Probabilistic & Unsupervised Learning
Approximate Inference

Parametric Variational Methods
and Recognition Models

Maneesh Sahani
maneesh@gatsby.ucl.ac.uk

Gatsby Computational Neuroscience Unit, and
MSc ML/CSML, Dept Computer Science
University College London

Term 1, Autumn 2020

Variational methods

- Our treatment of variational methods has (except EP) emphasised 'natural' choices of variational family – often factorised using the same functional (ExpFam) form as joint.
  - mostly restricted to joint exponential families – facilitates hierarchical and distributed models, but not non-linear/non-conjugate.
- Consider parametric variational approximations using a constrained family $q(\mathcal{Z};\rho)$.

  The constrained (approximate) variational E-step becomes:

$$q(\mathcal{Z}) := \underset{q\in\{q(\mathcal{Z};\rho)\}}{\text{argmax}}\ \mathcal{F}\big(q(\mathcal{Z}),\theta^{(k-1)}\big) \quad \Rightarrow \quad \rho^{(k)} := \underset{\rho}{\text{argmax}}\ \mathcal{F}\big(q(\mathcal{Z};\rho),\theta^{(k-1)}\big)$$

  and so we can replace constrained optimisation of $\mathcal{F}(q,\theta)$ with unconstrained optimisation of a constrained $\mathcal{F}(\rho,\theta)$:

$$\mathcal{F}(\rho,\theta) = \left\langle \log P(\mathcal{X},\mathcal{Z}|\theta^{(k-1)}) \right\rangle_{q(\mathcal{Z};\rho)} + \mathbf{H}[\rho]$$

  It might still be valuable to use coordinate ascent in ρ and θ, although this is no longer necessary.

Optimising the variational parameters

$$\mathcal{F}(\rho,\theta) = \left\langle \log P(\mathcal{X},\mathcal{Z}|\theta^{(k-1)}) \right\rangle_{q(\mathcal{Z};\rho)} + \mathbf{H}[\rho]$$

- In some special cases, the expectations of the log-joint under $q(\mathcal{Z};\rho)$ can be expressed in closed form, but these are rare.
- Otherwise we might seek to follow $\nabla_\rho\mathcal{F}$.
- Naively, this requires evaluting a high-dimensional expectation wrt $q(\mathcal{Z},\rho)$ as a function of ρ – not simple.
- At least three solutions:
  - "Score-based" gradient estimate, and Monte-Carlo (Ranganath et al. 2014).
  - Recognition network trained in separate phase – not strictly variational (Dayan et al. 1995).
  - Recognition network trained simultaneously with generative model using "frozen" samples (Kingma and Welling 2014; Rezende et al. 2014).

Score-based gradient estimate

We have:

$$\begin{aligned}\nabla_\rho\mathcal{F}(\rho,\theta) &= \nabla_\rho \int d\mathcal{Z}\, q(\mathcal{Z};\rho)(\log P(\mathcal{X},\mathcal{Z}|\theta) - \log q(\mathcal{Z};\rho)) \\ &= \int d\mathcal{Z}\, [\nabla_\rho q(\mathcal{Z};\rho)](\log P(\mathcal{X},\mathcal{Z}|\theta) - \log q(\mathcal{Z};\rho)) \\ &\qquad + q(\mathcal{Z};\rho)\nabla_\rho[\log P(\mathcal{X},\mathcal{Z}|\theta) - \log q(\mathcal{Z};\rho)]\end{aligned}$$

Now,

$$\nabla_\rho \log P(\mathcal{X},\mathcal{Z}|\theta) = 0 \qquad \text{(no direct dependence)}$$

$$\int d\mathcal{Z}\, q(\mathcal{Z};\rho)\nabla_\rho \log q(\mathcal{Z};\rho) = \nabla_\rho \int d\mathcal{Z}\, q(\mathcal{Z};\rho) = 0 \qquad \text{(always normalised)}$$

$$\nabla_\rho q(\mathcal{Z};\rho) = q(\mathcal{Z};\rho)\nabla_\rho \log q(\mathcal{Z};\rho)$$

So,

$$\nabla_\rho\mathcal{F}(\rho,\theta) = \left\langle [\nabla_\rho \log q(\mathcal{Z};\rho)](\log P(\mathcal{X},\mathcal{Z}|\theta) - \log q(\mathcal{Z};\rho)) \right\rangle_{q(\mathcal{Z};\rho)}$$

Reduced gradient of expectation to expectation of gradient – easier to compute. Also called the REINFORCE trick.

