Probabilistic Graphical Models
Learning with partial observations

Siamak Ravanbakhsh  
Fall 2019
Learning objectives

- different types of missing data
- learning with missing data and hidden vars:
  - directed models
  - undirected models
- develop an intuition for expectation maximization
  - variational interpretation
Two settings for partial observations

- missing data
  - each instance in $\mathcal{D}$ is missing some values
Two settings for partial observations

- missing data
  - each instance in $\mathcal{D}$ is missing some values
- hidden variables
  - variables that are never observed

image credit: Murphy's book
Two settings for partial observations

- missing data
  - each instance in $\mathcal{D}$ is missing some values
- hidden variables
  - variables that are never observed

Latent variable models

- observations have common cause
- widely used in machine learning

Image credit: Murphy’s book
Missing data

observation mechanism:

- generate the data point \( X = [X_1, \ldots, X_D] \)
- decide the values to observe \( O_X = [1, 0, \ldots, 0, 1] \)

\[ \text{hide} \rightarrow \text{observe} \]
Missing data

observation mechanism:

- generate the data point \( X = [X_1, \ldots, X_D] \)
- decide the values to observe \( O_X = [1, 0, \ldots, 0, 1] \)
- observe \( X_o \) while \( X_h \) is missing \( (X = [X_h; X_o]) \)
Missing data

observation mechanism:

• generate the data point \( X = [X_1, \ldots, X_D] \)
• decide the values to observe \( O_X = [1, 0, \ldots, 0, 1] \)

\[ X = [X_h; X_o] \]

• observe \( X_o \) while \( X_h \) is missing \( (X = [X_h; X_o]) \)

**missing completely at random (MCAR)**

\[ P(X, O_X) = P(X)P(O_X) \]

\[ p(x) = \theta^x(1-\theta)^{1-x} \]

throw to generate
Missing data

observation mechanism:

- generate the data point  \( X = [X_1, \ldots, X_D] \)
- decide the values to observe  \( O_X = [1, 0, \ldots, 0, 1] \)

\( \uparrow \uparrow \) hide observe

- observe \( X_o \) while \( X_h \) is missing \( (X = [X_h; X_o]) \)

- missing completely at random (MCAR) \( P(X, O_X) = P(X)P(O_X) \)

\[
p(x) = \theta^x (1 - \theta)^{1-x} \quad \text{throw to generate}
\]

\[
p(o) = \psi^o (1 - \psi)^{1-o} \quad \text{throw to decide show/hide}
\]
Learning with MCAR

missing completely at random (MCAR) \hspace{1cm} P(X, O) = P(X)P(O)

\[ p(x) = \theta^x (1 - \theta)^{1-x} \] \hspace{1cm} \text{throw to generate}

\[ p(o) = \psi^o (1 - \theta)^{1-o} \] \hspace{1cm} \text{throw to decide show/hide}
**Learning with MCAR**

**missing completely at random (MCAR)**  \[ P(X, O) = P(X)P(O) \]

\[
p(x) = \theta^x (1 - \theta)^{1-x} \quad \text{throw to generate}
\]

\[
p(o) = \psi^o (1 - \theta)^{1-o} \quad \text{throw to decide show/hide}
\]

**objective:** learn a model for \(X\), from the data  \[ \mathcal{D} = \{x_o^{(1)}, \ldots, x_o^{(M)}\} \]

each \(x_o\) may include values for a different subset of vars.
Learning with MCAR

**missing completely at random (MCAR)**

\[ P(X, O) = P(X)P(O) \]

- \( p(x) = \theta^x (1 - \theta)^{1-x} \) throw to generate
- \( p(o) = \psi^o (1 - \theta)^{1-o} \) throw to decide show/hide

**objective:** learn a model for \( X \), from the data \( D = \{x_o^{(1)}, \ldots, x_o^{(M)}\} \)

each \( x_o \) may include values for a different subset of vars.

