Variational Bayes as Surrogate Regression

Matthew Ashman and Will Tebbutt

March 4, 2021

VI Intro

Interested in the **posterior distribution**

$$p(\mathbf{z}|\mathbf{y}) = \frac{p(\mathbf{z})p(\mathbf{y}|\mathbf{z})}{p(\mathbf{y})}$$

where

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{z})p(\mathbf{z})d\mathbf{z}.$$

Typically can't compute $p(\mathbf{y})$:

- Non-conjugate likelihoods
- Computationally expensive

VI Intro

Variational methodology: reformulate quantities of interest in terms of finding a solution to an optimisation problem:

$$p(\mathbf{z}|\mathbf{y}) = q^*(\mathbf{z}) = \arg\min_{q(\mathbf{z})} \mathrm{KL}\left(q(\mathbf{z}) \parallel p(\mathbf{z}|\mathbf{y})\right)$$

where

$$\mathrm{KL}\left(q(\mathbf{z}) \parallel p(\mathbf{z}|\mathbf{y})\right) = \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{y})} d\mathbf{z}.$$

Restrict q to lie within a set of parametric distributions $q_\phi(\mathbf{z})$:

$$p(\mathbf{z}|\mathbf{y}) \approx q_\phi^*(\mathbf{z}) = \arg\min_{q_\phi(\mathbf{z})} \mathrm{KL}\left(q(\mathbf{z}) \parallel p(\mathbf{z}|\mathbf{y})\right)$$

VI Intro

Rather than working with the KL divergence, we minimise the **evidence lower bound** (ELBO):

$$\begin{aligned}\mathrm{KL}\left(q_\phi(\mathbf{z}) \parallel p(\mathbf{z}|\mathbf{y})\right) &= \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z})}{p(\mathbf{z}|\mathbf{y})}\right] \\ &= \underbrace{\mathbb{E}_q\left[\log q_\phi(\mathbf{z})\right] - \mathbb{E}_q\left[\log p(\mathbf{z}, \mathbf{y})\right]}_{-\mathcal{L}_{\mathrm{ELBO}}} + \log p(\mathbf{y})\end{aligned}$$

Minimising KL divergence = maximising $\mathcal{L}_{\mathrm{ELBO}}$

(Also estimate $\log p(\mathbf{y})$)

Other Approaches to VI: Laplace Approximation

Let $\hat{\mathbf{z}}$ be the MAP of $p(\mathbf{z}|\mathbf{y})$. Taylor expansion around $\hat{\mathbf{z}}$:

$$\log p(\mathbf{z}|\mathbf{y}) \approx \log p(\hat{\mathbf{z}}|\mathbf{y}) + \frac{1}{2}(\mathbf{z} - \hat{\mathbf{z}})^T H(\hat{\mathbf{z}})(\mathbf{z} - \hat{\mathbf{z}})$$

where

$$H(\hat{\mathbf{z}}) = \nabla^2 \log p(\mathbf{z}|\mathbf{y})|_{\mathbf{z}=\hat{\mathbf{z}}} = \nabla^2 \log p(\mathbf{y}, \mathbf{z})|_{\mathbf{z}=\hat{\mathbf{z}}}$$

This corresponds to the **Laplace approximation**:

$$q(\mathbf{z}) = \mathcal{N}(\mathbf{z};\ \hat{\mathbf{z}},\ -H(\hat{\mathbf{z}}))$$

Other Approaches to VI: EP and PEP

Replace the joint distribution with the approximation $q^*(\mathbf{z})$:

$$p(\mathbf{y}, \mathbf{z}) = p(\mathbf{z}) \prod_{n=1}^{N} p(y_n|\mathbf{z})$$
$$\approx p(\mathbf{z}) \prod_{n=1}^{N} t_n(\mathbf{z}) = q^*(\mathbf{z}).$$

Deletion: $q^{\backslash n}(\mathbf{z}) \propto \frac{q^*(\mathbf{z})}{t_n(\mathbf{z})^\alpha}$

Titled distribution: $\tilde{p}(\mathbf{z}) = q^{\backslash n}(\mathbf{z}) p(y_n|\mathbf{z})^\alpha$

