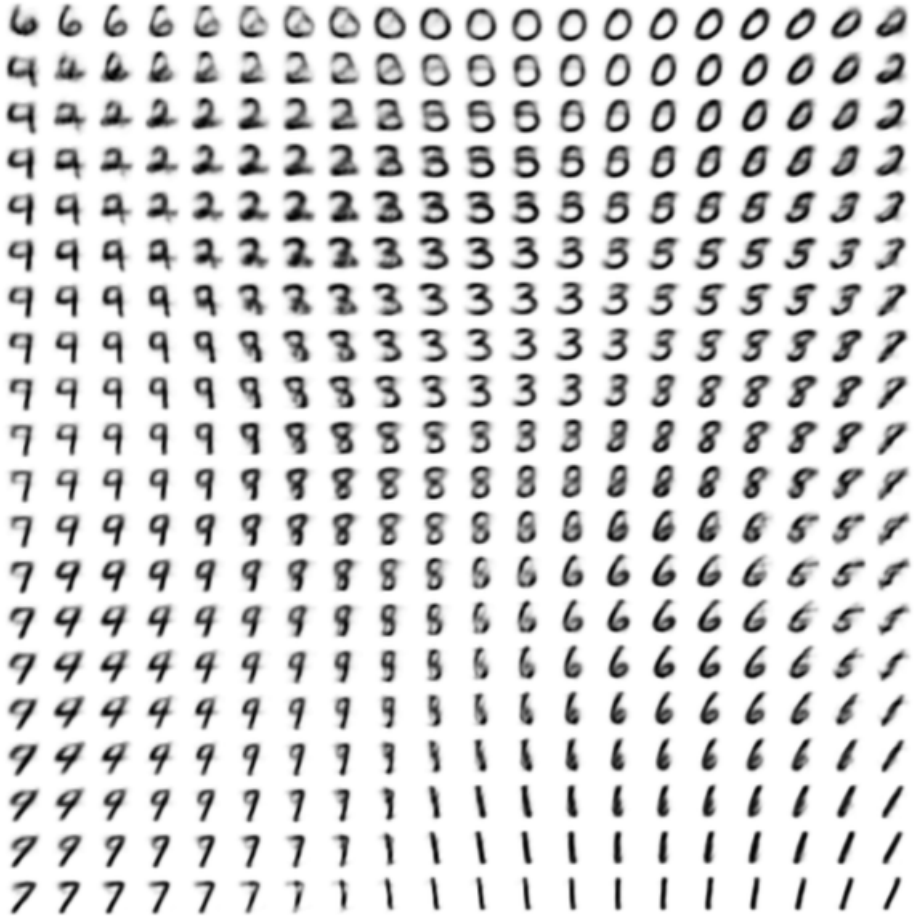
- details of the approximation

$$q_\phi(z_i|x_i) = \mathcal{N}(z_i; \mu_\phi(x_i), \sigma_\phi(x_i)^2) \quad \text{s.t.} \quad [\, \mu_\phi(x_i), \log \sigma_\phi(x_i)) \,] = \mathsf{NN}_\phi(x_i)$$

- training: SDG steps
  - sample $x_i \sim \hat{p}(x)$, sample $\epsilon_i^{(k)} \sim \mathcal{N}(0, I_d)$, $z_i^{(k)} = \mu_\phi(x_i) + \sigma_\phi(x_i)\, \epsilon_i^{(k)}$
  - approx of MC loss

$$\tilde{L}(\theta, \phi) = \frac{1}{N_x} \sum_i \left\{ \frac{1}{N_z} \sum_k \log p_\theta(x_i|z_i^{(k)}) - \underbrace{\mathbb{D}[q_\phi(z_i|x_i) \,\|\, \mathcal{N}(z_i; 0, I_d)]}_{\text{analytic}} \right\}$$

VAE / Experiments II



Structure of the latent space z

- fit a 2d latent space for z
- map 2d uniform grid $z_{i,j} = (i.j)/N, \quad i, j = 1, \ldots, N$ through normal inverse cdf
- sample / compute mean of $p_\theta(x|z_{i,j})$

β-VAE and friends

Objective

$$L(\theta, \phi; \hat{p}) = -\mathbb{E}_{\hat{p}(x)}\mathbb{E}_{q_\phi(z;x)}[\log p_\theta(x|z)] + \beta\, \mathbb{E}_{\hat{p}(x)}\mathbb{D}[q_\phi(z;x)||p_0(x)]$$