Factorisation

$$\nabla_\rho \mathcal{F}(\rho,\theta) = \Big\langle [\nabla_\rho \log q(\mathcal{Z};\rho)](\log P(\mathcal{X},\mathcal{Z}|\theta) - \log q(\mathcal{Z};\rho)) \Big\rangle_{q(\mathcal{Z};\rho)}$$

- Still requires a high-dimensional expectation, but can now be evaluated by Monte-Carlo.
- Dimensionality reduced by factorisation (particularly where $P(\mathcal{X},\mathcal{Z})$ is factorised).

  Let $q(\mathcal{Z}) = \prod_i q(\mathcal{Z}_i|\rho_i)$ factor over disjoint cliques; let $\bar{\mathcal{Z}}_i$ be the minimal Markov blanket of $\mathcal{Z}_i$ in the joint; $P_{\bar{\mathcal{Z}}_i}$ be the product of joint factors that include any element of $\mathcal{Z}_i$ (so the union of their arguments is $\bar{\mathcal{Z}}_i$); and $P_{\neg\bar{\mathcal{Z}}_i}$ the remaining factors. Then,

$$\begin{aligned}
\nabla_{\rho_i}\mathcal{F}(\{\rho_j\},\theta) &= \Big\langle [\nabla_{\rho_i} \textstyle\sum_j \log q(\mathcal{Z}_j;\rho_j)](\log P(\mathcal{X},\mathcal{Z}|\theta) - \textstyle\sum_j \log q(\mathcal{Z}_j;\rho_j)) \Big\rangle_{q(\mathcal{Z})} \\
&= \Big\langle [\nabla_{\rho_i} \log q(\mathcal{Z}_i;\rho_i)](\log P_{\bar{\mathcal{Z}}_i}(\mathcal{X},\bar{\mathcal{Z}}_i) - \log q(\mathcal{Z}_i;\rho_i)) \Big\rangle_{q(\bar{\mathcal{Z}}_i)} \\
&\quad + \Big\langle [\nabla_{\rho_i} \log q(\mathcal{Z}_i;\rho_i)]\ \underbrace{(\log P_{\neg\bar{\mathcal{Z}}_i}(\mathcal{X},\mathcal{Z}_{\neg i}) - \sum_{j\neq i} \log q(\mathcal{Z}_j;\rho_j)}_{\text{constant wrt } \mathcal{Z}_i} \Big\rangle_{q(\mathcal{Z})}
\end{aligned}$$

So the second term is proportional to $\langle \nabla_{\rho_i} \log q(\mathcal{Z}_i;\rho_i)\rangle_{q(\mathcal{Z}_i)}$, this $= 0$ as before.

So expectations are only needed wrt $q(\bar{\mathcal{Z}}_i)$ → variational message passing!

Sampling

So the "black-box" variational approach is as follows:

- Choose a parametric (factored) variational family $q(\mathcal{Z}) = \prod_i q(\mathcal{Z}_i;\rho_i)$.
- Initialise factors.
- Repeat to convergence:
  - **Stochastic VE-step**. For each i:
    - Sample from $q(\bar{\mathcal{Z}}_i)$ and estimate expected gradient $\nabla_{\rho_i}\mathcal{F}$.
    - Update ρ_i along gradient.
  - **Stochastic M-step**. For each i:
    - Sample from each $q(\bar{\mathcal{Z}}_i)$.
    - Update corresponding parameters.
- Stochastic gradient steps may employ a Robbins-Munro step-size sequence to promote convergence.
- Variance of the gradient estimators can also be controlled by clever Monte-Carlo techniques (orginal authors used a "control variate" method that we have not studied).

Recognition Models

We have not generally distinguished between multivariate models and iid data instances, grouping all variables together in $\mathcal{Z}$.

However, even for large models (such as HMMs), we often work with multiple data draws (e.g. multiple strings) and each instance requires a separate variational optimisation.