Projection: $q^*(\mathbf{z}) \leftarrow \arg\min_{q^* \in \mathcal{Q}} \mathrm{KL}\left(\tilde{p}_n(\mathbf{z}) \parallel q^*(\mathbf{z})\right)$

Update: $t_{n,\mathsf{new}}(\mathbf{z})^\alpha = \frac{q^*(\mathbf{z})}{q^{\backslash n}(\mathbf{z})} \;\Rightarrow\; t_n(\mathbf{z}) = t_{n,\mathsf{new}}(\mathbf{z})^\alpha t_{n,\mathsf{old}}(\mathbf{z})^{1-\alpha}$

Central question: form of approximate posterior

How should we construct $q(\mathbf{z})$?

Desiderata:

1. Close to true posterior
2. Computationally simple

List some options

Full? e.g. multi-variate Gaussian distribution:

$$q(\mathbf{z}) = \mathcal{N}\left(\mathbf{z};\ \boldsymbol{\mu},\ \boldsymbol{\Sigma}\right)$$

Mean-field? e.g. fully-factorised Gaussian distribution:

$$q(\mathbf{z}) = \prod_i \mathcal{N}\left(z_i;\ \mu_i,\ \sigma_i^2\right)$$

Flexible? e.g. apply normalising flow to fully-factorised Gaussian distribution:

$$\mathbf{z}_K = f_K \circ \cdots \circ f_1(\mathbf{z}_0)$$

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \ln \left| \det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \right|$$

Another option: Posterior of Tractable Model

Replace likelihoods? e.g. replace non-conjugate likelihoods with Gaussian approximations:

$$p(\mathbf{z}|\mathbf{y}) = \frac{1}{p(\mathbf{y})} p(\mathbf{z}) \prod_n p(y_n|\mathbf{z})$$
$$\approx \frac{1}{\mathcal{Z}_q} p(\mathbf{z}) \prod_n q_n(\mathbf{z}) = q(\mathbf{z})$$

Equivalent to posterior of tractable model:

$$q(\mathbf{z}) := \hat{p}(\mathbf{z} \mid \mathbf{y}) \propto p(\mathbf{z}) \hat{p}(\mathbf{y} \mid \mathbf{z})$$

Sidenote: relationship with EP

Recall that EP replaces the joint distribution:

$$
\begin{aligned}
p(\mathbf{y}, \mathbf{z}) &= p(\mathbf{z}) \prod_{n=1}^{N} p(y_n|\mathbf{z}) \\
&\approx p(\mathbf{z}) \prod_{n=1}^{N} t_n(\mathbf{z}) = q(\mathbf{z}) \times \mathcal{Z}_q.
\end{aligned}
$$

Important:

$$q^*_{\mathrm{EP}}(\mathbf{z}) \neq q^*_{\mathrm{VI}}(\mathbf{z})$$

- Same family of distributions
- Different "free energies"
- Different optimisation procedure

Important questions

How flexible / large is this family?

Efficient inference?

How many parameters?

Any other drawbacks / benefits?

Optimality: Exponential-Family Prior

$$p(\mathbf{z}) = h(\mathbf{z}) \exp[t(\mathbf{z})^\top \eta_{\mathbf{z}} - A(\eta_{\mathbf{z}})]$$

Refresher: Inference in Conjugate Exponential Families

$$
\begin{aligned}
p(\mathbf{z}) &= h(\mathbf{z}) \exp[t(\mathbf{z})^\top \eta_{\mathbf{z}} - A(\eta_{\mathbf{z}})] \\
p(\mathbf{y} \mid \mathbf{z}) &= \exp[t(\mathbf{z})^\top \eta_{\mathbf{y}} + C(\mathbf{y})] \\
p(\mathbf{z} \mid \mathbf{y}) &= h(\mathbf{z}) \exp[t(\mathbf{z})^\top (\eta_{\mathbf{z}} + \eta_{\mathbf{y}}) - A(\eta_{\mathbf{z}} + \eta_{\mathbf{y}})]
\end{aligned}
$$