Annealing (Sønderby et al., 2016)

$$\beta^{(t)} = \begin{cases} 0, & \beta^{(t)} < t_{\text{start}} \\ (t - t_{\text{start}})/(t_{\text{end}} - t_{\text{start}}), & t_{\text{start}} < \beta^{(t)} \leq t_{\text{end}} \\ 1, & t_{\text{end}} < \beta^{(t)} \end{cases}$$

β-VAE (Higgins et al., 2017)

$\beta > 1$ results in better disentanglement, custom utility function to adapt β

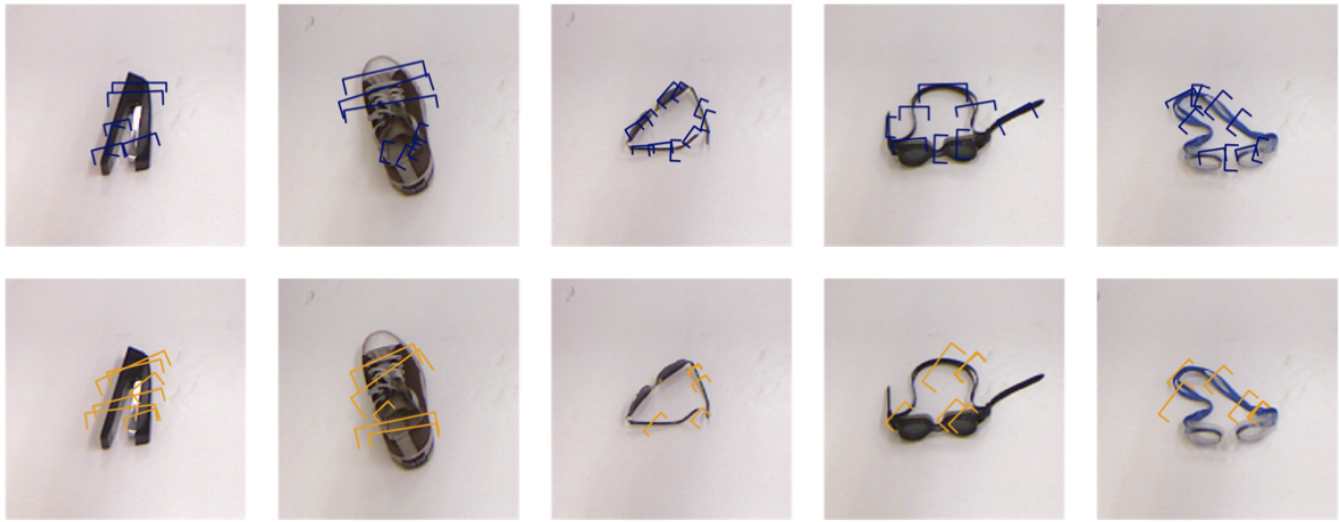
Langrange multiplier / EMM / quasi-ascent-descent (Rezende and Viola, 2018)

$$\beta^{(t)} = 1/\lambda^{(t)}, \quad \lambda^{(t+1)} = \lambda^{(t)} \exp\left\{-\eta \left(\mathbb{E}_{\hat{p}(x)}\mathbb{E}_{q_\phi(z;x)}[\log p_\theta(x|z)] - \mathcal{L}_0\right)\right\}$$

Application
Conditional Auto-encoders for learning grasping

Conditional VAEs and grasping

(Klushyn et al., 2019)



[Cornell Robot Grasping dataset]

Conditional VAE model

$$p_\theta(y|x) = \int dz\, p_\theta(y|z,\cancel{x})\, p_\theta(z|x), \qquad p(z|x,y) \approx q_\phi(z;x,y)$$