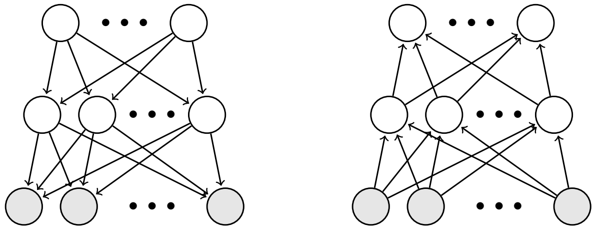
Suppose that we have fixed length vectors $\{(\mathbf{x}_i,\mathbf{z}_i)\}$ ($\mathbf{z}$ is still latent).

- Optimal variational distribution $q^*(\mathbf{z}_i)$ depends on $\mathbf{x}_i$.
- Learn this mapping (in parametric form): $q\big(\mathbf{z}_i;\rho = f(\mathbf{x}_i;\phi)\big)$.
- Now ρ is the output of a general function approximator f (a GP, neural network or similar) parametrised by ϕ, trained to map $\mathbf{x}_i$ to the variational parameters of $q(\mathbf{z}_i)$.
- The mapping function f is called a recognition model.
- This is approach is now often called amortised inference.

How to learn f?

The Helmholtz Machine

Dayan et al. (1995) originally studied binary sigmoid belief net, with parallel recognition model:



Two phase learning:

- Wake phase: given current f, estimate mean-field representation from data (mean sufficient stats for Bernoulli are just probabilities):

$$q(\mathbf{z}_i) = \text{Bernoulli}[\hat{\mathbf{z}}_i] \qquad \hat{\mathbf{z}}_i = f(\mathbf{x}_i;\phi)$$

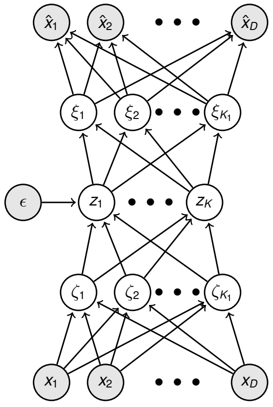
  Update generative parameters θ according to $\nabla_\theta\mathcal{F}(\{\hat{\mathbf{z}}_i\},\theta)$.
- Sleep phase: sample $\{\mathbf{z}_s,\mathbf{x}_s\}_{s=1}^S$ from current generative model. Update recognition parameters ϕ to direct $f(\mathbf{x}_s)$ towards $\mathbf{z}_s$ (simple gradient learning).

$$\Delta\phi \propto \sum_s (\mathbf{z}_s - f(\mathbf{x}_s;\phi))\nabla_\phi f(\mathbf{x}_s;\phi)$$

The Helmholtz Machine

- Can sample $\mathbf{z}$ from recognition model rather than just evaluate means.
  - Expectations in free-energy can be computed directly rather than by mean substitution.
  - In hierarchical models, output of higher recognition layers then depends on samples at previous stages, which introduces correlations between samples at different layers.
- Recognition model structure need not exactly echo generative model.
- More general approach is to train f to yield mean parameters of ExpFam $q(\mathbf{z})$ (later).
- Sleep phase learning minimises $\mathbf{KL}[p_\theta(\mathbf{z}|\mathbf{x})\|q(\mathbf{z}; f(\mathbf{x}, \phi))]$. Opposite to variational objective, but may not matter if divergence is small enough.

Variational Autoencoders



- Fuses the wake and sleep phases.
- Generate recognition samples using deterministic transformations of external random variates (reparametrisation trick).
  - E.g. if $\mathbf{f}$ gives marginal μ_i and σ_i for latents z_i and $\epsilon_i^s \sim \mathcal{N}(0,1)$, then $z_i^s = \mu_i + \sigma_i \epsilon_i^s$.
- Now generative and recognition parameters can be trained together by gradient descent (backprop), holding $\boldsymbol{\epsilon}^s$ fixed.