Example: Multivariate Gaussian

$$p(\mathbf{z}) = \mathcal{N}\left(\mathbf{z};\ \mathbf{m},\ \Lambda_{\mathbf{z}}^{-1}\right)$$
$$p(\mathbf{y} \mid \mathbf{z}) = \mathcal{N}\left(\mathbf{y};\ \mathbf{z},\ \Lambda_{\mathbf{y}}^{-1}\right)$$

$$t(\mathbf{z}) = \operatorname{vec}\left(\mathbf{z}, \mathbf{z}\mathbf{z}^\top\right)$$
$$\eta_{\mathbf{z}} = \operatorname{vec}\left(\Lambda_{\mathbf{z}}\mathbf{m}, -\frac{1}{2}\Lambda_{\mathbf{z}}\right)$$
$$\eta_{\mathbf{y}} = \operatorname{vec}\left(\Lambda_{\mathbf{y}}\mathbf{y}, -\frac{1}{2}\Lambda_{\mathbf{y}}\right)$$
$$\eta_{\mathbf{z}|\mathbf{y}} = \eta_{\mathbf{z}} + \eta_{\mathbf{y}}$$

A and h are tractable

Non-Conjugate Likelihood

What if $p(\mathbf{y} \mid \mathbf{z})$ and $p(\mathbf{z})$ aren't conjugate, but p *is* Exponential Family?

Exponential-Family Approximation

Assume q in same family as prior

$$q(\mathbf{z};\eta_q) = h(\mathbf{z})\exp[t(\mathbf{z})^\top \eta_q - A(\eta_q)]$$

η_q^* satisfies

$$\eta_q^* = \eta_{\mathbf{z}} + \underbrace{\left.\frac{\mathrm{d}r}{\mathrm{d}\mu}\right|_{\mu(\eta_q^*)}}_{=:\eta_{\hat{\mathbf{y}}}}, \quad \mu(\eta_q^*) := \mathbb{E}_q[t(\mathbf{z})]\,, \quad r := \mathbb{E}_q[\log p(\mathbf{y} \mid \mathbf{z})]\,.$$

i.e. do exact inference under surrogate likelihood

$$\hat{p}(\hat{\mathbf{y}} \mid \mathbf{z}) = \exp[t(\mathbf{z})^\top \eta_{\hat{\mathbf{y}}} + C(\hat{\mathbf{y}})].$$

Why Bother?

Why are we telling you this?

If $\eta_{\mathbf{z}}$ and $\eta_{\hat{\mathbf{y}}}$ etc are arbitrary vectors of numbers, not especially interesting.

What if $\eta_{\mathbf{z}}$ and $\eta_{\hat{\mathbf{y}}}$ have exploitable structure?

Example: Independent Observations of a Gaussian

$$p(\mathbf{z}) = \mathcal{N}\left(\mathbf{z};\ \mathbf{m},\ \Lambda_{\mathbf{z}}^{-1}\right), \quad p(\mathbf{y} \mid \mathbf{z}) = \prod_{n=1}^{N} p(\mathbf{y}_n \mid \mathbf{z}_n)$$

then

$$\hat{p}(\hat{\mathbf{y}} \mid \mathbf{z}) = \prod_{n=1}^{N} \mathcal{N}\left(\hat{\mathbf{y}}_n;\ \mathbf{z}_n,\ \lambda_n^{-1}\right)$$

is optimal.

$$\Lambda_q^* = \Lambda_{\mathbf{z}} + \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix}.$$

$2N$ variational parameters vs $(N+1)N$. See Opper[1].

[1] Opper and Archambeau, "The Variational Gaussian Approximation Revisited".

Towards Gaussian Processes

$$p(\mathbf{f}, \mathbf{f}_*) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f} \\ \mathbf{f}_*\end{bmatrix};\ 0,\ \begin{bmatrix}\mathbf{K}_{\mathbf{ff}} & \mathbf{K}_{\mathbf{ff}_*} \\ \mathbf{K}_{\mathbf{f}_*\mathbf{f}} & \mathbf{K}_{\mathbf{f}_*\mathbf{f}_*}\end{bmatrix}\right)$$

$$p(\mathbf{y} \mid \mathbf{f}, \mathbf{f}_*) = \prod_{n=1}^{N} p(\mathbf{y}_n \mid \mathbf{f}_n)$$

