Latent Variable Models
CS598PS MLSP

Cem Subakan

University of Illinois at Urbana-Champaign

November 10’th, 2017
Basic definition

- LVMs are multivariate probability distributions. Of the form:

\[ p(x, h|\theta) \]

- \( x \): observations (data)
- \( h \): latent (hidden) variables
- \( \theta \): parameters

- Examples:

HMM, Linear Dynamical System

Mixture Model, PCA, ICA
Things to consider

- Goal of this lecture: To give a general sense on Bayesian Machine Learning.
- It is a nice framework to understand how models are related to each other.
- I will mostly look things at modeling. (Not too much details on optimization/inference techniques, theoretical analysis)
Examples

- Mixture of HMMs

\[ r_1, n \rightarrow r_2, n \rightarrow \cdots \rightarrow r_T, n \rightarrow h_n \rightarrow x_1, n \rightarrow x_2, n \rightarrow \cdots \rightarrow x_{T, n} \]

\[ n = 1 \ldots N \]

- Factorial HMM

\[ r^1_1 \rightarrow r^1_2 \rightarrow \cdots \rightarrow r^1_T \]

\[ x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_T \]

- Switching HMMs

\[ h_1 \rightarrow h_2 \rightarrow \cdots \rightarrow h_T \rightarrow r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_T \rightarrow x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_T \]

- HMM with Mixture observations

\[ h_1 \rightarrow h_2 \rightarrow \cdots \rightarrow h_T \rightarrow r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_T \rightarrow x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_T \]
More Examples

- **Convolutive Neural Nets**

  \[ \hat{x}_t = \sigma \left( \sum_{t' = 1}^{T'} w_{t'} x_{t-t'} \right) . \]

- **Recurrent Nets**

  \[ \hat{h}_t = r( h_{t-1}, x_{t-1} ), \hat{x}_t = f( h_{t-1} ) . \]
(I am stealing this image from Taylan Cemgil)
Main Questions in LVMs
  Mixture Model Example

Exploring some models

Monte Carlo Epilogue
Plan

Main Questions in LVMs
  Mixture Model Example

Exploring some models

Monte Carlo Epilogue
Main Questions in LVMs

- **Learning/Parameter Estimation:**

  \[
  \max_{\theta} p(x, h|\theta)
  \]

  This usually is a non-convex problem.
  - This is okay (but not okay).
Main Questions in LVMs

▶ Learning/Parameter Estimation:

$$\max_{\theta} p(x, h|\theta)$$

This usually is a non-convex problem.
▶ This is okay (but not okay).

▶ Inference:

$$p(h|x, \theta) = \frac{p(x|h, \theta)p(h|\theta)}{\int p(x|h, \theta)p(h|\theta)dh}$$

The integral in denominator is not always tractable.
▶ We don’t like this. We use approximations such as Monte-Carlo sampling, or variational techniques.
Mixture Model Example

Model:

\[ h_n \sim \text{Categorical}(\pi) \]
\[ x_n|h_n \sim \mathcal{N}(x; \mu_h, \sigma^2 I), \text{ for } n \in \{1, \ldots N\} \]

- \( h_n \in \{1, \ldots, K\} \), cluster indicators.
- \( x_n \in \mathbb{R}^L \), observed data items.
- \( \theta = \{\mu_1, \mu_2, \ldots, \mu_K\} \) parameters/cluster centers.
Find cluster indicators $\hat{h}_{1:N}$ and parameters $\hat{\theta}$ such that:

$$\hat{h}_{1:N}, \hat{\theta} = \arg \max_{h_{1:N}, \theta} p(x_{1:N} | h_{1:N}, \theta)$$
Learning Variant 1 for GMM

- Find cluster indicators $\hat{h}_{1:N}$ and parameters $\hat{\theta}$ such that:

$$\hat{h}_{1:N}, \hat{\theta} = \arg \max_{h_{1:N}, \theta} p(x_{1:N} | h_{1:N}, \theta)$$

- Write down log-likelihood:

$$\log p(x_{1:N}, h_{1:N} | \theta) = \log \prod_{n=1}^{N} p(x_n | h_n, \theta) p(h_n | \theta)$$

$$= \log \prod_{n=1}^{N} \left( \prod_{k=1}^{K} \mathcal{N}(x_n; \mu_k, \sigma^2 I)_{[h_n=k]} \times \prod_{k=1}^{K} \mu_{[h_n=k]} \right)$$

$$= + \sum_{n=1}^{N} \left( \sum_{k=1}^{K} [h_n = k] \left( \frac{-\|x_n - \mu_k\|^2}{2\sigma^2} + \log \pi_k \right) \right)$$
Learning Variant 1 for GMM

- Algorithm: Fix $\theta$, update $h$. Fix $h$, update $\theta$, repeat until convergence (and fix $\pi_k = 1/K$).
Learning Variant 1 for GMM

- Algorithm: Fix $\theta$, update $h$. Fix $h$, update $\theta$, repeat until convergence (and fix $\pi_k = 1/K$).

- Update $\mu_{k'}$: compute the gradient while $h_{1:N}$ is fixed:

$$
\frac{\partial \log p(x_{1:N}, h_{1:N}|\theta)}{\partial \mu_k} = \frac{\partial \sum_{n=1}^N \left( \sum_{k=1}^K [h_n = k] \left( \frac{-\|x_n - \mu_k\|^2}{2\sigma^2} + \log \pi_k \right) \right)}{\partial \mu_{k'}} 
= \sum_{n=1}^N [h_n = k'] \frac{(x_n - \mu_{k'})}{\sigma^2} = \sum_{n=1}^N [h_n = k'] \frac{x_n}{\sigma^2} - [h_n = k'] \frac{\mu_{k'}}{\sigma^2}
$$

set the gradient equal to 0, solve for $\mu_{k'} \rightarrow \hat{\mu}_{k'} = \frac{\sum_{n=1}^N [h_n=k'] x_n}{\sum_{n=1}^N [h_n=k']}$. 