since \( P(X, O) = P(X)P(O) \), we can **ignore the obs. patterns**

**optimize:**

\[ \ell(D, \theta) = \sum_{x_o \in D} \log \sum_{x_h} p(x_o, x_h) \]
A more general criteria

**missing at random (MAR)** \( O_X \perp X_h | X_o \)

if there is information about the obs. pattern \( O_X \) in \( X_h \) then it is also in \( X_o \)
A more general criteria

missing at random (MAR) \( O_X \perp X_h | X_o \)

if there is information about the obs. pattern \( O_X \) in \( X_h \) then it is also in \( X_o \)

throw the thumb-tack twice \( X = [X_1, X_2] \)
if \( X_2 = 1 \) hide \( X_1 \)
otherwise show \( X_1 \)

example

missing at random ✔
missing completely at random ✗
A more general criteria

**missing at random (MAR)** $O_X \perp X_h | X_o$

if there is information about the obs. pattern $O_X$ in $X_h$
then it is also in $X_o$

throw the thumb-tack twice $X = [X_1, X_2]$
if $X_2 = 1$ hide $X_1$
otherwise show $X_1$

no "extra" information in the **obs. pattern** > ignore it

**optimize:**
\[
\ell(D, \theta) = \sum_{x_o \in D} \log \sum_{x_h} p(x_o, x_h)
\]
Likelihood function

for partial observations

- fully observed data:
  - directed: likelihood decomposes
  - undirected: does not decompose, but it is concave
Likelihood function

for partial observations

- **fully observed** data:
  - **directed**: likelihood decomposes
  - **undirected**: does not decompose, but it is concave

- **partially observed**:
  - does not decompose
  - not convex anymore
Likelihood function

for partial observations

- fully observed data:
  - directed: likelihood decomposes
  - undirected: does not decompose, but it is concave

- partially observed:
  - does not decompose
  - not convex anymore

\[
\ell(D, \theta) = \sum_{x_o \in D} \log \sum_{x_h} p(x_o, x_h)
\]
marginal **Likelihood function: example**

for a directed model

fully observed case decomposes:

\[ \ell(D, \theta) = \sum_{x,y,z \in D} \log p(x, y, z) \]

\[ = \sum_x \log p(x) + \sum_{x,y} \log p(y|x) + \sum_{x,z} \log p(z|x) \]
**Likelihood function:** *example*

for a directed model

fully observed case *decomposes*:

\[
\ell(D, \theta) = \sum_{x,y,z \in D} \log p(x, y, z) \\
= \sum_x \log p(x) + \sum_{x,y} \log p(y|x) + \sum_{x,z} \log p(z|x)
\]

\(x\) is always missing (e.g., in a *latent variable model*)

\[
\ell(D, \theta) = \sum_{y,z \in D} \log \sum_x p(x)p(y|x)p(z|x)
\]

*cannot decompose it!*
Parameter learning with missing data

Directed models:

- **option 1:** obtain the gradient of marginal likelihood
- **option 2:** expectation maximization (EM)
  - variational interpretation
Parameter learning with missing data

Directed models:

**option 1:** obtain the gradient of marginal likelihood

**option 2:** expectation maximization (EM)
  - variational interpretation

Undirected models:

obtain the gradient of marginal likelihood
  - EM is not a good option here
Parameter learning with missing data

Directed models:

**option 1:** obtain the gradient of marginal likelihood

**option 2:** expectation maximization (EM)
  - variational interpretation

Undirected models:

obtain the gradient of marginal likelihood
  - EM is not a good option here

All of these options need inference for each step of learning
Gradient of the **marginal** likelihood

\[
\ell(D) = \sum_{(a,d) \in \mathcal{D}} \log \sum_{b,c} p(a) p(b) p(c|a,b) p(d|c)
\]
Gradient of the marginal likelihood

\[
\ell(D) = \sum_{(a,d) \in D} \log \sum_{b,c} p(a)p(b)p(c|a,b)p(d|c)
\]

take the derivative:

\[
\frac{\partial}{\partial p(d'|c')} \ell(D) = \frac{1}{p(d'|c')} \sum_{(a,d) \in D} p(d', c'|a,d)
\]

need inference for this

Example
Gradient of the marginal likelihood

log marginal likelihood:

\[ \ell(D) = \sum_{(a,d) \in D} \log \sum_{b,c} p(a)p(b)p(c|a,b)p(d|c) \]

take the derivative:

\[ \frac{\partial}{\partial p(d'|c')} \ell(D) = \frac{1}{p(d'|c')} \sum_{(a,d) \in D} p(d',c'|a,d) \]

need inference for this

what happens to this expression if every variable is observed?
Gradient of the \textit{marginal} likelihood for a Bayesian Network with CPT

\[
\frac{\partial}{\partial p(x_i|pa_{x_i})} \ell(D) = \frac{1}{p(x_i|pa_{x_i})} \sum_{x_o \in D} p(x_i|pa_{x_i}|x_o)
\]

some specific assignment

run inference for each observation
Gradient of the **marginal likelihood**

for a Bayesian Network with CPT

\[
\frac{\partial}{\partial p(x_i|pa_{x_i})} \mathcal{L}(\mathcal{D}) = \frac{1}{p(x_i|pa_{x_i})} \sum_{x_o \in \mathcal{D}} p(x_i|pa_{x_i}|x_o)
\]

some specific assignment

run inference for each observation

**a technical issue:**

- gradient is always non-negative
  - no constraint of the form \( \sum_x p(x|pa_x) = 1 \)
    - reparametrize (e.g., using softmax)
    - or use Lagrange multipliers
Gradient of the marginal likelihood

for a Bayesian Network with CPT

\[ \frac{\partial}{\partial p(x_i|pa_{x_i})} \ell(D) = \frac{1}{p(x_i|pa_{x_i})} \sum_{x_o \in D} p(x_i|pa_{x_i}|x_o) \]

some specific assignment

run inference for each observation

a technical issue:

- gradient is always non-negative
  - no constraint of the form \( \sum_x p(x|pa_x) = 1 \)
    - reparametrize (e.g., using softmax)
    - or use Lagrange multipliers

for other parametrizations (beyond simple CPTs) use the chain rule:

\[ \frac{\partial}{\partial \theta} \ell(D; \theta) = \sum_{(c',d') \in D} \frac{\partial \ell(D)}{\partial p(d'|c')} \frac{\partial p(d'|c')}{\partial \theta} \]
Expectation Maximization

**E-step:**
for each \( a, d \in D \)
use the current parameters \( \theta \) to get the marginals

more generally: expected sufficient statistics
Expectation Maximization

**E-step:**
for each \( a, d \in \mathcal{D} \)
use the current parameters \( \theta \) to get the marginals

\[
p_{\theta, \mathcal{D}}(B), p_{\theta, \mathcal{D}}(A), p_{\theta, \mathcal{D}}(C), p_{\theta, \mathcal{D}}(A, B, C), p_{\theta, \mathcal{D}}(D, C)
\]

more generally: expected sufficient statistics
Expectation Maximization

E-step:
for each $a, d \in D$
use the current parameters $\theta$ to get the marginals

$$p_{\theta, D}(B), p_{\theta, D}(A), p_{\theta, D}(C), p_{\theta, D}(A, B, C), p_{\theta, D}(D, C)$$

more generally: expected sufficient statistics

$$p_{\theta, D}(C = c', D = d') = \frac{1}{N} \sum_{(a, d) \in D} p_{\theta}(c', d'|a, d)$$

nonzero for $d' = d$

in general we need inference to estimate this sufficient statistics
Expectation Maximization