$$\mathcal{F}_i(\theta, \phi) = \sum_s \log P(\mathbf{x}_i, \mathbf{z}_i^s; \theta) - \log q(\mathbf{z}_i^s; \mathbf{f}(\mathbf{x}_i, \phi))$$

$$\frac{\partial}{\partial \theta}\mathcal{F}_i = \sum_s \nabla_\theta \log P(\mathbf{x}_i, \mathbf{z}_i^s; \theta)$$

$$\frac{\partial}{\partial \phi}\mathcal{F}_i = \sum_s \frac{\partial}{\partial \mathbf{z}_i^s}(\log P(\mathbf{x}_i, \mathbf{z}_i^s; \theta) - \log q(\mathbf{z}_i^s; \mathbf{f}(\mathbf{x}_i)))\frac{d\mathbf{z}_i^s}{d\phi} + \frac{\partial}{\partial \mathbf{f}(\mathbf{x}_i)} \log q(\mathbf{z}_i^s; \mathbf{f}(\mathbf{x}_i))\frac{d\mathbf{f}(\mathbf{x}_i)}{d\phi}$$

Variational Autoencoders

- Frozen samples ϵ^s can be redrawn to avoid overfitting.
- May be possible to evaluate entropy and log $P(\mathbf{z})$ without sampling, reducing variance.
- Differentiable reparametrisations are available for a number of different distributions.
- Conditional $P(\mathbf{x}|\mathbf{z}, \theta)$ is often implemented as a neural network with additive noise at output, or at transitions. If at transitions recognition network must estimate each noise input.
- In practice, hierarchical models appear difficult to learn.

More recent work

- Changing the variational cost function (tightening the bound):
  - Importance-Weighted autoencoder (IWAE)
  - Filtering variational objective (FIVO)
  - Thermodynamic variational objective (TVO)
- Flexible variational distributions (and avoiding inference)
  - Normalising flows
  - DDC-Helmholtz machine
  - Amortised learning
- Structured generative models
  - “standard” VAE generative model both too powerful and too simple for learning
  - local conjugate inference – structured VAEs

Far from exhaustive . . . these are all areas of active research. We’ll survey a few ideas.

Importance-weighted free energy

Another interpretation of the free energy:

$$\mathcal{F}(q,\theta) = \left\langle \log \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})} \right\rangle_q = \mathbb{E}_{\mathbf{z}\sim \overset{\overset{\text{proposal}}{\downarrow}}{q}}\left[\log p(\mathbf{x}) \underset{\underset{\text{importance weight}}{\uparrow}}{\frac{p(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}}\right]$$

Jensen bound on importance sampled estimate:

$$\ell(\theta) = \log \mathbb{E}_{\mathbf{z}\sim q}\left[\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right] \geq \mathbb{E}_{\mathbf{z}\sim q}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right]$$

Suggests more accurate importance sampling:

$$\ell(\theta) = \log \mathbb{E}_{\mathbf{z}_1\ldots\mathbf{z}_K \overset{\text{iid}}{\sim} q}\left[\frac{1}{K}\sum_k \frac{p(\mathbf{x},\mathbf{z}_k)}{q(\mathbf{z}_k)}\right] \geq \mathbb{E}_{\mathbf{z}_1\ldots\mathbf{z}_K \overset{\text{iid}}{\sim} q}\left[\log \frac{1}{K}\sum_k \frac{p(\mathbf{x},\mathbf{z}_k)}{q(\mathbf{z}_k)}\right]$$

Tighter bound, and reparametrisation friendly, but as $K \to \infty$ the signal for learning amortised q grows weaker so VAE learning doesn't always improve.

Normalising flows

$$\mathcal{F}(q,\theta) = \langle \log p(\mathbf{x},\mathbf{z}|\theta)\rangle_q - \langle \log q(\mathbf{z})\rangle_q$$

To evaluate $\mathcal{F}$ (or its gradients) we need to be able to find expectations wrt q (e.g. by Monte Carlo) and evaluate the log-density – usually restricts us to tractable inferential families.