- Looks like a familiar algorithm?
Learning Variant 1 for GMM

- Algorithm: Fix $\theta$, update $h$. Fix $h$, update $\theta$, repeat until convergence (and fix $\pi_k = 1/K$).

- Update $\mu_{k'}$: compute the gradient while $h_{1:N}$ is fixed:

$$
\frac{\partial \log p(x_{1:N}, h_{1:N}|\theta)}{\partial \mu_k} = \frac{\partial \sum_{n=1}^{N} \left( \sum_{k=1}^{K} [h_n = k] \left( \frac{-\|x_n - \mu_k\|^2}{2\sigma^2} + \log \pi_k \right) \right)}{\partial \mu_{k'}}
$$

$$
= \sum_{n=1}^{N} [h_n = k'] \frac{(x_n - \mu_{k'})}{\sigma^2} = \sum_{n=1}^{N} [h_n = k'] \frac{x_n}{\sigma^2} - [h_n = k'] \frac{\mu_{k'}}{\sigma^2}
$$

set the gradient equal to 0, solve for $\mu_{k'} \rightarrow \hat{\mu}_{k'} = \frac{\sum_{n=1}^{N} [h_n = k'] x_n}{\sum_{n=1}^{N} [h_n = k']}$.

- Update $h_{1:N}$ while $\mu_{k'}$ is fixed:

$$
\hat{h}_n = \arg \max_{h_n} \log p(x_n, h_n|\theta) = \arg \min_k \|x_n - \mu_k\|^2, \quad \text{we therefore assign } h_n \text{ as the index of the mean closest to } x_n.
$$
Learning Variant 1 for GMM

- Algorithm: Fix $\theta$, update $h$. Fix $h$, update $\theta$, repeat until convergence (and fix $\pi_k = 1/K$).

- Update $\mu_{k'}$: compute the gradient while $h_{1:N}$ is fixed:

$$\frac{\partial \log p(x_{1:N}, h_{1:N} | \theta)}{\partial \mu_k} = \frac{\partial \sum_{n=1}^N \left( \sum_{k=1}^K [h_n = k] \left( \frac{-\|x_n - \mu_k\|^2}{2\sigma^2} + \log \pi_k \right) \right)}{\partial \mu_{k'}}$$

$$= \sum_{n=1}^N [h_n = k'] \frac{x_n - \mu_{k'}}{\sigma^2} = \sum_{n=1}^N [h_n = k'] \frac{x_n}{\sigma^2} - [h_n = k'] \frac{\mu_{k'}}{\sigma^2}$$

set the gradient equal to 0, solve for $\mu_{k'} \rightarrow \hat{\mu}_{k'} = \frac{\sum_{n=1}^N [h_n = k'] x_n}{\sum_{n=1}^N [h_n = k']}$. 

- Update $h_{1:N}$ while $\mu_{k'}$ is fixed:

$$\hat{h}_n = \arg \max_{h_n} \log p(x_n, h_n | \theta) = \arg \min_k \|x_n - \mu_k\|^2,$$

we therefore assign $h_n$ as the index of the mean closest to $x_n$.

- Looks like a familiar algorithm?
Learning Variant 2 for GMM

- Find cluster indicator parameters $\hat{\theta}$ while integrating out hidden variables, such that:

\[
\hat{\theta} = \arg\max_{\theta} p(x_{1:N} | \theta) \\
= \arg\max_{\theta} \sum_{h_{1:N}} p(x_{1:N}, h_{1:N} | \theta)
\]
Learning Variant 2 for GMM

- Find cluster indicator parameters $\hat{\theta}$ while integrating out hidden variables, such that:

$$\hat{\theta} = \arg\max\theta p(x_{1:N} | \theta)$$

$$= \arg\max\theta \sum_{h_{1:N}} p(x_{1:N}, h_{1:N} | \theta)$$

- Write down log-likelihood:

$$\log p(x_{1:N} | \theta) = \log \sum_{h_{1:N}} \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} q(h_{1:N}) = \log E_q \left[ \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} \right]$$

$$\geq VLB := E_q \left[ \log \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} \right] =^+ E_q [\log p(x_{1:N}, h_{1:N} | \theta)]$$

$$=^+ \sum_{n=1}^{N} \left( \sum_{k=1}^{K} E_q[h_n = k] \left( -\frac{\|x_n - \mu_k\|^2}{2\sigma^2} + \log \pi_k \right) \right)$$
Learning Variant 2 for GMM

- Algorithm: Fix $\theta$, update $q$. Fix $q$, update $\theta$, repeat until convergence (and fix $\pi_k = 1/K$).

$$\text{Update } \mu_k': \text{compute the gradient while } h_1: N \text{ is fixed:}$$

$$\frac{\partial VLB}{\partial \mu_k'} = \frac{\partial}{\partial \mu_k'} \sum_{n=1}^{N} \left( \sum_{k=1}^{K} E[h_n = k] \left( -\|x_n - \mu_k\|^2 + \log \pi_k \right) \right)$$

$$= \frac{1}{N} \sum_{n=1}^{N} \left[ h_n = k' \right] \left( x_n - \mu_k' \right) \sigma^2$$

Set the gradient equal to 0, solve for $\mu_k' \rightarrow \hat{\mu}_k' = \frac{1}{N} \sum_{n=1}^{N} E[h_n = k'] x_n \sigma^2 - E[h_n = k'] \mu_k' \sigma^2$.