**E-step:**

for each $a, d \in \mathcal{D}$

use the current parameters $\theta$ to get the marginals

$$p_{\theta, \mathcal{D}}(B), p_{\theta, \mathcal{D}}(A), p_{\theta, \mathcal{D}}(C), p_{\theta, \mathcal{D}}(A, B, C), p_{\theta, \mathcal{D}}(D, C)$$

more generally: expected sufficient statistics

$$p_{\theta, \mathcal{D}}(C = c', D = d') = \frac{1}{N} \sum_{(a, d) \in \mathcal{D}} p_{\theta}(c', d' | a, d)$$

nonzero for $d' = d$

in general we need inference to estimate this sufficient statistics

**M-step:**

use the marginals (similar to completely observed data) to learn $\theta$

expected sufficient statistics
Expectation Maximization

**E-step:**
For each $a, d \in D$
Use the current parameters $\theta$ to get the marginals

$$p_{\theta, D}(B), p_{\theta, D}(A), p_{\theta, D}(C), p_{\theta, D}(A, B, C), p_{\theta, D}(D, C)$$

$$p_{\theta, D}(C = c', D = d') = \frac{1}{N} \sum_{(a, d) \in D} p_{\theta}(c', d' | a, d)$$

Nonzero for $d' = d$

In general we need inference to estimate this sufficient statistics

**M-step:**
Use the marginals (similar to completely observed data) to learn $\theta$

E.g., update $\theta_{C|D}$ using $p_{\theta, D}(C, D)$ and $p_{\theta, D}(C)$

$$\theta_{new}^{D|C} = \frac{p_{\theta, D}(C, D)}{p_{\theta, D}(C)}$$
Expectation Maximization

for a Bayesian Network with CPT

**E-step:**
for each \( x_o \in \mathcal{D} \)
use the current parameters \( \theta \) to get the marginals

\[
\{p_{\theta,\mathcal{D}}(X_i), p_{\theta,\mathcal{D}}(X_i, Pa_{X_i})\}
\]
Expectation Maximization

for a Bayesian Network with CPT

**E-step:**
for each $x_o \in \mathcal{D}$
use the current parameters $\theta$ to get the marginals

$$\{p_{\theta,\mathcal{D}}(X_i), p_{\theta,\mathcal{D}}(X_i, Pa_{X_i})\}$$

**M-step:**
use the marginals (similar to completely observed data) to learn $\theta^{\text{new}}$

$$\theta^{\text{new}}_{X_i | Pa_{X_i}} = \frac{p_{\theta,\mathcal{D}}(X_i, Pa_{X_i})}{p_{\theta,\mathcal{D}}(Pa_{X_i})}$$
Expectation Maximization

E-step:
for each \( x_o \in D \)
use the current parameters \( \theta \) to get the marginals

\[
\{ p_{\theta,D}(X_i), p_{\theta,D}(X_i, Pa_{X_i}) \}
\]

M-step:
use the marginals (similar to completely observed data) to learn \( \theta^{new} \)

\[
\theta^{new}_{X_i|Pa_{X_i}} = \frac{p_{\theta,D}(X_i, Pa_{X_i})}{p_{\theta,D}(Pa_{X_i})}
\]

for undirected models: M-step is the expensive part
- perform E-step within each iteration of M-step: equivalent to gradient descent
Expectation Maximization: example

- 1000 training instances
- 50% of variables are observed (in each instance)

alarm network

fast initial improvement
Expectation Maximization: example

- 1000 training instances
- 50% of variables are observed (in each instance)

![Graph showing change in different parameter values](image)

- Fast initial improvement
- Train log-likelihood
- Test log-likelihood
Expectation Maximization: example

local optima in EM:

alarm network

number of local maxima

# of distinct log-LL

- 25% missing
- 50% Missing
- Hidden variable

Sample size

Train LL/instance

Percentage of runs

a single hidden variable
Expected log-likelihood (directed models)