$$\hat{p}(\mathbf{y} \mid \mathbf{f}, \mathbf{f}_*) = \prod_{n=1}^{N} \mathcal{N}\left(\hat{\mathbf{y}}_n;\ \mathbf{f}_n,\ \lambda_n^{-1}\right)$$

then

$$\begin{aligned} q(\mathbf{f}, \mathbf{f}_*) &= p(\mathbf{f}_* \mid \mathbf{f})\, \hat{p}(\mathbf{f} \mid \hat{\mathbf{y}}) \\ &\propto p(\mathbf{f}_* \mid \mathbf{f})\, p(\mathbf{f}) \prod_{n=1}^{N} \mathcal{N}\left(\hat{\mathbf{y}}_n;\ \mathbf{f}_n,\ \lambda_n^{-1}\right). \end{aligned}$$

$2N$ variational parameters vs $(N + N_* + 1)(N + N_*)$.

Optimising the Variational Parameters

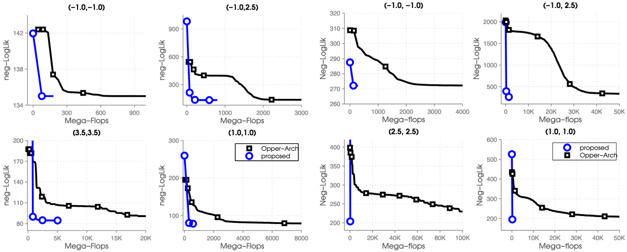
Easy to optimise the variational parameters?

Apparently not[2]

[2] M. E. Khan, Mohamed, and Murphy, "Fast Bayesian Inference for Non-Conjugate Gaussian Process Regression."

Proposed Solutions

Coordinate ascent procedure[3]



(a) Ionosphere data (b) Forensic glass data

Figure 1: Convergence results for (a) the binary classification on the ionosphere data set and (b) the multi-class classification on the glass dataset. We plot the negative of the lower bound vs the number of flops. Each plot shows the progress of algorithms for a hyperparameter setting $\{\log(s), \log(\sigma)\}$ shown at the top of the plot. The proposed algorithm always converges faster than the other method, in fact, in less than 5 iterations.

---

[3] M. E. Khan, Mohamed, and Murphy, "Fast Bayesian Inference for Non-Conjugate Gaussian Process Regression."

Proposed Solutions

Natural Gradient Ascent in $\eta_{\hat{\mathbf{y}}}$[4]

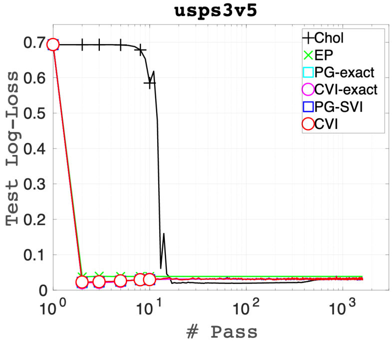


Figure 2: Comparison on Gaussian Process Classification.

[4] M. Khan and Lin, "Conjugate-Computation Variational Inference: Converting Variational Inference in Non-Conjugate Models to Inferences in Conjugate Models".

Natural Gradients in General

Assume posterior, $q(\mathbf{z})$, belongs to the same exponential family as the prior $p(\mathbf{z})$:

$$q(\mathbf{z}; \eta_q) \propto h(\mathbf{z}) \exp\left[t(\mathbf{z})^\top \eta_q - A(\eta_q)\right]$$

Gradient ascent:

$$\eta_q^{(t+1)} = \eta_q^{(t)} + \rho \nabla_{\eta_q} \mathcal{L}(\eta_q)\big|_{\eta_q^{(t)}}$$

Natural gradient ascent:

$$\eta_q^{(t+1)} = \eta_q^{(t)} + \alpha \underbrace{\mathbf{F}(\eta_q^{(t)})^{-1} \nabla_{\eta_q} \mathcal{L}(\eta_q)\big|_{\eta_q^{(t)}}}_{\tilde{\nabla}_{\eta_q} \mathcal{L}(\eta_q)|_{\eta_q^{(t)}}}$$

$$\mathbf{F}(\eta_q) = \mathbb{E}_{q(\mathbf{z})}\left[\nabla_{\eta_q} \log q(\mathbf{z}) \nabla \eta_q \log q(\mathbf{z})^\top\right].$$