Learning the prior to improve approximation

- optimal prior in VAEs

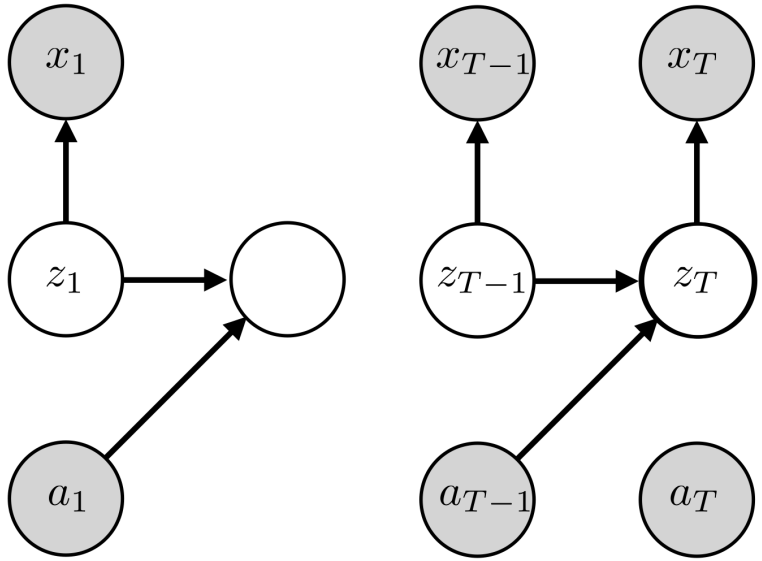
$$p^*(z|x) = \frac{1}{|D|}\sum_i q(z;x,y_i)$$

- as scalable version with learned pesudo-data (VAMP prior)

$$p_{\tilde{y}_{1:K}}(z|x) = \frac{1}{|K|}\sum_k q_\phi(z;x,\tilde{y}_k)$$

Contents

Markov model



Sequence model with latent variables

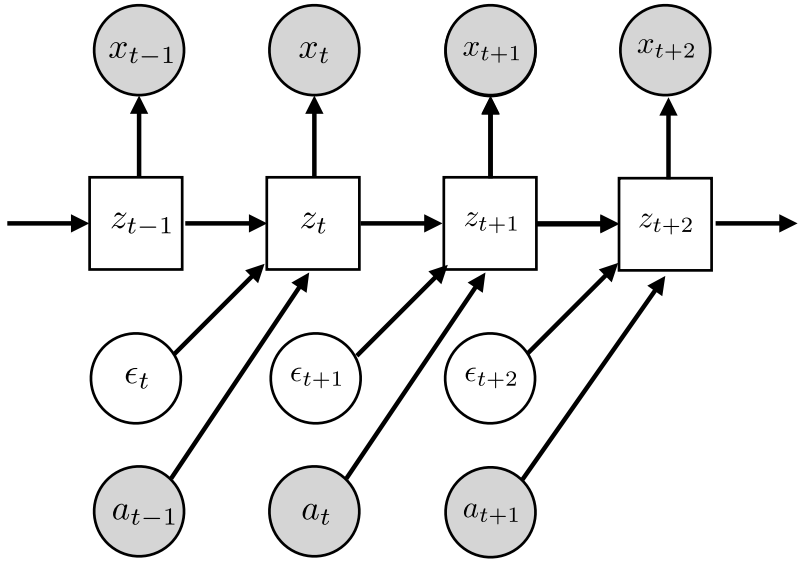
$$p(x, z|a) = p(z_1)p(x_1|z_1) \prod_t p(z_{t+1}|z_t, a_t)\, p(x_{t+1}|z_{t+1})$$

Variational objective

$$L(\theta, \phi) = \mathbb{E}_{\hat{p}(x,a)}\mathbb{E}_{q(z;x,a)}[\log p_\theta(x|z) - \mathbb{D}[q(z; x, a) \parallel p_\theta(z|a)]]$$

STORN / Increment reparameterisation

(Bayer and Osendorfer, 2014)



Sequence model with latent variables

$$p_\theta(x|a) = \int p_\theta(x|z) \prod_t p_\theta(z_{t+1}|z_t, a_t) dz_{1:T}$$

$$= \int p_\theta(x|z) \prod_t \delta_0(z_{t+1} - f(z_t, a_t, \epsilon_{t+1}))\, p(\epsilon_{1:T}) d\epsilon_{1:T}$$

Encoder

$$q_\phi(\epsilon_{1:T}|x_{1:T}) = \prod_t \mathcal{N}(\epsilon_t|\mu_t^{\mathrm{BRNN}}(x_{1:T}), \sigma_t^{\mathrm{BRNN}}(x_{1:T})^2)$$

Contents

Probabilistic Graphical Models

- representation of hierarchical Bayesian models
- inference if hierarchical Bayesian models