Consider defining a recognition model $q(\mathbf{z})$ implicitly by:

$$\begin{aligned} \mathbf{z}_0 &\sim q_0(\cdot;\mathbf{x}) && \leftarrow \text{ fixed, tractable, e.g. } \mathcal{N}(\mathbf{x}, I) \\ \mathbf{z} &= f_K(f_{K-1}(\ldots f_1(\mathbf{z}_0))) && \leftarrow\ f_k \text{ smooth, invertible, parametrised by } \phi \end{aligned}$$

Then

$$\begin{aligned} \langle F(\mathbf{z})\rangle_q &= \langle F(f_K(f_{K-1}(\ldots f_1(\mathbf{z}_0))))\rangle_{q_0} \\ \log q(\mathbf{z}) &= \log q_0(f_1^{-1}(f_2^{-1}(\ldots f_K^{-1}(\mathbf{z})))) - \sum_k \log|\nabla f_k| \end{aligned}$$

where the second result applies from repeated transformations of variables

$$\mathbf{z}_k = f_k(\mathbf{z}_{k-1}) \ \Rightarrow\ q(\mathbf{z}_k) = q(f_k^{-1}(\mathbf{z}_k))\left|\frac{\partial \mathbf{z}_{k-1}}{\partial \mathbf{z}_k}\right| = q(f_k^{-1}(\mathbf{z}_k))\,|\nabla f_k(\mathbf{z}_{k-1})|^{-1}$$

Normalising flows

So, given a sample $\mathbf{z}_0^s \overset{\text{iid}}{\sim} q_0(\cdot;\mathbf{x})$:

$$\mathcal{F}(q,\theta) \approx \frac{1}{S}\sum_s \log p(\mathbf{x}, f_K(\ldots f_1(\mathbf{z}_0^s)))) + \mathbf{H}[q_0] + \frac{1}{S}\sum_s\sum_k \left|\nabla f_k(f_{k-1}(\ldots f_1(\mathbf{z}_0^s)))\right|$$

and we can compute gradients of this expression wrt θ and ϕ.

Useful fs (from Rezende & Mohammed 2015):

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u}h(\mathbf{w}^{\mathsf{T}}\mathbf{z}+b) \qquad \Rightarrow |\nabla f| = \left|1 + \mathbf{u}^{\mathsf{T}}\psi(\mathbf{z})\right| \qquad \psi(\mathbf{z}) = h'(\mathbf{w}^{\mathsf{T}}\mathbf{z}+b)\mathbf{w}$$

$$f(\mathbf{z}) = \mathbf{z} + \frac{\beta}{\alpha + |\mathbf{z}-\mathbf{z}_0|} \qquad \Rightarrow |\nabla f| = [1+\beta h]^{d-1}[1+\beta h + \beta h' r]$$

$$r = |\mathbf{z}-\mathbf{z}_0|,\ h = \frac{1}{\alpha + r}$$

Both can be cascaded to give a flexible variational family.

DDC Helmholtz machine

A (loosely) neurally inspired idea. Define q as an unnormalisable exponential family with a large set of sufficient statistics

$$q(\mathbf{z}) \propto e^{\sum_i \eta_i \psi_i(\mathbf{z})}$$

and parametrise by mean parameters $\boldsymbol{\mu} = \langle \phi(\mathbf{z})\rangle$: Distributed distributional code (DDC).

Train recognition model using sleep samples:

$$\begin{aligned} \boldsymbol{\mu} &= \langle \boldsymbol{\psi}(\mathbf{z})\rangle_q = f(\mathbf{x};\phi) \\ \Delta\phi &\propto \sum_s (\boldsymbol{\psi}(\mathbf{z}_s) - f(\mathbf{x}_s;\phi))\nabla_\phi f(\mathbf{x}_s;\phi) \end{aligned}$$

Also learn linear approximation $\nabla \log p(\mathbf{x},\mathbf{z}|\theta) \approx A\boldsymbol{\psi}(\mathbf{z})$

$$A = \left(\sum_s \nabla \log p(\mathbf{x}_s,\mathbf{z}_s|\theta)\boldsymbol{\psi}(\mathbf{z}_s)\right)^{\mathsf{T}}\left(\sum_s \boldsymbol{\psi}(\mathbf{z}_s)\boldsymbol{\psi}(\mathbf{z}_s)^{\mathsf{T}}\right)^{-1}$$

Then

$$\langle \nabla \log p(\mathbf{x},\mathbf{z})\rangle_q \approx A\langle\boldsymbol{\psi}(\mathbf{z})\rangle_q \approx Af(\mathbf{x},\phi)$$

Approach can be generalised to an infinite dimensional $\boldsymbol{\psi}$ using the kernel trick.