Original objective:

\[
\ell(D, \theta) = \sum_{x_o \in D} \log \left( \sum_{x_h} p_{\theta}(x_o, x_h) \right) p_{\theta}(x_o)
\]
**Expected log-likelihood**

**(directed models)**

**Original objective:**

\[ \ell(D, \theta) = \sum_{x_o \in D} \log \sum_{x_h} p_\theta(x_o, x_h) \]

**EM iteration:**

- **E-step:** soft-complete the data
- **M-step:** maximize the full likelihood

maximizes the expected log-likelihood
Expected log-likelihood

Original objective:
\[
\ell(D, \theta) = \sum_{x_o \in D} \log \sum_{x_h} p_{\theta}(x_o, x_h) p_{\theta}(x_o)
\]

EM iteration:
\[
\sum_{x_o \in D} \mathbb{E}_{p_{\theta}(x_h | x_o)} \left[ \log p_{\theta}(x_o, x_h) \right]
\]
maximizes the expected log-likelihood

- how are these objectives related?
- any guarantees for EM?
- variational interpretation relates these two

E-step:
soft-complete the data

M-step:
maximize the full likelihood
Variational interpretation of EM

Recall: variational inference

$$\min_q \mathbb{D}_{KL}(q(x) \| p(x)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x)]$$
Variational interpretation of EM

**Recall:** variational inference

\[
\min_q \mathbb{D}_{KL}(q(x)|p(x)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x)]
\]

\[
p(x) = \frac{p(x)}{Z}
\]
Variational interpretation of EM

Recall: variational inference

\[ \min_q \mathbb{D}_{KL}(q(x) \| p(x)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x)] \]

\[ = -\mathbb{H}(q) - \mathbb{E}_q[\log \tilde{p}(x)] + \log Z \]

\[ p(x) = \frac{\tilde{p}(x)}{Z} \]
**Variational interpretation of EM**

**Recall:** variational inference

\[
\min_q \mathbb{D}_{KL}(q(x) \mid p(x)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x)] = -\mathbb{H}(q) - \mathbb{E}_q[\log \hat{p}(x)] + \log Z
\]

\[
p(x) = \frac{\hat{p}(x)}{Z}
\]

for a latent variable model

\[
p(x_h \mid x_o) = \frac{p(x_h, x_o)}{p(x_o)}
\]
Variational interpretation of EM

Recall: variational inference

$$\min_q \mathbb{D}_{KL}(q(x) \| p(x)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x)] = -\mathbb{H}(q) - \mathbb{E}_q[\log \tilde{p}(x)] + \log Z$$

$$p(x) = \frac{\tilde{p}(x)}{Z}$$

for a latent variable model

\[
p(x_h \mid x_o) = \frac{p(x_h, x_o)}{p(x_o)}
\]

\[
\min_q \mathbb{D}_{KL}(q(x_h) \| p(x_h \mid x_o)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x_h \mid x_o)]
\]
Variational interpretation of EM

Recall: variational inference

\[
\min_q \mathbb{D}_{KL}(q(x) \| p(x)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x)] = -\mathbb{H}(q) - \mathbb{E}_q[\log \tilde{p}(x)] + \log Z
\]

\[
p(x) = \frac{\tilde{p}(x)}{Z}
\]

for a latent variable model

\[
\min_q \mathbb{D}_{KL}(q(x_h) \| p(x_h | x_o)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x_h | x_o)] + \mathbb{E}_q[\log p(x_h, x_o)] - \log p(x_o)
\]
Variational interpretation of EM

for a latent variable model

$$\mathbb{D}_{KL}(q(x_h) || p(x_h | x_o)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x_h, x_o)] - \log p(x_o)$$
Variational interpretation of EM

for a latent variable model

\[
\mathbb{D}_{KL}(q(x_h) \mid p(x_h \mid x_o)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x_h, x_o)] - \log p(x_o)
\]