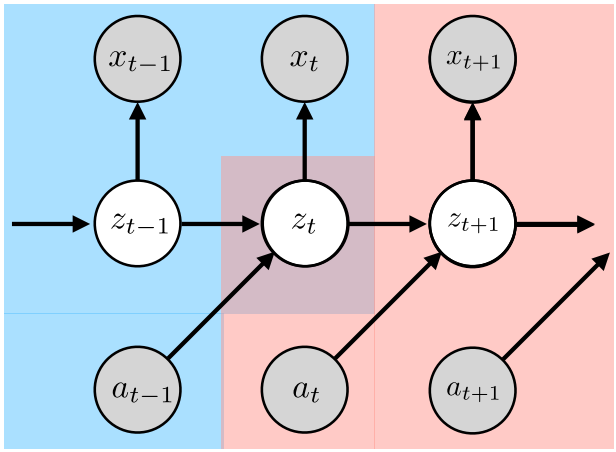
Inference as optimisation

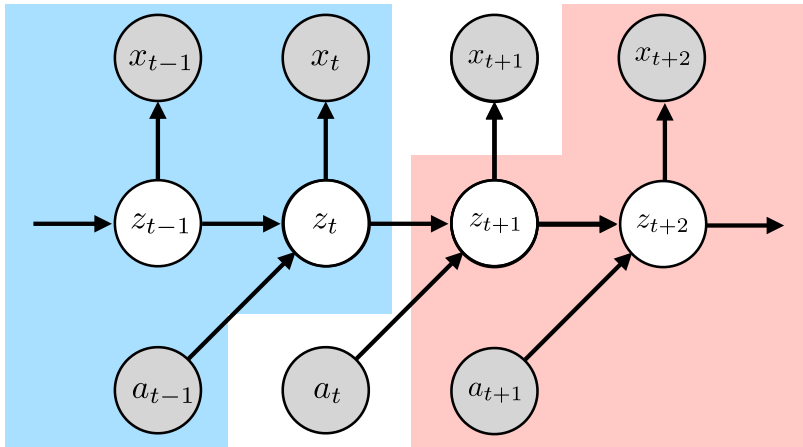
- divergence measures for approximating densities
- approximate inference in PGMs
- inference as learning with auto-encoders

State Space Models

- State space models and auto-encoders
- exact inference/computation in SSMs
- inference as learning with SSM auto-encoders

Marginals





Single-time marginals

$$p(z_t|x_{1:T}, a_{1:T}) \propto \underbrace{p(x_{t+1:T}|z_t, a_{t:T})}_{\text{likelihood}} \times \underbrace{p(z_t|x_{1:t}, a_{1:t-1})}_{\text{filter}}$$

Joint marginals

$$p(z_t, z_{t+1}|x_{1:T}, a_{1:T}) \propto \underbrace{p(x_{t+2:T}|z_{t+1}, a_{t+1:T})}_{\text{likelihood}} \underbrace{p(x_{t+1}|z_{t+1})}_{\text{likelihood/current}}$$

$$\underbrace{p(z_{t+1}|z_t, a_t)}_{\text{transition}} \underbrace{p(z_t|x_{1:t}, a_{1:t-1})}_{\text{filter}}$$

Posterior processes

Single-time marginals

$$p(z_t|x_{1:T}, a_{1:T}) \propto \underbrace{p(x_{t+1:T}|z_t, a_{t:T})}_{\text{likelihood}} \underbrace{p(z_t|x_{1:t}, a_{1:t-1})}_{\text{filter}}$$