- Update $q(1:N)$ while $\mu_k' \text{ is fixed. Notice that:}$

$$\text{VLB} = E_q[\log p(x_1:N, h_1:N | \theta)] q(h_1:N)] = \text{KL}(q(h) \| p(x,h | \theta)).$$

What is the variational distribution that would minimize this divergence?
Algorithm: Fix $\theta$, update $q$. Fix $q$, update $\theta$, repeat until convergence (and fix $\pi_k = 1/K$).

Update $\mu_{k'}$: compute the gradient while $h_{1:N}$ is fixed:

$$\frac{\partial VLB}{\partial \mu_{k'}} = \frac{\partial}{\partial \mu_{k'}} \left( \sum_{n=1}^{N} E[h_n = k] \left( -\frac{\|x_n - \mu_k\|^2}{2\sigma^2} + \log \pi_k \right) \right)$$

$$= \sum_{n=1}^{N} \left[ h_n = k' \right] \frac{x_n - \mu_{k'}}{\sigma^2} = \sum_{n=1}^{N} E[h_n = k'] \frac{x_n}{\sigma^2} - E[h_n = k'] \frac{\mu_{k'}}{\sigma^2}$$

set the gradient equal to 0, solve for $\mu_{k'} \rightarrow \hat{\mu}_{k'} = \frac{\sum_{n=1}^{N} E[h_n = k'] x_n}{\sum_{n=1}^{N} E[h_n = k']}$.
Learning Variant 2 for GMM

- Algorithm: Fix $\theta$, update $q$. Fix $q$, update $\theta$, repeat until convergence (and fix $\pi_k = 1/K$).

- Update $\mu_{k'}$: compute the gradient while $h_{1:N}$ is fixed:

\[
\frac{\partial VLB}{\partial \mu_{k'}} = \frac{\partial}{\partial \mu_{k'}} \sum_{n=1}^{N} \left( \sum_{k=1}^{K} \mathbb{E}[h_n = k] \left( -\frac{\|x_n - \mu_k\|^2}{2\sigma^2} + \log \pi_k \right) \right)
\]

\[
= \sum_{n=1}^{N} [h_n = k'] \left( \frac{x_n - \mu_{k'}}{\sigma^2} \right) = \sum_{n=1}^{N} \mathbb{E}[h_n = k'] \frac{x_n}{\sigma^2} - \mathbb{E}[h_n = k'] \frac{\mu_{k'}}{\sigma^2}
\]

set the gradient equal to 0, solve for $\mu_{k'} \rightarrow \hat{\mu}_{k'} = \frac{\sum_{n=1}^{N} \mathbb{E}[h_n=k']x_n}{\sum_{n=1}^{N} \mathbb{E}[h_n=k']}$. 

- Update $q(h_{1:N})$ while $\mu_{k'}$ is fixed. Notice that:

\[
VLB = \mathbb{E}_q \left[ \log \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} \right] = KL(q(h) \| p(x, h | \theta)).
\]

What is the variational distribution that would minimize this divergence?
Learning Variant 2 for GMM - optimal $q(h)$

- See board for derivation.
Learning Variant 2 for GMM - optimal $q(h)$

See board for derivation.

$$
\frac{\partial L}{\partial q} = \frac{\partial}{\partial q} \left( \int q(h) \log p(x, h|\theta) dh - \int q(h) \log q(h) dh + \lambda \left( \int q(h) dh - 1 \right) \right) \\
= \log p(x, h) - \log q(h) - 1 + \lambda = 0 \\
\rightarrow q(h) = \frac{p(x, h|\theta)}{\exp(1 - \lambda)} \\
\rightarrow \exp(1 - \lambda) = p(x|\theta) \\
\rightarrow q(h) = \frac{p(x, h|\theta)}{p(x|\theta)} = p(h|x, \theta)$$
Learning Variant 2 for GMM - optimal $q(h)$

- See board for derivation.

\[
\frac{\partial \mathcal{L}}{\partial q} = \frac{\partial}{\partial q} \left( \int q(h) \log p(x, h|\theta) dh - \int q(h) \log q(h) dh + \lambda \left( \int q(h) dh - 1 \right) \right) \\
= \log p(x, h) - \log q(h) - 1 + \lambda = 0
\]

\[
\rightarrow q(h) = \frac{p(x, h|\theta)}{\exp(1 - \lambda)} \\
\rightarrow \exp(1 - \lambda) = p(x|\theta)
\]

\[
\rightarrow q(h) = \frac{p(x, h|\theta)}{p(x|\theta)} = p(h|x, \theta)
\]

- Note that in our case $q(h) = q(h_{1:N}) = \prod_n q(h_n)$, where

\[
q(h_n = k) = \frac{p(x_n, h_n = k|\theta)}{p(x_n|\theta)} = \frac{\pi_k \mathcal{N}(x_n; \mu_k, \sigma^2 I)}{\sum_{k'} \pi_{k'} \mathcal{N}(x_n; \mu_{k'}, \sigma^2 I)}
\]
Randomly initialize $\mu_{1:K}$.

while Not converged do

E-step:

if ICM then

$\hat{h}_n = \arg \max_{h_n} \log p(x_n, h_n | \theta) = \arg \min_k \|x_n - \mu_k\|_2^2$

else if EM then

$q(h_n = k) = \frac{\pi_k \mathcal{N}(x_n; \mu_k, \sigma^2 I)}{\sum_{k'} \pi_{k'} \mathcal{N}(x_n; \mu_{k'}, \sigma^2 I)}$

end if

M-step:

if ICM then

$\hat{\mu}_{k'} = \frac{\sum_{n=1}^{N} [h_n = k'] x_n}{\sum_{n=1}^{N} [h_n = k']}$

else if EM then

$\hat{\mu}_{k'} = \frac{\sum_{n=1}^{N} \mathbb{E}_q[h_n = k'] x_n}{\sum_{n=1}^{N} \mathbb{E}_q[h_n = k']}$

end if

end while
Learning Variant 3 for GMM - Going Full Bayesian

- Model:

- $\mu_k \sim N(\mu_k; 0, \sigma_0^2 I)$, for $k \in \{1, \ldots, K\}$
- $h_n \sim \text{Categorical}(\pi)$
- $x_n|h_n \sim N(x; \mu_h, \sigma^2 I)$, for $n \in \{1, \ldots, N\}$

- $h_n \in \{1, \ldots, K\}$, cluster indicators.
- $x_n \in \mathbb{R}^L$, observed data items.
- $\theta = \{\mu_1, \mu_2, \ldots, \mu_K\}$ parameters/cluster centers. But we are not treating these guys as parameters anymore.
Inference for Variant 3 GMM

- Approximate the posterior distribution \( p(h, \theta|x) \), with a variational distribution \( \hat{q} \) such that,

\[
\hat{q}(h, \theta) = \arg \min_q KL(q(h, \theta)\|p(x, h, \theta))
\]

- We will use the mean field approximation. English: \( q(h, \theta) = q_h(h)q_\theta(\theta) \).
Inference for Variant 3 GMM

- Approximate the posterior distribution \( p(h, \theta | x) \), with a variational distribution \( \hat{q} \) such that,

\[
\hat{q}(h, \theta) = \arg \min_q KL(q(h, \theta) \| p(x, h, \theta))
\]

- We will use the mean field approximation. English: \( q(h, \theta) = q_h(h)q_\theta(\theta) \).

- Algorithm: Fix \( q_h \), update \( q_\theta \). We can show that: (via same process as the EM case)

\[
\hat{q}_\theta(\theta) = \arg \min_{q_\theta} KL(q_h(h)q_\theta(\theta) \| p(x, h, \theta)) = \frac{1}{Z} \exp \left( \mathbb{E}_{q_h} [\log p(x, h, \theta)] \right)
\]

where \( Z \) is the normalization constant. Similarly,
Approximate the posterior distribution \( p(h, \theta|x) \), with a variational distribution \( \hat{q} \) such that,

\[
\hat{q}(h, \theta) = \arg \min_q KL(q(h, \theta)\|p(x, h, \theta))
\]

We will use the mean field approximation. English: \( q(h, \theta) = q_h(h)q_{\theta}(\theta) \).

Algorithm: Fix \( q_h \), update \( q_{\theta} \). We can show that: (via same process as the EM case)

\[
\hat{q}_{\theta}(\theta) = \arg \min_{q_{\theta}} KL(q_h(h)q_{\theta}(\theta)\|p(x, h, \theta)) = \frac{1}{Z} \exp \left( \mathbb{E}_{q_h} [\log p(x, h, \theta)] \right)
\]

where \( Z \) is the normalization constant. Similarly,

Fix \( q_{\theta} \), update \( q_h \):

\[
\hat{q}_h(h) = \arg \min_{q_h} KL(q_h(h)q_{\theta}(\theta)\|p(x, h, \theta)) = \frac{1}{Z} \exp \left( \mathbb{E}_{q_{\theta}} [\log p(x, h, \theta)] \right)
\]
Inference for Variant 3 GMM - Specifics:

\[
\log \hat{q}_\theta(\mu_k) =^+ \mathbb{E}_{q\mu}[\log p(x, h, \mu_k)] \\
= ^+ \sum_{n=1}^N \mathbb{E}[h_n = k] \frac{-(x_n^T x_n - 2x_n^T \mu_k + \mu_k^T \mu_k)}{2\sigma^2} - \frac{\mu_k^T \mu_k}{2\sigma_0^2} \\
= ^+ \sum_{n=1}^N \mathbb{E}[h_n = k] 2x_n^T \mu_k - (\sum_{n=1}^N \mathbb{E}[h_n = k] + \sigma^2) \mu_k^T \mu_k \\
= ^+ \log \mathcal{N} \left( \mu_k; \frac{\sum_n \mathbb{E}[h_n = k] x_n}{\sum_n \mathbb{E}[h_n = k] + \sigma^2}, \frac{\sigma^2 \sigma_0^2}{\sum_n \mathbb{E}[h_n = k] + \sigma^2} \right)
\]
Inference for Variant 3 GMM - Specifics:

\[
\log \hat{q}_\theta(\mu_k) = \mathbb{E}_{q_h}[\log p(x, h, \mu_k)]
\]

\[
= + \sum_{n=1}^{N} \mathbb{E}[h_n = k] \frac{-(x_n^T x_n - 2x_n^T \mu_k + \mu_k^T \mu_k)}{2\sigma^2} - \frac{\mu_k^T \mu_k}{2\sigma_0^2}
\]

\[
= + \sum_{n=1}^{N} \mathbb{E}[h_n = k]2x_n^T \mu_k - (\sum_{n=1}^{N} \mathbb{E}[h_n = k] + \sigma^2)\mu_k^T \mu_k
\]

\[
= + \log \mathcal{N}\left(\mu_k; \frac{\sum_n \mathbb{E}[h_n = k] x_n}{\sum_n \mathbb{E}[h_n = k] + \sigma^2}, \frac{\sigma^2 \sigma_0^2}{\sum_n \mathbb{E}[h_n = k] + \sigma^2}\right)
\]
Inference for Variant 3 GMM - Specifics:

\[
\log \hat{q}_h(h_n = k) = \left( \frac{\mathbb{E}[-(x_n - \mu_k)^2]}{2\sigma^2} + \log \pi_k \right)
\]

\[
\rightarrow \hat{q}_h(h_n = k) = \frac{\exp \left( \frac{\mathbb{E}[-(x_n - \mu_k)^2]}{2\sigma^2} + \log \pi_k \right)}{\sum_k \exp \left( \frac{\mathbb{E}[-(x_n - \mu_k)^2]}{2\sigma^2} + \log \pi_k \right)}
\]
Inference for Variant 3 GMM - Why:

- Variational lower bound:
  \[
  \int p(x, h, \theta) dhd\theta \geq \mathbb{E}_{q(h)q(\theta)}[\log p(x, h, \theta)] - \mathbb{E}_{q(h)q(\theta)}[\log q(h) + \log q(\theta)]
  \]

- You can use VLB to determine $K$: (plot taken from Bishop, 2006)
Inference for Variant 3 GMM - Why:

- Variational lower bound:
  \[
  \int p(x, h, \theta)dhd\theta \geq \mathbb{E}_{q(h)q(\theta)}[\log p(x, h, \theta)] - \mathbb{E}_{q(h)q(\theta)}[\log q(h) + \log q(\theta)]
  \]

- You can use VLB to determine $K$: (plot taken from Bishop, 2006)

- But admittedly the algebra gets tiring.
Variant 4 for GMM - Going Ultra Bayesian

Model:

\[
\begin{aligned}
\pi_k &\sim \text{Dirichlet}(1/K, \ldots, 1/K) \\
\mu_k &\sim \mathcal{N}(\mu; 0, \sigma_0^2 I), \text{ for } k \in \{1, \ldots, K\} \\
h_n &\sim \text{Categorical}(\pi) \\
x_n|h_n &\sim \mathcal{N}(x; \mu_h, \sigma^2 I), \text{ for } n \in \{1, \ldots, N\}
\end{aligned}
\]

- \( h_n \in \{1, \ldots, K\} \), cluster indicators.
- \( x_n \in \mathbb{R}^L \), observed data items.
- \( \theta = \{\mu_1, \mu_2, \ldots, \mu_K\} \cup \{\pi\} \)
Integrate out the parameters, sample from the full conditionals:

\[
p(h_n = k | h_{-n}, x_{1:N}) \propto \int p(x_{1:N}, h_{1:N}, \pi, \mu_{1:K}) d\mu_{1:K} d\pi
\]

\[
\propto \frac{\alpha/K + N_k^{-n}}{\alpha + N - 1} p(x_n | \{x_m : m \neq n, h_m = k\})
\]

And, sample from these full conditionals!
Integrate out the parameters, sample from the full conditionals:

\[ p(h_n = k | h_{-n}, x_{1:N}) \propto \int p(x_{1:N}, h_{1:N}, \pi, \mu_{1:K}) d\mu_{1:K} d\pi \]

\[ \propto \frac{\alpha / K + N_k^{-n}}{\alpha + N - 1} p(x_n | \{x_m : m \neq n, h_m = k\}) \]

Take \( K \) to infinity:

\[ p(h_n = k, k \text{ occupied} | h_{-n}, x_{1:N}) \propto \frac{N_k^{-n}}{\alpha + N - 1} p(x_n | \{x_m : m \neq n, h_m = k\}) \]

\[ p(h_n = k, k \text{ empty} | h_{-n}, x_{1:N}) \propto \frac{\alpha}{\alpha + N - 1} p(x_n) \]

And, sample from these full conditionals!
Collapsed Gibbs sampling in Infinite GMM

Top left: Histogram of observed data, Top right: Samples from full conditional of $h_{1:N}$, Bottom: Histogram of $K$
Collapsed Gibbs sampling in Infinite GMM

Top left: Histogram of observed data, Top right: Samples from full conditional of $h_{1:N}$, Bottom: Histogram of $K$
Collapsed Gibbs sampling in Infinite GMM

Top left: Histogram of observed data, Top right: Samples from full conditional of $h_{1:N}$, Bottom: Histogram of $K$
Top left: Histogram of observed data, Top right: Samples from full conditional of $h_{1:N}$, Bottom: Histogram of $K$
Collapsed Gibbs sampling in Infinite GMM

Top left: Histogram of observed data, Top right: Samples from full conditional of $h_{1:N}$, Bottom: Histogram of $K$
Collapsed Gibbs sampling in Infinite GMM

Top left: Histogram of observed data, Top right: Samples from full conditional of $h_{1:N}$, Bottom: Histogram of $K$
Collapsed Gibbs sampling in Infinite GMM

Top left: Histogram of observed data, Top right: Samples from full conditional of $h_{1:N}$, Bottom: Histogram of $K$
Collapsed Gibbs sampling in Infinite GMM

Top left: Histogram of observed data, Top right: Samples from full conditional of $h_{1:N}$, Bottom: Histogram of $K$
Collapsed Gibbs sampling in Infinite GMM

Top left: Histogram of observed data, Top right: Samples from full conditional of $h_{1:N}$, Bottom: Histogram of $K$
What’s the point of going all Bayesian then

- (Automatic) Model Selection for Unsupervised Learning
What’s the point of going all Bayesian then

- (Automatic) Model Selection for Unsupervised Learning
- Model Averaging (Model plays all its cards)
What’s the point of going all Bayesian then

- (Automatic) Model Selection for Unsupervised Learning
- Model Averaging (Model plays all its cards)
- Principled way of regularization
What’s the point of going all Bayesian then

- (Automatic) Model Selection for Unsupervised Learning
- Model Averaging (Model plays all its cards)
- Principled way of regularization
- All of these 4 variants are extendable for other models. We can play with:
  - Distribution of $h$.
  - Impose structure on $h$.
  - We can change the conditional distribution $p(x|h, \theta)$. (Application decides)
  - We can play with how we do inference and learning.
What’s the point of going all Bayesian then

- (Automatic) Model Selection for Unsupervised Learning
- Model Averaging (Model plays all its cards)
- Principled way of regularization
- All of these 4 variants are extendable for other models. We can play with:
  - Distribution of $h$.
  - Impose structure on $h$.
  - We can change the conditional distribution $p(x|h, \theta)$. (Application decides)
  - We can play with how we do inference and learning.

