Composing graphical models with neural networks for structured representations and fast inference

Matt Johnson, David Duvenaud, Alex Wiltschko, Bob Datta, Ryan Adams
<table>
<thead>
<tr>
<th>Probabilistic graphical models</th>
<th>Deep learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ structured representations</td>
<td>– neural net “goo”</td>
</tr>
<tr>
<td>+ priors and uncertainty</td>
<td>– difficult parameterization</td>
</tr>
<tr>
<td>+ data and computational efficiency</td>
<td>– can require lots of data</td>
</tr>
<tr>
<td>– rigid assumptions may not fit</td>
<td>+ flexible</td>
</tr>
<tr>
<td>– feature engineering</td>
<td>+ feature learning</td>
</tr>
<tr>
<td>– top-down inference</td>
<td>+ recognition networks</td>
</tr>
</tbody>
</table>
Modeling idea: graphical models on latent variables, neural network models for observations
unsupervised learning

supervised learning

unsupervised learning
\[ \pi = \begin{bmatrix} \pi^{(1)} & \pi^{(2)} & \pi^{(3)} \end{bmatrix} \]

\[ z_{t+1} \sim \pi^{(z_t)} \]

\[ x_{t+1} = A^{(z_t)} x_t + B^{(z_t)} u_t \quad u_t \overset{\text{iid}}{\sim} \mathcal{N}(0, I) \]
\[ y_t \mid x_t, \gamma \sim \mathcal{N}(\mu(x_t; \gamma), \Sigma(x_t; \gamma)) \]
conjugate prior on global variables
exponential family on local variables
any prior on observation parameters
neural network observation model
Inference?
\[ p(x \mid \theta) \text{ is linear dynamical system} \]
\[ p(y \mid x, \theta) \text{ is linear-Gaussian} \]
\[ p(\theta) \text{ is conjugate prior} \]

\[ \mathcal{L}[q(\theta)q(x)] \triangleq \mathbb{E}_{q(\theta)q(x)} \left[ \log \frac{p(\theta, x, y)}{q(\theta)q(x)} \right] \]

\[ q(\theta) \leftrightarrow \eta_{\theta} \quad q(x) \leftrightarrow \eta_{x} \]
Proposition (natural gradient SVI of Hoffman et al. 2013)

\[ \nabla \mathcal{L}_{\text{SVI}}(\eta_{\theta}) = \eta_{\theta}^0 + \mathbb{E}_{q^*(x)}(t_{xy}(x, y), 1) - \eta_{\theta} \]
\( p(x \mid \theta) \) is linear dynamical system
\( p(y \mid x, \theta) \) is linear-Gaussian
\( p(\theta) \) is conjugate prior

\[
\mathcal{L}(\eta_\theta, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(x)} \left[ \log \frac{p(\theta, x, y)}{q(\theta)q(x)} \right]
\]

\( \eta_x^*(\eta_\theta) \triangleq \arg \max_{\eta_x} \mathcal{L}(\eta_\theta, \eta_x) \quad \mathcal{L}_{SVI}(\eta_\theta) \triangleq \mathcal{L}(\eta_\theta, \eta_x^*(\eta_\theta)) \)

Proposition (natural gradient SVI of Hoffman et al. 2013)

\[
\tilde{\nabla} \mathcal{L}_{SVI}(\eta_\theta) = \eta_\theta^0 + \sum_{n=1}^{N} \mathbb{E}_{q^*(x_n)}(t_{xy}(x_n, y_n), 1) - \eta_\theta
\]
Step 1: compute evidence potentials

Step 1: compute evidence potentials

Step 1: compute evidence potentials

Step 2: run fast message passing

Step 3: compute natural gradient

Inference?
SVAEs: recognition networks output conjugate potentials, then apply fast graphical model algorithms
\( p(x \mid \theta) \) is a linear dynamical system
\( p(y \mid x, \gamma) \) is a neural network decoder
\( p(\theta) \) is a conjugate prior, \( p(\gamma) \) is generic

\[
\mathcal{L}(\eta_{\theta}, \eta_{\gamma}, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)} \left[ \log \frac{p(\theta, \gamma, x)p(y \mid x, \gamma)}{q(\theta)q(\gamma)q(x)} \right]
\]

\[
\eta_x^*(\eta_{\theta}, \eta_{\gamma}) \triangleq \arg \max_{\eta_x} \mathcal{L}(\eta_{\theta}, \eta_{\gamma}, \eta_x)
\]