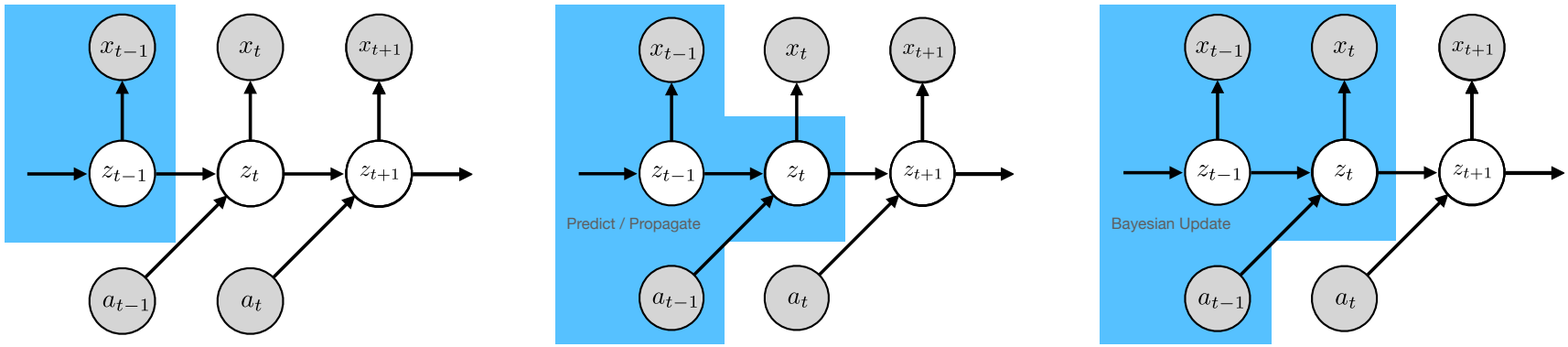
Joint marginals

$$p(z_{t+1}, z_t|x_{1:T}, a_{1:T}) \propto \underbrace{p(x_{t+2:T}|z_{t+1}, a_{t+1:T})}_{\text{likelihood}} \underbrace{p(x_{t+1}|z_{t+1})}_{\text{likelihood/current}} \underbrace{p(z_{t+1}|z_t, a_t)}_{\text{transition}} \underbrace{p(z_t|x_{1:t}, a_{1:t-1})}_{\text{filter}}$$

Posterior forward process

$$\underbrace{p(z_{t+1}|z_t, x_{t+1:T}, a_{t:T})}_{\text{posterior transition}} \propto \underbrace{\frac{1}{p(x_{t+1:T}|z_t, a_{t:T})}}_{\text{likelihood}} \underbrace{p(x_{t+2:T}|z_{t+1}, a_{t+1:T})}_{\text{likelihood}} \underbrace{p(x_{t+1}|z_{t+1})}_{\text{likelihood/current}} \underbrace{p(z_{t+1}|z_t, a_t)}_{\text{transition}}$$

Forward recursion



Predict/propagate

$$p(z_{t+1}|x_{1:t}, a_{1:t}) = \int p(z_{t+1}|z_t, a_t)\, p(z_t|x_{1:t}, a_{1:t-1}) dz_t$$

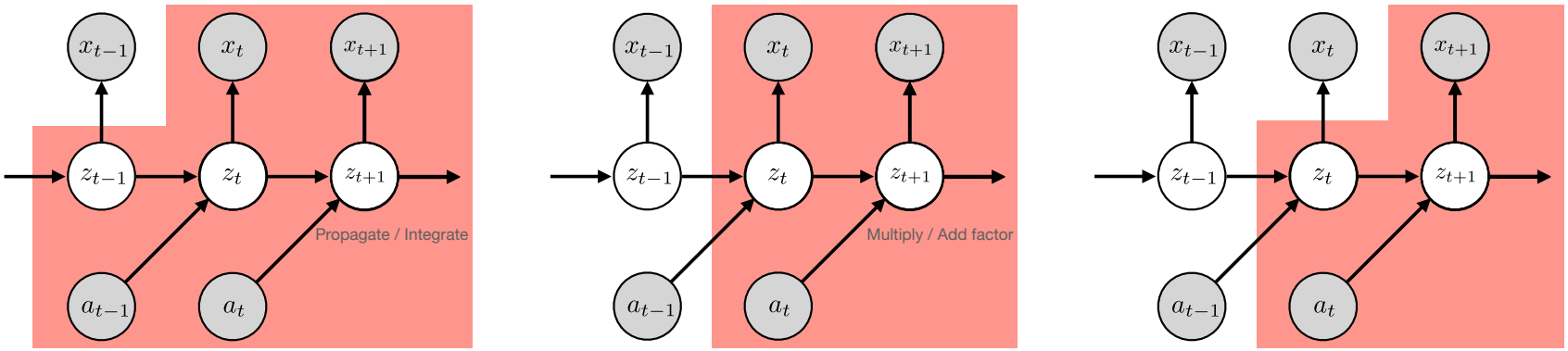
Bayesian update

$$p(z_{t+1}|x_{1:t+1}, a_{1:t}) = \frac{1}{p(x_{t+1}|x_{1:t}, a_{1:t})}\, p(x_{t+1}|z_{t+1})\, p(z_{t+1}|x_{1:t}, a_{1:t})$$