Natural Gradients in Our Case

Fisher information matrix is simple:

$$\mathbf{F} = \frac{\mathrm{d}\mu}{\mathrm{d}\eta}, \quad \frac{\mathrm{d}A}{\mathrm{d}\eta} = \mu, \quad \mu(\eta) := \mathbb{E}[t(\mathbf{z})]\,.$$

Working everything through:

$$\eta_q^{(t+1)} = \eta_q^{(t)} + \alpha\left[\eta_{\mathbf{z}} - \eta_q^{(t)} + \eta_{\hat{\mathbf{y}}}^{(t)}\right], \quad \eta_{\hat{\mathbf{y}}}^{(t)} := \left.\frac{\mathrm{d}r}{\mathrm{d}\mu}\right|_{\mu(\eta_q^{(t)})}$$

Natural Gradients in Our Case

Reparametrisation:

$$\eta_q^{(t)} = \eta_{\mathbf{z}} + \tilde{\eta}_q^{(t)}$$

Then

$$\begin{aligned}\tilde{\eta}_q^{(t+1)} &= \tilde{\eta}_q^{(t)} + \alpha[\eta_{\hat{\mathbf{y}}}^{(t)} - \tilde{\eta}_q^{(t)}] \\ &= (1-\alpha)\tilde{\eta}_q^{(t)} + \alpha\eta_{\hat{\mathbf{y}}}^{(t)}.\end{aligned}$$

The Frontier

What's interesting work that people are doing at the moment?

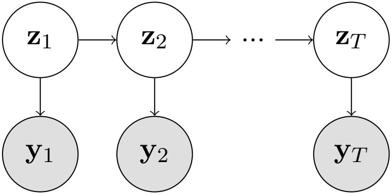
Very helpful in state-space + pseudo-point approximations

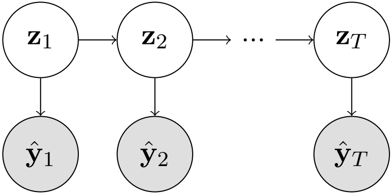
Exploiting (Approximate) Markov Structure in the Prior

Combine with state-space approx.[5] Linear-time approx. inference.

$$p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) := \prod_{t=1}^{T} p(\mathbf{z}_t \mid \mathbf{z}_{t-1}) p(\mathbf{y}_t \mid \mathbf{z}_t)$$

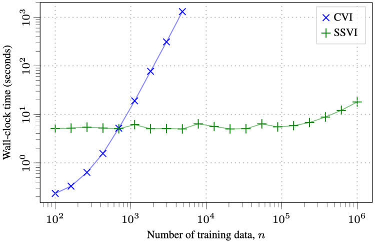
$$q(\mathbf{z}_{1:T}) \propto \prod_{t=1}^{T} p(\mathbf{z}_t \mid \mathbf{z}_{t-1}) \mathcal{N}\left(\hat{\mathbf{y}}_t;\ \mathbf{z}_t,\ \hat{\sigma}^2\right)$$



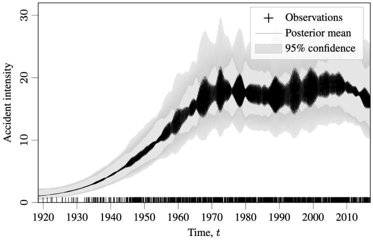


[5] Chang et al., "Fast Variational Learning in State-Space Gaussian Process Models"; Grigorievskiy, Lawrence, and Särkkä, "Parallelizable sparse inverse formulation Gaussian processes (SpInGP)".

Exploiting (Approximate) Markov Structure in the Prior



(a) Number of data vs. wall-clock time



(b) Airline accidents ($n \approx 40\,\mathrm{k}$)

Other Exploitable Structure

$p(\mathbf{y} \mid \mathbf{z}) = \prod_{n=1}^{N} p(\mathbf{y}_n \mid \mathbf{z})$ – see[6]

Pseudo-points + state-space[7][8]

Potential for new variational methods tailored to sparse graph structure

VB analogue of INLA[9]?

---

[6] Bui et al., "Partitioned Variational Inference: A Unified Framework Encompassing Federated and Continual Learning".