re-arrange

\[
\log p_\theta(x_o) = \mathbb{D}_{KL}(q(x_h) \mid p_\theta(x_h \mid x_o)) + \mathbb{H}(q) + \mathbb{E}_q[\log p_\theta(x_h, x_o)]
\]
**Variational interpretation** of EM

for a *latent variable* model

\[
\mathbb{D}_{KL}(q(x_h) \mid p(x_h \mid x_o)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x_h, x_o)] - \log p(x_o)
\]

re-arrange

\[
\log p_\theta(x_o) = \mathbb{D}_{KL}(q(x_h) \mid p_\theta(x_h \mid x_o)) + \mathbb{H}(q) + \mathbb{E}_q[\log p_\theta(x_h, x_o)]
\]

original objective
Variational interpretation of EM

for a latent variable model

$$\mathcal{D}_{KL}(q(x_h) \mid p(x_h \mid x_o)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x_h, x_o)] - \log p(x_o)$$

re-arrange

$$\log p_\theta(x_o) = \mathcal{D}_{KL}(q(x_h) \mid p_\theta(x_h \mid x_o)) + \mathbb{H}(q) + \mathbb{E}_q[\log p_\theta(x_h, x_o)]$$

original objective

expected log-likelihood wrt q
Variational interpretation of EM

for a latent variable model

$$\mathbb{D}_{KL}(q(x_h) | p(x_h | x_o)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x_h, x_o)] - \log p(x_o)$$

re-arrange

$$\log p_\theta(x_o) = \mathbb{D}_{KL}(q(x_h) | p_\theta(x_h | x_o)) + \mathbb{H}(q) + \mathbb{E}_q[\log p_\theta(x_h, x_o)]$$

original objective

expected log-likelihood wrt q
Variational interpretation of EM

for a latent variable model

$$\mathbb{D}_{KL}(q(x_h) | p(x_h | x_o)) = -\mathbb{H}(q) - \mathbb{E}_q[\log p(x_h, x_o)] - \log p(x_o)$$

re-arrange

$$\log p_{\theta}(x_o) = \mathbb{D}_{KL}(q(x_h) | p_{\theta}(x_h | x_o)) + \mathbb{H}(q) + \mathbb{E}_q[\log p_{\theta}(x_h, x_o)]$$

original objective

expected log-likelihood wrt q

ignored by EM
**Variational interpretation of EM**

for a **latent variable model**

\[ \mathbb{D}_{KL}(q(x_h) \| p(x_h \| x_o)) = -H(q) - \mathbb{E}_q[\log p(x_h, x_o)] - \log p(x_o) \]

re-arrange

\[ \log p_\theta(x_o) = \mathbb{D}_{KL}(q(x_h) \| p_\theta(x_h \| x_o)) + H(q) + \mathbb{E}_q[\log p_\theta(x_h, x_o)] \]

original objective

expected log-likelihood wrt q

ignored by EM

Coordinate ascent:

- **E-step**: optimize \( q \) for a fixed \( \theta \) (variational inference)
- **M-step**: optimize \( \theta \) for a fixed \( q \)
EM as coordinate ascent

Coordinate ascent:

- **E-step**: optimize q for a fixed $\theta$
- **M-step**: optimize $\theta$ for a fixed q

guaranteed improvement of $\log p_\theta(x_o)$
EM as coordinate ascent

Coordinate ascent:
- **E-step**: optimize $q$ for a fixed $\theta$
- **M-step**: optimize $\theta$ for a fixed $q$

guaranteed improvement of $\log p_\theta(x_o)$
EM as coordinate ascent

Coordinate ascent:

- **E-step**: optimize $q$ for a fixed $\theta$
- **M-step**: optimize $\theta$ for a fixed $q$

guaranteed improvement of $\log p_\theta(x_o)$ converges to a local optimum
Amortized inference in latent variable models

$$\log p_\theta(x_o) = \mathbb{D}_{KL}(q(x_h) | p_\theta(x_h | x_o)) + \mathbb{H}(q) + \mathbb{E}_q[\log p_\theta(x_h, x_o)]$$
Amortized inference in latent variable models

\[
\log p_\theta(x_o) = D_{KL}(q(x_h)|p_\theta(x_h|x_o)) + \mathbb{H}(q) + \mathbb{E}_q[\log p_\theta(x_h, x_o)]
\]

evidence lower bound (ELBO) is a lower-bound on the likelihood
Amortized inference in latent variable models

\[ \log p_\theta(x_o) = \mathbb{D}_{KL}(q(x_h) \mid p_\theta(x_h \mid x_o)) + \mathbb{H}(q) + \mathbb{E}_q[\log p_\theta(x_h, x_o)] \]

Evidence lower bound (ELBO) is a lower-bound on the likelihood

\[ q_\psi(x_h \mid x_o) \text{ instead of } q(x_h) \]

Amortization: make \( q \) a function of observations
Amortized inference in latent variable models

\[
\log p_\theta(x_o) = \mathbb{D}_{KL}(q(x_h)\|p_\theta(x_h|x_o)) + \mathbb{H}(q) + \mathbb{E}_q[\log p_\theta(x_h, x_o)]
\]

evidence lower bound (ELBO) is a lower-bound on the likelihood

\[
q_\psi(x_h | x_o) \text{ instead of } q(x_h)
\]

amortization: make q a \textit{function} of observations

\[
p_\theta(x_h, x_o) = p_\theta(x_h)p_\theta(x_o|x_h)
\]
Amortized inference in latent variable models

\[
\log p_\theta(x_o) = D_{KL}(q(x_h)|p_\theta(x_h|x_o)) + H(q) + E_q[\log p_\theta(x_h, x_o)]
\]

Evidence lower bound (ELBO) is a lower-bound on the likelihood

\[
q_\psi(x_h | x_o) \text{ instead of } q(x_h)
\]

Amortization: make \( q \) a function of observations

\[
p_\theta(x_h, x_o) = p_\theta(x_h)p_\theta(x_o|x_h)
\]

\[
-D_{KL}(q_\psi(x_h | x_o)|p_\theta(x_h)) + E_q[\log p_\theta(x_o|x_h)]
\]
Amortized inference in latent variable models

\[
\log p_\theta(x_o) = \mathbb{D}_{KL}(q(x_h|x_\theta)\|p_\theta(x_h|x_o)) + H(q) + \mathbb{E}_q[\log p_\theta(x_h, x_o)]
\]

Evidence lower bound (ELBO) is a lower-bound on the likelihood

\[
q_\psi(x_h | x_o) \text{ instead of } q(x_h)
\]

Amortization: make \( q \) a function of observations

\[
p_\theta(x_h, x_o) = p_\theta(x_h)p_\theta(x_o|x_h)
\]

\[
-\mathbb{D}_{KL}(q_\psi(x_h | x_o)\|p_\theta(x_h)) + \mathbb{E}_{q_\psi}[\log p_\theta(x_o|x_h)]
\]

Maximize ELBO by jointly optimizing \( \psi, \theta \)
**Amortized inference** in latent variable models

\[
\log p_\theta(x_o) = D_{KL}(q(x_h|x_o)p_\theta(x_h|x_o)) + H(q) + E_q[\log p_\theta(x_h, x_o)]
\]

evidence lower bound (ELBO) is a lower-bound on the likelihood

\[
q_\psi(x_h | x_o) \text{ instead of } q(x_h) \quad \text{amortization: make } q \text{ a function of observations}
\]

\[
p_\theta(x_h, x_o) = p_\theta(x_h)p_\theta(x_o|x_h)
\]

\[
-D_{KL}(q_\psi(x_h | x_o)p_\theta(x_h)) + E_{q_\psi}[\log p_\theta(x_o|x_h)]
\]