\[
\mathcal{L}_{SVI}(\eta_{\theta}, \eta_{\gamma}) \triangleq \mathcal{L}(\eta_{\theta}, \eta_{\gamma}, \eta_x^*(\eta_{\theta}, \eta_{\gamma}))
\]
\[ \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)} \left[ \log \frac{p(\theta, \gamma, x)p(y | x, \gamma)}{q(\theta)q(\gamma)q(x)} \right] \]

\[ \hat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)} \left[ \log \frac{p(\theta, \gamma, x)\exp\{\psi(x; y, \phi)\}}{q(\theta)q(\gamma)q(x)} \right] \]

where \( \psi(x; y, \phi) \) is a conjugate potential for \( p(x | \theta) \)

\[ \eta_x^*(\eta_\theta, \phi) \triangleq \arg \max_{\eta_x} \hat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) \quad \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \triangleq \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi)) \]
Step 1: apply recognition network
Step 1: apply recognition network
Step 1: apply recognition network

Step 2: run fast PGM algorithms

Step 3: sample, compute flat grads

Step 4: compute natural gradient
$$q^*(x) \triangleq \text{arg max}_{q(x)} \mathcal{L}[q(\theta)q(x)]$$

$$q^*(x) \triangleq \mathcal{N}(x | \mu(y; \phi), \Sigma(y; \phi))$$

$$q^*(x) \triangleq ?$$

**Natural gradient SVI**

- expensive for general obs.
+ optimal local factor
+ exploits conj. graph structure
+ arbitrary inference queries
+ natural gradients

**Variational autoencoders**

+ fast for general obs.
- suboptimal local factor
- \(\phi\) does all local inference
- limited inference queries
- no natural gradients

**Structured VAEs** [1]

+ fast for general obs.
\pm optimal given conj. evidence
+ exploits conj. graph structure
+ arbitrary inference queries
+ natural gradients on \(\eta|\theta\)

\[
\pi = \begin{bmatrix}
\pi^{(1)} & - & - \\
- & \pi^{(2)} & - \\
- & - & \pi^{(3)} 
\end{bmatrix}
\]

\[z_{t+1} \sim \pi^{(z_t)}\]

\[x_{t+1} = A^{(z_t)} x_t + B^{(z_t)} u_t \quad u_t \overset{iid}{\sim} \mathcal{N}(0, I)\]
rearing up
fall from rear
grooming
Limitations and future work

- How expressive is latent linear structure?
  - word embeddings [1], analogical reasoning in image models
  - SVAE can use nonlinear latent structure

- PGMs get complicated
  - SVAE keeps complexity modular

- model-based reinforcement learning
- automatic structure search [2,3]
- semi-supervised applications

---

Thanks!

github.com/mattjj/svae
\[
\begin{align*}
\mu_t(y_t; \phi_\mu) \\
J_{t,t}(y_t; \phi_D) \\
J_{t,t+1}(y_t, y_{t+1}; \phi_B)
\end{align*}
\]
SVAEs can use any inference network architecture

Per-variable recognition nets allow arbitrary inference queries
The graph shows the negative log likelihood ($-\mathcal{L}$) over iterations for different gradient methods. The graph compares the performance of natural gradient and flat gradient methods. The natural gradient method exhibits a smoother and more stable decrease in $-\mathcal{L}$ compared to the flat gradient method, which shows more fluctuations and appears to stagnate at certain points (marked by 'X' in the graph).
Fact (conjugate graphical models are easy)

The local variational parameter $\eta^*_x(\eta_\theta, \phi)$ is easy to compute.

Proposition (log evidence lower bound)

$$\max_{\eta_z} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_z)$$

log $p(y)$

max $\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_z)$

$\mathcal{L}_{SVAE}(\eta_\theta, \eta_\gamma, \phi)$

$\eta_\theta, \eta_\gamma$

Proposition (reparameterization trick)

Estimate $\nabla_{\eta_\gamma, \phi} \mathcal{L}_{SVAE}(\eta_\theta, \eta_\gamma, \phi)$ with samples $\hat{\gamma} \sim q(\gamma)$ and $\hat{x} \sim q^*(x \mid \phi)$ via

$$\mathcal{L}_{SVAE}(\eta_\theta, \eta_\gamma, \phi) \approx \log p(y \mid \hat{x}, \hat{\gamma}) - KL(q(\theta)q(\gamma)q^*(x \mid \phi) \parallel p(\theta, \gamma, x))$$

Proposition (easy natural gradient)

$$\tilde{\nabla}_{\eta_\theta} \mathcal{L}_{SVAE}(\eta_\theta, \eta_\gamma, \phi) = (\eta_\theta^0 + \mathbb{E}_{q^*(x \mid \phi)}(t_x(x), 1) - \eta_\theta) + (\nabla_{\eta_z} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta^*_x(\eta_\theta, \phi)), 0)$$