Likelihood and the predictive distribution

$$p(x_{1:t+1}|a_{1:t}) = \prod_t p(x_{t+1}|x_{1:t}, a_{1:t})$$

Backward recursion



Add factor to likelihood

$$p(x_{t:T}|z_t, a_{t:T}) = p(x_{t+1:T}|z_t, a_{t:T})\, p(x_t|z_t)$$

Propagate backward

$$p(x_{t:T}|z_{t-1}, a_{t-1:T}) = \int p(x_{1:t}|z_t, a_{t:T})\, p(z_t|z_{t-1}, a_{t-1}) dz_t$$

Contents

Probabilistic Graphical Models

- representation of hierarchical Bayesian models
- inference if hierarchical Bayesian models

Inference as optimisation

- divergence measures for approximating densities
- approximate inference in PGMs
- inference as learning with auto-encoders

State Space Models

- State space models and auto-encoders
- exact inference/computation in SSMs
- inference as learning with SSM auto-encoders

Bits and pieces / paths sums

Loss function

$$L(\theta, \phi; x, a) = -\mathbb{E}_{q_\phi(z;x,a)}[\log p_\theta(x|z)] + \mathbb{D}[q_\phi(z; x, a) \parallel p_\theta(z|a)]$$

The likelihood

$$\mathbb{E}_{q_\phi(z;x,a)}[\log p_\theta(x|z)] = \mathbb{E}_{q_\phi(z;x,a)}[\sum\nolimits_t \log p_\theta(x_t|z_t)]$$

The divergence

$$\begin{aligned}\mathbb{D}[q_\phi(z; x, a) \parallel p_\theta(z|a)] &= \mathbb{E}_{q_\phi(z;x,a)}[\sum_t \log q_\phi(z_{t+1}|z_t, a, x) - \log p_\theta(z_{t+1}|z_t, a_t)] \\ &= \mathbb{E}_{q_\phi(z;x,a)}\left[\sum_t \mathbb{D}[q_\phi(z_{t+1}|z_t, a, x) \| p_\theta(z_{t+1}|z_t, a_t)]\right]\end{aligned}$$

Deep Variational Bayes Filter

(Karl et al., 2016)

Posterior forward process

$$\underbrace{p_\theta(z_{t+1}|\bar{z}_t, x_{t+1:T}, a_{t:T})}_{\text{post.transition}} \propto \underbrace{p_\theta(x_{t+2:T}|z_{t+1}, a_{t+1:T})}_{\text{likelihood}} \underbrace{p_\theta(x_{t+1}|z_{t+1})}_{\text{likelihood/current}} \underbrace{p_\theta(z_{t+1}|\bar{z}_t, a_t)}_{\text{transition}}$$

DVBF approximation

- posterior transition

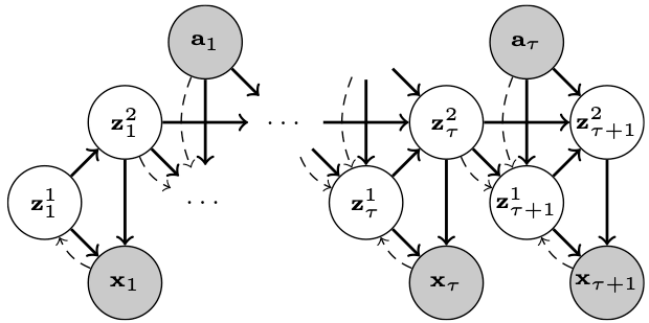
$$\underbrace{q_\phi(z_{t+1}|z_t, x_{t+1}, a_{t:T})}_{\text{post. transition appx.}} \propto \underbrace{q_\phi(z_{t+1}|x_{t+1})}_{\text{encoder}} \underbrace{p_\theta(z_{t+1}|z_t, a_t)}_{\text{transition}}$$

- initial distributions

$$p(z_1) = \mathcal{T}_\theta \mathcal{N}(z_1; 0, 1)$$
$$q(z_1|x_{1:K}) = \mathcal{T}_\theta \mathcal{N}(z_1; \mu_{\text{NN}}(x_{1:K}), \sigma_{\text{NN}}(x_{1:K})^2)$$

Stochastic Latent Actor Critic

(Lee et al., 2020)



SLAC filter approximation

- prior transition

$$p_\theta(z_{t+1}|z_t, a_t) = p_\theta(z_{t+1}^2|z_t^2, z_{t+1}^1, a_t)\, p_\theta(z_{t+1}^1|z_t^2, a_t)$$

