Probabilistic & Unsupervised Learning

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 2017

Variational methods

- Our treatment of variational methods has (except EP) emphasised ‘natural’ choices of variational family – most 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.
- Parametric variational methods might extend our reach.
  Define a parametric family of posterior approximations $q(\mathcal{Y}; \rho)$.
  The constrained (approximate) variational E-step becomes:

$$q(\mathcal{Y}) := \operatorname*{argmax}_{q \in \{q(\mathcal{Y};\rho)\}} \mathcal{F}\big(q(\mathcal{Y}), \theta^{(k-1)}\big) \quad \Rightarrow \quad \rho^{(k)} := \operatorname*{argmax}_{\rho} \mathcal{F}\big(q(\mathcal{Y}; \rho), \theta^{(k-1)}\big)$$

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

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

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

Optimising the variational parameters

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

- In some special cases, the expectations of the log-joint under $q(\mathcal{Y};\rho)$ can be expressed in closed form, but these are rare.

Optimising the variational parameters

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

- In some special cases, the expectations of the log-joint under $q(\mathcal{Y}; \rho)$ can be expressed in closed form, but these are rare.
- Otherwise we might seek to follow $\nabla_\rho \mathcal{F}$.

Optimising the variational parameters

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

- In some special cases, the expectations of the log-joint under $q(\mathcal{Y}; \rho)$ 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 $q(\mathcal{Y}, \rho)$ as a function of ρ – not simple.

Optimising the variational parameters

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

- In some special cases, the expectations of the log-joint under $q(\mathcal{Y}; \rho)$ 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 $q(\mathcal{Y}, \rho)$ as a function of ρ – not simple.
- At least three solutions:

Optimising the variational parameters

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

- In some special cases, the expectations of the log-joint under $q(\mathcal{Y}; \rho)$ 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 $q(\mathcal{Y}, \rho)$ as a function of ρ – not simple.
- At least three solutions:
  - “Score-based” gradient estimate, and Monte-Carlo (Ranganath et al. 2014).

Optimising the variational parameters

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

- In some special cases, the expectations of the log-joint under $q(\mathcal{Y};\rho)$ can be expressed in closed form, but these are rare.
- Otherwise we might seek to follow $\nabla_\rho \mathcal{F}$.
- Naively, this requires evaluating a high-dimensional expectation wrt $q(\mathcal{Y},\rho)$ 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).

Optimising the variational parameters

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

- In some special cases, the expectations of the log-joint under $q(\mathcal{Y}; \rho)$ can be expressed in closed form, but these are rare.
- Otherwise we might seek to follow $\nabla_\rho \mathcal{F}$.
- Naively, this requires evaluating a high-dimensional expectation wrt $q(\mathcal{Y}, \rho)$ 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:

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

Score-based gradient estimate

We have:

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

Now,

$$
\begin{aligned}
\nabla_\rho \log P(\mathcal{X},\mathcal{Y}|\theta) &= 0 && \text{(no direct dependence)} \\
\int d\mathcal{Y}\, q(\mathcal{Y};\rho)\nabla_\rho \log q(\mathcal{Y};\rho) &= \nabla_\rho \int d\mathcal{Y}\, q(\mathcal{Y};\rho) = 0 && \text{(always normalised)} \\
\nabla_\rho q(\mathcal{Y};\rho) &= q(\mathcal{Y};\rho)\nabla_\rho \log q(\mathcal{Y};\rho)
\end{aligned}
$$

So,

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

Score-based gradient estimate

We have:

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

Now,

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

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

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

So,

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

Reduced gradient of expectation to expectation of gradient – easier to compute.

Factorisation

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

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

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

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

So the second term is proportional to $\langle \nabla_{\rho_i} \log q(\mathcal{Y}_i;\rho_i)\rangle_{q(\mathcal{Y}_i)}$, which $= 0$ as before. So expectations are only needed wrt $q(\bar{\mathcal{Y}}_i)$ → Message passing!