[7] Adam et al., "Doubly sparse variational Gaussian processes".

[8] Tebbutt, Solin, and Turner, "Combining Pseudo-Point and State Space Approximations for Sum-Separable Gaussian Processes".

[9] Rue, Martino, and Chopin, "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations".

What can you do with amortisation?

Ashman et al[10] amortise the parameters of the surrogate regression problem:

$$
\begin{aligned}
p(f|\mathbf{y}, \mathbf{X}) &= \frac{1}{\mathcal{Z}_p} p(f) \prod_{n=1}^{N} p(\mathbf{y}_n \mid f, \mathbf{x}_n) \\
&\approx \frac{1}{\mathcal{Z}_q} p(f) \prod_{n=1}^{N} \hat{p}(\hat{y}_n \mid f, \mathbf{x}_n) = q(f)
\end{aligned}
$$

where

$$
\begin{gathered}
\hat{y}_n = k_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{u} + \hat{\sigma}_n \epsilon_n \\
\hat{y}_n, \ \hat{\sigma}_n \underset{g_\phi(\cdot)}{\longleftarrow} \mathbf{y}_n
\end{gathered}
$$

[10] Ashman et al., "Sparse Gaussian Process Variational Autoencoders".

What can you do with amortisation?

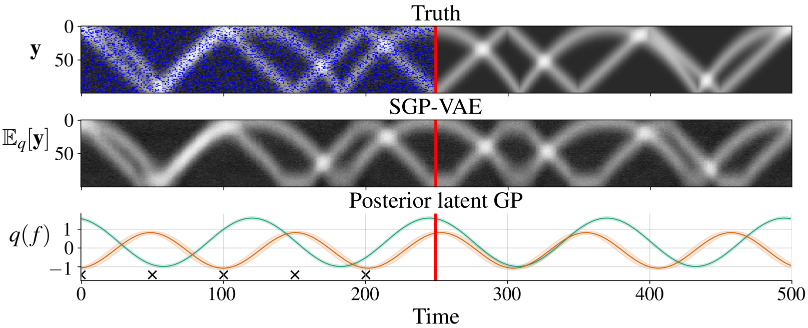


Figure: True regression problem (top) vs. surrogate regression problem (bottom).

Structured variational autoencoder

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(\mathbf{z})q(\theta)}\left[\log \frac{p(\theta)p(\mathbf{z}|\theta)p(\mathbf{y}|\mathbf{z})}{q(\mathbf{z})q(\theta)}\right]$$

Conditionally-conjugate $\implies$ $q^*(\mathbf{z}) = \exp[t(\mathbf{z})^T \eta_q^* - A(\eta_q^*)]$

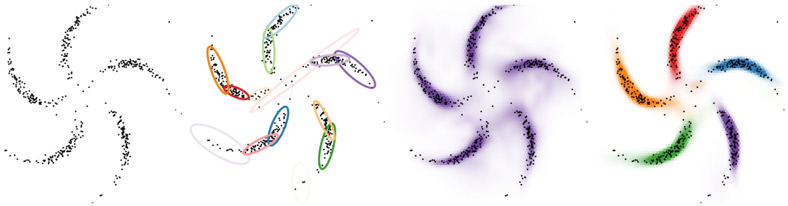
$p(\mathbf{y}|\mathbf{z})$ non-conjugate? Replace with conjugate approximation[11]:

$$\hat{\mathcal{L}} = \mathbb{E}_{q(\mathbf{z})q(\theta)}\left[\log \frac{p(\theta)p(\mathbf{z}|\theta)\exp\{\psi(\mathbf{z};\mathbf{y},\phi)\}}{q(\mathbf{z})q(\theta)}\right]$$

$$\psi(\mathbf{z};\mathbf{y},\phi) = t(\mathbf{z})^T r(\mathbf{y};\phi)$$

[11] Johnson, "Structured VAEs: Composing probabilistic graphical models and variational autoencoders".

Structured variational autoencoder



(a) Data (b) GMM (c) Density net (VAE) (d) GMM SVAE

Figure 1: Comparison of generative models fit to spiral cluster data. See Section 2.1.

Summary

Basic idea

Principle advantages in use cases

Interesting future directions?