- posterior transition

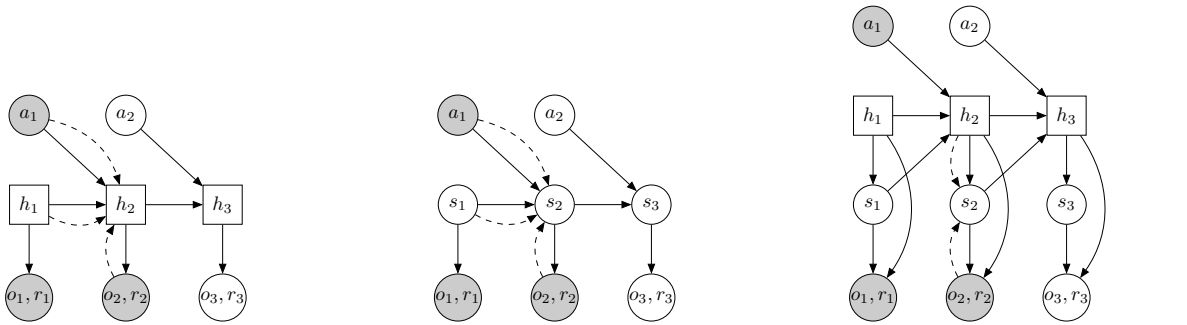
$$q_{\theta,\phi}(z_{t+1}|z_t, a_t) = p_\theta(z_{t+1}^2|z_t^2, z_{t+1}^1, a_t)\, q_\phi(z_{t+1}^1|z_t^2, a_t, x_{t+1})$$

- initial distributions

$$p_\theta(z_1) = p_\theta(z_1^2|z_0^1)\, \mathcal{N}(z_1^1; 0, I)$$
$$q_{\theta,\phi}(z_1|x_1) = p_\theta(z_1^2|z_1^1)\, q_\phi(z_1^1|x_1)$$

PlaNet

(Hafner et al., 2019)



(a) Deterministic model (RNN) (b) Stochastic model (SSM) (c) Recurrent state-space model (RSSM)

Figure: $h_t \to z_t^2, s_t \to z_t^1$

Deterministic/stochastic decomposition

- prior transition

$$p_\theta(z_{t+1}|z_t, a_t) = \delta_0(z_{t+1}^2 - f_\theta(z_t^2, z_t^1, a_t))\, p_\theta(z_{t+1}^1|z_{t+1}^2)$$

- posterior transition

$$q_{\theta,\phi}(z_{t+1}|z_t, a_t) = \delta_0(z_{t+1}^2 - f_\theta(z_t^2, z_t^1, a_t))\, q_\phi(z_{t+1}^1|z_{t+1}^2, x_{t+1})$$

- initial distributions

$$p_\theta(z_1) = p_\theta(z_0^2|z_1^1)\, \mathcal{N}(z_1^1; 0, I)$$
$$q_{\theta,\phi}(z_1|x_1) = p_\theta(z_1^2|z_0^1)\, q_\phi(z_1^1|x_1)$$

Contents

References

C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. 50, 2003.

J. Bayer and C. Osendorfer. Learning stochastic recurrent networks, 2014.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks*. 2007.

F. Frank, A. Paraschos, P. van der Smagt, and B. Cseke. Constrained probabilistic movement primitives for robot trajectory adaptation. *IEEE, Transactions in Robotics (accepted)*, 2022.

D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning Latent Dynamics for Planning from Pixels. In *International Conference on Machine Learning*, pages 2555–2565. PMLR, 2019.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations*. OpenReview.net, 2017.

M. Karl, M. Soelch, J. Bayer, and P. van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data, 2016.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*. OpenReview.net, 2014.

A. Klushyn, N. Chen, B. Cseke, J. Bayer, and P. van der Smagt. Increasing the generalisation capacity of conditional vaes. *Artificial Neural Networks and Machine Learning – ICANN 2019: Deep Learning*, page 779–791, 2019.

A. X. Lee, A. Nagabandi, P. Abbeel, and S. Levine. Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model. page 12, 2020.

T. Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, 2005.

A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

D. J. Rezende and F. Viola. Taming VAEs. *arXiv preprint arXiv:1810.00597*, 2018.

C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. *NeurIPS*, 2016.