Amortised Learning

If we aren't actually interested in inference, we can short-circuit general recognition and compute expectations for learning directly.

$$\nabla_\theta \ell(\theta) = \partial_\theta \mathcal{F}(q^*, \theta) = \partial_\theta \left\langle \log p(\mathcal{X}, \mathcal{Z}|\theta) \right\rangle_{q^*} = \left\langle \partial_\theta \log p(\mathcal{X}, \mathcal{Z}|\theta) \right\rangle_{p(\mathcal{Z}|\mathcal{X},\theta)}$$

Suggests a wake-sleep approach:

- Sample $\{\mathbf{x}_s, \mathbf{z}_s\} \sim p(\mathcal{X}, \mathcal{Z}|\theta^k)$.
- Train regressor $\hat{J}_{\theta^k} : \mathbf{x}_s \mapsto \nabla_\theta \log p(\mathbf{x}_s, \mathbf{z}_s|\theta)|_{\theta^k}$
  (or, for specific regressors, $\mapsto \log p(\mathbf{x}_s, \mathbf{z}_s|\theta^k)$ and differentiate prediction)
- Set $\theta^{k+1} = \theta^k + \eta \sum_i \hat{J}_{\theta^k}(\mathbf{x}_i)$
  (or $= \theta^k + \eta \sum_i \nabla_\theta \hat{J}_\theta(\mathbf{x}_i)|_{\theta^k}$).

Derivate form works for (kernel/GP) regression for which regressor is linear in targets.

For conditional exponential family models

$$\log p(\mathcal{X}, \mathcal{Z}|\theta) = \eta(\mathbf{z}, \theta)^{\mathsf{T}} \mathbf{T}(\mathbf{x}) - \Phi(\mathbf{z}, \theta) + \log p(\mathbf{z}|\theta)$$

$$\Rightarrow \left\langle \log p(\mathcal{X}, \mathcal{Z}|\theta) \right\rangle_{q^*} = \left\langle \eta(\mathbf{z}, \theta) \right\rangle_{q^*}^{\mathsf{T}} \mathbf{T}(\mathbf{x}) - \left\langle \Phi(\mathbf{z}, \theta) + \log p(\mathbf{z}|\theta) \right\rangle_{q^*}$$

and regressors can be trained to functions of $\mathbf{z}$ alone, with $T(\mathbf{x})$ then evaluated on (wake-phase) data.

Generative models

In practice, much of the VAE and related work has used a common generative model:

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$$
$$\mathbf{x} \sim \mathcal{N}(\mathbf{g}(\mathbf{z}; \boldsymbol{\theta}), \psi I)$$

where g is a neural network.

- Overcomplicated: if $\dim(\mathbf{z})$ is large enough the optimal solution has $\psi \to 0$, $q(\mathbf{z}; \mathbf{x}) \to \delta(\mathbf{z} - f(\mathbf{x}, \phi))$. In effect, the generative model learns a flow to transform a normal density to the target.
- Oversimplified: if $\dim(\mathbf{z})$ is small, this is just non-linear PCA!

Interesting latent representations are likely to require more structured generative models. Recent work has approached such models in both VAE and DDC frameworks.

Structured VAEs

Consider a model where $p(\mathcal{Z}|\theta)$ has tractable joint exponential-family potentials and

$$p(\mathcal{X}|\mathcal{Z}, \Gamma) = \prod_i p(\mathbf{x}_i|\mathbf{z}_i, \gamma_i)$$

are intractable (say neural net + normal) cond ind observations. γ_i might be the same for all i.

Consider factored variational inference $q(\mathcal{Z}) = \prod_i q_i(\mathbf{z}_i)$. With no further constraint,

$$\log q_i^*(\mathbf{z}_i) \underset{+C}{=} \left\langle \log p(\mathcal{Z}, \mathcal{X}) \right\rangle_{q_{\neg i}} \underset{+C}{=} \left\langle \log p(\mathbf{z}_i|\mathcal{Z}_{\neg i}) + \log p(\mathbf{x}_i|\mathbf{z}_i) \right\rangle_{q_{\neg i}}$$
$$\underset{+C}{=} \left\langle \boldsymbol{\eta}_{\neg i} \right\rangle_{q_{\neg i}}^{\mathsf{T}} \boldsymbol{\psi}_i(\mathbf{z}_i) + \log p(\mathbf{x}_i|\mathbf{z}_i)$$

where we have exploited the exponential-family form of $p(\mathcal{Z})$. $\boldsymbol{\psi}_i$ are effective suff stats – including log normalisers of children in a DAG; $\boldsymbol{\eta}_{\neg i}$ is a function of $\mathcal{Z}_{\neg i}$.