- (Little controversial - but best part of it) You don’t need to read paper/take ML classes if you learn these.
Plan

Main Questions in LVMs
  Mixture Model Example

Exploring some models

Monte Carlo Epilogue
Probabilistic PCA

▶ Model: [Bishop, Tipping 1999]

\[ h_n \sim \mathcal{N}(h_n; 0, I) \]
\[ x_n| h_n \sim \mathcal{N}(x; Wh_n + \mu, \sigma^2 I), \text{ for } n \in \{1, \ldots N\} \]

▶ \( h_n \in \mathbb{R}^K \), latent variables (embeddings).
▶ \( x_n \in \mathbb{R}^L \), observed data items.
▶ \( \theta = \{W, \mu, \sigma^2\} \)
Probabilistic PCA

- Model: [Bishop, Tipping 1999]

\[ h_n \sim \mathcal{N}(h_n; 0, I) \]
\[ x_n|h_n \sim \mathcal{N}(x; Wh_n + \mu, \sigma^2 I), \text{ for } n \in \{1, \ldots, N\} \]

- \( h_n \in \mathbb{R}^K \), latent variables (embeddings).
- \( x_n \in \mathbb{R}^L \), observed data items.
- \( \theta = \{W, \mu, \sigma^2\} \)

Note that \( p(x) = \int p(x|h)p(h)dh = \mathcal{N}(\mu, WW^\top + \sigma^2 I) \). Then ML estimate \( \hat{W}_{ML} = U_K(\Lambda_K - \sigma^2 I)^{1/2} \). \( U_q, \Lambda_K \) are the first \( K \) eigenvectors-eigenvalues of the covariance matrix. Familiar?
Factor Analysis

- Model: [Bartholomew 1987]

\[ h_n \sim \mathcal{N}(h_n; 0, I) \]
\[ x_n|h_n \sim \mathcal{N}(x; W h_n + \mu, \Psi), \text{ for } n \in \{1, \ldots, N\} \]

- \( h_n \in \mathbb{R}^K \), latent variables (embeddings).
- \( x_n \in \mathbb{R}^L \), observed data items.
- \( \theta = \{W, \mu, \Psi\} \)
NMF

- Model: [Lee, Seung 1999]

\[ h_n, x_n \sim \mathcal{P}(x_n; Wh_n), \text{ for } n \in \{1, \ldots, N\} \]

- \( h_n \in \mathbb{R}^{\geq 0, K} \), latent variables (embeddings).
- \( x_n \in \mathbb{R}^{\geq 0, L} \), observed data items.
- \( \theta = \{W \geq 0\} \)
Model:

\[ h_n \sim \mathcal{N}(h_n; 0, I) \]
\[ x_n|h_n \sim \mathcal{N}(x; \phi(t_n)h_n, \sigma^2 I), \text{ for } n \in \{1, \ldots N\} \]

- \( h_n \in \mathbb{R}^K \), latent variables (embeddings).
- \( \phi(t_n) \in \mathbb{R}^{L_2 \times K} \), the design matrix
- \( t_n \in \mathbb{R}^{L_1} \), input variable.
- \( x_n \in \mathbb{R}^{>0,L_2} \), observed data items.
Neural Network Regression

- **Model:**

  \[
  n = 1 \ldots N
  \]

  \[
  x_n \mid h_n \sim \mathcal{N}(x_n; f_\theta(t_n), \sigma^2 I), \text{ for } n \in \{1, \ldots N\}
  \]

- **\(f_\theta(t_n) : \mathbb{R}^{L_1} \to \mathbb{R}^{L_2}\)**, the neural network! (Convolutional, recurrent, feed-forward what have you)

- **\(t_n \in \mathbb{R}^{L_1}\)**, input variable.

- **\(x_n \in \mathbb{R}^{L_2}\)**, observed data items.

- **\(\theta\)**, neural network parameters.
Neural Network Regression

▶ Model:

\[ t_n \rightarrow x_n \]

\[ x_n | h_n \sim \mathcal{N}(x_n; f_\theta(t_n), \sigma^2 I), \text{ for } n \in \{1, \ldots, N\} \]

▶ \( f_\theta(t_n) : \mathbb{R}^{L_1} \rightarrow \mathbb{R}^{L_2} \), the neural network! (Convolutive, recurrent, feed-forward what have you)

▶ \( t_n \in \mathbb{R}^{L_1} \), input variable.

▶ \( x_n \in \mathbb{R}^{L_2} \), observed data items.

▶ \( \theta \), neural network parameters.