Sampling

So the “black-box” variational approach is as follows:

- Choose a parametric (factored) variational family $q(\mathcal{Y}) = \prod_i q(\mathcal{Y}_i; \rho_i)$.
- Initialise factors.
- Repeat to convergence:
  - **Stochastic VE-step**. For each *i*:
    - Sample from $q(\bar{\mathcal{Y}}_i)$ and estimate expected gradient $\nabla_{\rho_i}\mathcal{F}$.
    - Update ρ_i along gradient.
  - **Stochastic M-step**. For each *i*:
    - Sample from each $q(\bar{\mathcal{Y}}_i)$.
    - 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. However, even for large models (such as HMMs), we often work with multiple data draws (e.g. multiple strings) and each instance requires its own variational optimisation.

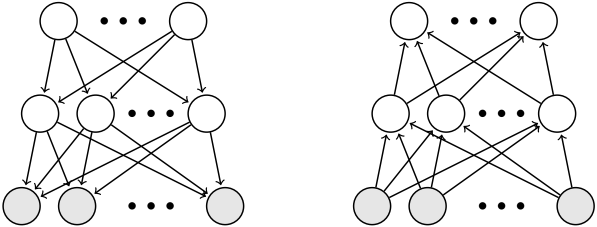
Suppose we have fixed length vectors $\{(\mathbf{x}_i, \mathbf{y}_i)\}$ ($\mathbf{y}$ is still latent).

- Optimal variational distribution $q^*(\mathbf{y}_i)$ depends on $\mathbf{x}_i$.
- Learn this mapping (in parametric form): $q\big(\mathbf{y}_i; f(\mathbf{x}_i; \rho)\big)$.
- f is a general function approximator (a GP, neural network or similar) parametrised by ρ, trained to map $\mathbf{x}_i$ to the variational parameters of $q(\mathbf{y}_i)$.
- The mapping function f is called a recognition model.
- This is approach is now sometimes 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 Bernouilli are just probabilities):

  $$\hat{\mathbf{y}}_i = f(\mathbf{x}_i; \rho)$$

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

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

The Helmholtz Machine

- Can sample **y** from recognition model rather than just evaluate means.
  - Expectations in free-energy can be computed directly rather than by mean substitution.
  - In higherarchical 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.

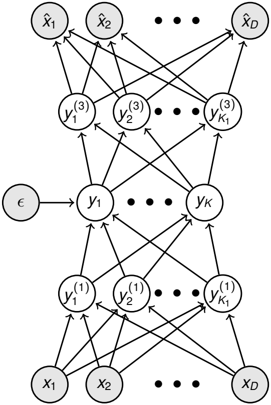
- More general approach is to train f to yield expected sufficent statistics of ExpFam $q(\mathbf{y})$:

$$\Delta\rho \propto \sum_s (\mathbf{s}_q(\mathbf{y}_s) - \mathbf{f}(\mathbf{x}_s;\rho))\nabla_\rho f(\mathbf{x}_s;\rho)$$

  Current work extends this to extremely flexible (non-normalisable) exponential families.

- Sleep phase learning minimises $\mathbf{KL}[p_\theta(\mathbf{y}|\mathbf{x})\|q(\mathbf{y};f(\mathbf{x},\rho))]$. 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 y_i and $\epsilon_i^s \sim \mathcal{N}(0,1)$, then $y_i^s = \mu_i + \sigma_i \epsilon_i^s$.
- Now generative and recognition parameters can be trained together by gradient descent (backprop), holding $\boldsymbol{\epsilon}^s$ fixed.

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

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

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

Variational Autoencoders

- Frozen samples ϵ^s can be redrawn to avoid overfitting.
- May be possible to evaluate entropy and log $P(\mathbf{y})$ without sampling, reducing variance.
- Differentiable reparametrisations are available for a number of different distributions.
- Conditional $P(\mathbf{x}|\mathbf{y}, \theta)$ 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

- Dynamical VAE (to train RNNs) – “draw” network.
- Train proposal networks for particle filtering.
- Importance weighted VAE.
- DDC Helmholt machines – arbitrary (non-normalisable) ExpFam posteriors.
- . . .