Now, choose the parametric form $q_i(\mathbf{z}_i) = e^{\tilde{\boldsymbol{\eta}}_i^{\mathsf{T}} \boldsymbol{\psi}_i(\mathbf{z}_i) - \Phi_i(\tilde{\boldsymbol{\eta}}_i)}$. Constrained optimum has form

$$\log q_i^*(\mathbf{z}_i) \underset{+C}{=} \left\langle \boldsymbol{\eta}_{\neg i} \right\rangle_{q_{\neg i}}^{\mathsf{T}} \boldsymbol{\psi}_i(\mathbf{z}_i) + \boldsymbol{\rho}(\mathbf{x}_i)^{\mathsf{T}} \boldsymbol{\psi}_i(\mathbf{z}_i)$$

for some $\mathbf{x}_i$-dependent natural parameter. Introduce recognition models:

$$\boldsymbol{\rho}(\mathbf{x}_i) = f_i(\mathbf{x}_i, \phi_i)$$

Recognition function f_i might be same for all i if all likelihoods are the same (*e.g.* HMM).

Structured VAE learning

Now, the free-energy can be written as a function of parameters and recognition parameters:

$$\mathcal{F}(\theta, \Gamma, \{\phi_i\}) = \left\langle \sum_i \log p(\mathbf{x}_i|\mathbf{z}_i, \gamma_i) + \log p(\mathcal{Z}|\theta) \right\rangle_{q(\mathcal{Z};\theta,\{\phi_i\})} + \sum_i \mathbf{H}[q_i]$$
$$= \sum_i \underbrace{\left\langle \log p(\mathbf{x}_i|\mathbf{z}_i, \gamma_i) \right\rangle_{q_i(\mathbf{z}_i;\theta,\phi_i)} + \mathbf{H}[q_i]}_{\mathcal{F}_i} + \left\langle \log p(\mathcal{Z}|\theta) \right\rangle_{q(\mathcal{Z};\theta,\{\phi_i\})}$$

Updates on θ are just as for tractable model.

To update each ϕ_i and γ_i, find $\left\langle \boldsymbol{\eta}_{\neg i} \right\rangle_{q_{\neg i}}$ to give the "prior". Generate reparametrised samples $\mathbf{z}_i^s \sim q_i$. Then

$$\frac{\partial}{\partial \gamma_i} \mathcal{F}_i = \sum_s \nabla_{\gamma_i} \log p(\mathbf{x}_i, \mathbf{z}_i^s; \gamma_i)$$
$$\frac{\partial}{\partial \phi_i} \mathcal{F}_i = \sum_s \frac{\partial}{\partial \mathbf{z}_i^s} \left(\log p(\mathbf{x}_i, \mathbf{z}_i^s; \gamma_i) - \log q(\mathbf{z}_i^s; \mathbf{f}(\mathbf{x}_i)) \right) \frac{d\mathbf{z}_i^s}{d\phi} + \frac{\partial}{\partial \mathbf{f}(\mathbf{x}_i)} \log q(\mathbf{z}_i^s; \mathbf{f}(\mathbf{x}_i)) \frac{d\mathbf{f}(\mathbf{x}_i)}{d\phi}$$

as for the standard VAE.

. . . just a brief survey of a subset of current ideas.

A few things we hope you've learned in this course . . .

- Exponential families are your friends.
- Latent variable models and conditional independence to uncover structured representations.
- Free-energies, maximum likelihood, variational approximation theory and variational Bayes.
- Message passing exploits conditional independence.
- A rich toolkit of approximations, that you can compose in novel and useful ways.
- A theory of many approximations that helps ensure you understand their use and limitations (and may help derive new approaches).