Notice that this is not a Latent Variable Model. Why?
Here’s a neural net LVM - Variational Autoencoder

- Model: [Kingma, Welling 2013]

\[ h_n \sim \mathcal{N}(h_n; 0, I) \]
\[ x_n | h_n \sim \mathcal{N}(x; f_\theta(h_n), \sigma^2 I), \text{ for } n \in \{1, \ldots N\} \]

- \( h_n \in \mathbb{R}^K \), latent variables (embeddings).
- \( f_\theta(h_n) : \mathbb{R}^K \to \mathbb{R}^L \), the forward mapping.
- \( x_n \in \mathbb{R}^{L_2} \), observed data items.
- \( \theta \), neural network parameters.
Tired of IID models? HMMs

- Model:

\[
\begin{align*}
\begin{array}{c}
\text{Model:} \\
\begin{array}{c}
h_1 \\
\downarrow \\
x_1
\end{array} & \rightarrow \\
\begin{array}{c}
h_2 \\
\downarrow \\
x_2
\end{array} & \rightarrow \quad \ldots \rightarrow \\
\begin{array}{c}
h_T \\
\downarrow \\
x_T
\end{array}
\end{array}
\end{align*}
\]

\[
h_n | h_{n-1} \sim \text{Discrete}(A(:, h_{n-1})
\]

\[
x_n | h_n \sim p(x_n | h_n, O)
\]

- \(h_n \in \{1, \ldots, K\}\), latent variables (embeddings).
- \(x_n \in \mathbb{R}^L\), observed data items.
- \(O\), the emission matrix, \(A \in \mathbb{R}^{K \times K}\), the transition matrix.
- \(\theta = \{O, A\}\).
- Learning is conceptually all the same. Just that E-step is little non-trivial.
Tired of IID models? Linear Dynamical System

Model:

\[ h_1 \rightarrow h_2 \rightarrow \cdots \rightarrow h_T \]

\[ x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_T \]

\[ h_n \mid h_{n-1} \sim \mathcal{N}(h_n; Ah_{n-1}, \Sigma_1) \]
\[ x_n \mid h_n \sim \mathcal{N}(x_n; Oh_n, \Sigma_2) \]

- \( h_n \in \mathbb{R}^K \), latent variables (embeddings).
- \( x_n \in \mathbb{R}^L \), observed data items.
- \( O \in \mathbb{R}^{L \times K} \), the emission matrix, \( A \in \mathbb{R}^{K \times K} \), the transition matrix.
- \( \theta = \{O, A\} \).
What about other cases? HMM

- A chain structure: (HMMs, LDS, etc.)

\[
p(h_t|x_{1:T}) \propto p(h_t, x_{1:T})
= p(h_t, x_{1:t}) p(x_{t+1:T}|h_t)
= \alpha(h_t) \beta(h_t)
\]

where,

\[
\alpha(h_t) = p(x_t|h_t) \sum_{h_{t-1}} p(h_t|h_{t-1}) p(x_{t-1}|h_{t-1}) \ldots p(x_2|h_2) \sum_{h_1} p(h_2|h_1) p(x_1|h_1) p(h_1)
\]

\[
\beta(h_t) = \sum_{h_{t+1}} p(h_t|h_{t+1}) p(x_{t+1}|h_{t+1}) \ldots \sum_{h_T} p(h_T|h_{T-1}) p(x_T|h_T) \frac{1}{\beta(h_T)}
\]

\[
\beta(h_t) = \frac{\beta(h_T)}{\beta(h_{T-1})}\frac{\beta(h_{T-1})}{\beta(h_{T-2})}\ldots\frac{\beta(h_1)}{\beta(h_0)}\frac{1}{\beta(h_{t+1})}
\]
Inference in HMMs

- \( \alpha(h_t) \) are “forward messages”. \( \beta(h_t) \) are “backward messages”. One forward pass and one backward pass is sufficient since,

\[
p(h_t|x_{1:T}) \propto p(h_t, x_{1:T})
\]

\[
= p(h_t, x_{1:t}) p(x_{t+1:T} | h_t)
\]

\[
= \alpha(h_t) \beta(h_t)
\]

- Traditionally (EE traditions), \( \alpha_{1:T} \) is known as the filtering density. \( \gamma_{1:T} := \alpha_{1:T} \ast \beta_{1:T} \) is the smoothing density.
Forward Pass in Action
The joint distribution is defined with clique “potentials”.

\[ p(h_{1:K}, x_{1:J}|\theta) = \frac{1}{Z(\theta)} \prod_{C \in G} \exp(\theta^T \phi(x_C, h_C)) \]
Tired of directed graphs? MRFs

- The joint distribution is defined with clique “potentials”.

\[
p(h_{1:K}, x_{1:J}|\theta) = \frac{1}{Z(\theta)} \prod_{C \in G} \exp(\theta^T \phi(x_C, h_C))
\]

- Example: (An image segmentation model)

\[
\phi(x_C, h_C) = \phi_1(h_i, h_{N(i)}) + \phi_2(x_i, h_i) \\
= \theta_1 \textbf{1}_{[h_i=h_{N(i)}]} + \theta_2 \textbf{1}_{[h_i\neq h_{N(i)}]} \\
+ \sum_{l,k} \theta_{3,i,k} \textbf{1}_{[x_i=l][h_i=k]}
\]

\[
Z(\theta) = \int \prod_{C \in G} \exp(\theta^T \phi(x_C, h_C)) dx_{1:J} dh_{1:K}
\]

The notorious partition function!
How to do inference in general graphs?

- Forward-Backward algorithm is an instance of “Belief Propagation”.

**Example**

\[ p(h_{1:4}) = \frac{1}{Z} \psi(h_1, h_2) \psi(h_2, h_4) \psi(h_2, h_3) \]

\[ p(h_2) \propto \sum_{h_1, h_3, h_4} \psi(h_1, h_2) \psi(h_2, h_4) \psi(h_2, h_3) \]

\[ = \left( \sum_{h_1} \psi(h_1, h_2) \right) \left( \sum_{h_4} \psi(h_2, h_4) \right) \left( \sum_{h_3} \psi(h_2, h_3) \right) \]

\[ \begin{aligned}
    &\text{m}_1 \rightarrow 2 \\
    &\text{m}_4 \rightarrow 2 \\
    &\text{m}_3 \rightarrow 2
\end{aligned} \]
Example

\[ p(h_{1:4}) = \frac{1}{Z} \psi(h_1, h_2) \psi(h_2, h_4) \psi(h_2, h_3) \]

\[
p(h_1) \propto \sum_{h_2, h_3, h_4} \psi(h_1, h_2) \psi(h_2, h_4) \psi(h_2, h_3)
\]

\[
= \sum_{h_2} \psi(h_1, h_2) \left( \sum_{h_4} \psi(h_2, h_4) \right) \left( \sum_{h_3} \psi(h_2, h_3) \right)
\]

\[
= \sum_{h_2} \psi(h_1, h_2) m_{4 \rightarrow 2}(h_2) m_{3 \rightarrow 2}(h_2)
\]
BP, summarized

- Compute all messages for all possible \((i, j)\) pairs with,

\[
\textbf{m}_{i \rightarrow j}(h_j) = \sum_{h_i} \psi(h_i, h_j) \prod_{l \in N(i) \setminus j} \textbf{m}_{l \rightarrow i}(h_i)
\]

Incoming Messages to node \(i\)

Figure is taken from Yedidia et al. 2001.
BP, summarized

- Compute all messages for all possible \((i, j)\) pairs with,

\[
m_{i \rightarrow j}(h_j) = \sum_{h_i} \psi(h_i, h_j) \prod_{l \in N(i) \setminus j} m_{l \rightarrow i}(h_l)
\]

Incoming Messages to node \(i\)

![Diagram showing message passing between nodes]

Figure is taken from Yedidia et al. 2001.

- The Belief for node \(i\) is \(B(h_i) = p(h_i) = \prod_{j \in N(i)} m_{j \rightarrow i}(h_i)\).
BP, summarized

- Compute all messages for all possible \((i, j)\) pairs with,

\[
m_{i \rightarrow j}(h_j) = \sum_{h_i} \psi(h_i, h_j) \prod_{l \in N(i) \setminus j} m_{l \rightarrow i}(h_i)
\]

Incoming Messages to node \(i\)

\[
\bigodot_i \rightarrow \bigodot_j = \sum_{i \in N(i) \setminus j}
\]

Figure is taken from Yedidia et al. 2001.

- The Belief for node \(i\) is \(B(h_i) = p(h_i) = \prod_{j \in N(i)} m_{j \rightarrow i}(h_i)\).

- One pass from leaves to root and one pass from leaves to root, and we are done.
BP, summarized

- Compute all messages for all possible \((i, j)\) pairs with,

\[
m_{i \rightarrow j}(h_j) = \sum_{h_i} \psi(h_i, h_j) \prod_{l \in N(i) \setminus j} m_{l \rightarrow i}(h_i)
\]

Figure is taken from Yedidia et al. 2001.

- The Belief for node \(i\) is \(B(h_i) = p(h_i) = \prod_{j \in N(i)} m_{j \rightarrow i}(h_i)\).

- One pass from leaves to root and one pass from leaves to root, and we are done.

- BP converges to true beliefs in trees. What about general graphs?
We can still run BP on a loopy graph. It converges (most of the time) in practice!

Example:

(Left) Original Image, (Center) Noisy Image, (Right) Image cleared with BP
Main Questions in LVMs
   Mixture Model Example

Exploring some models

Monte Carlo Epilogue
As we have seen, obtaining the posterior can be difficult.
Monte Carlo Methods for Inference

- As we have seen, obtaining the posterior can be difficult.
- Monte Carlo methods are about drawing samples from the posterior.
Monte Carlo Methods for Inference

- As we have seen, obtaining the posterior can be difficult.
- Monte Carlo methods are about drawing samples from the posterior.
- One instance of these methods is Gibbs sampling. (Special case of Metropolis-Hastings algorithm)
Gibbs Sampling

- This is a Markov Chain Monte Carlo algorithm.
Gibbs Sampling

- This is a Markov Chain Monte Carlo algorithm.
- **The key idea:** Drawn samples form a Markov chain. And, the stationary distribution is the posterior!
Gibbs Sampling

- This is a Markov Chain Monte Carlo algorithm.
- **The key idea:** Drawn samples form a Markov chain. And, the stationary distribution is the posterior!
- Gibbs sampling is an instance of Metropolis-Hastings sampling with a particular transition kernel.

**Input:** A model structure with variables $h_{1:N}$

**Output:** Samples $h_{1:N}^{1:E}$

```plaintext
while You are not satisfied, (say $e \leq E$) do
  for $n = 1 : N$ do
    $h_n \sim p(h_n|h_{1:N}^{-n})$
  end for
end while
```
Let’s derive a Gibbs sampler

- \( p(h_n|h_{1:n}^-) \) is known as the full conditional. It is generally easy to derive/sample from. An example:

\[
p(x_{1:4}) = \frac{1}{Z} \psi_{1,2}(x_1, x_2)\psi_{2,4}(x_2, x_4)\psi_{1,3}(x_1, x_3)\psi_{3,4}(x_3, x_4)
\]

- Here’s our Gibbs sampler! \( others \) is essentially the variables that have functional dependence. It is known as the Markov blanket.
Sampling from a 2D Gaussian with Gibbs sampling. Figures are taken from C.Bishop’s and D.Barber’s books.
Conclusions

- If you learn Bayesian machine learning/graphical models, you don’t need to learn anything. (semi-true)
- Great Pedagogical Tool. (true)
- Great to build unsupervised models. / Model Selection.
- Things I wanted to but couldn’t talk about: Gaussian Processes (Probabilistic Kernel Methods).
- Active Research Fields: Stochastic Variational Inference, Probabilistic Programming (to avoid going through tedious algebra), Efficient Sampling Methods, Likelihood-free methods (GANs - next time)