**Variational Auto-Encoder (VAE)**

maximize ELBO by jointly optimizing \( \psi, \theta \)

use neural networks to represent cond. distributions

use back propagation for optimization
Undirected models with latent variables

Linear exponential family:

\[ p(x; \theta) = \frac{1}{Z(\theta)} \exp(\langle \theta, \phi(x) \rangle) \]

Gradient in the fully observed setting:

\[ \nabla_\theta \ell(\theta, D) = |D| (\mathbb{E}_D [\phi(x)] - \mathbb{E}_{p_0} [\phi(x)]) \]

- Expectation wrt the data
- Expectation wrt the model
**Undirected models with latent variables**

Linear exponential family

\[ p(x; \theta) = \frac{1}{Z(\theta)} \exp(\langle \theta, \phi(x) \rangle) \]

Gradient in the fully observed setting

\[ \nabla_\theta \ell(\theta, D) = |D| (\mathbb{E}_D [\phi(x)] - \mathbb{E}_{p_0} [\phi(x)]) \]

Partial observation: \( x = (x_o, x_n) \)

Not observed
**Undirected models** with latent variables

- **Linear exponential family**
  \[ p(x; \theta) = \frac{1}{Z(\theta)} \exp(\langle \theta, \phi(x) \rangle) \]

- **Gradient** in the fully observed setting
  \[ \nabla_{\theta} \ell(\theta, D) = |D| (\mathbb{E}_D[\phi(x)] - \mathbb{E}_{p_{\theta}}[\phi(x)]) \]

- **Partial observation**: \( x = (x_o, x_h) \)
  - \( x_h \) not observed

- **Marginal likelihood**:
  \[ p(x_o; \theta) = \sum_{x_h} \frac{1}{Z(\theta)} \exp(\langle \theta, \phi(x) \rangle) \]

- **Expectation**: wrt the data, wrt the model
**Undirected models** with latent variables

linear exponential family

\[ p(x; \theta) = \frac{1}{Z(\theta)} \exp(\langle \theta, \phi(x) \rangle) \]

gradient in the fully observed setting

\[ \nabla_\theta \ell(\theta, D) = |D| (\mathbb{E}_D[\phi(x)] - \mathbb{E}_{p_\theta}[\phi(x)]) \]

partial observation: \( x = (x_o, x_h) \)

not observed

marginal likelihood: \( p(x_o; \theta) = \sum_{x_h} \frac{1}{Z(\theta)} \exp(\langle \theta, \phi(x) \rangle) \)

gradient in the partially obs. case

\[ \nabla_\theta \ell(\theta, D) = |D| (\mathbb{E}_{D,\theta}[\phi(x)] - \mathbb{E}_{p_\theta}[\phi(x)]) \]

wrt both data and model: we need to do inference to calculate expected sufficient statistics (similar to E-step in EM)
Example: Restricted Boltzmann Machine (RBM)

**binary RBM:** $p(h,v) = \frac{1}{Z(\theta)} \exp(\sum_{i,j} \theta_{i,j} v_i h_j)$

data: $\mathcal{D} = \{v^{(m)}\}_m$ for $v_i, h_j \in \{0, 1\}$
Example: Restricted Boltzmann Machine (RBM)

binary RBM: \[ p(h, v) = \frac{1}{Z(\theta)} \exp(\sum_{i,j} \theta_{i,j} v_i h_j) \]

data: \( D = \{v^{(m)}\}_m \) for \( v_i, h_j \in \{0, 1\} \)

sufficient statistics: \( \phi(v_i, h_j) = v_i h_j \)
Example: Restricted Boltzmann Machine (RBM)

binary RBM:  \[ p(h, v) = \frac{1}{Z(\theta)} \exp(\sum_{i,j} \theta_{i,j} v_i h_j) \]

data:  \[ \mathcal{D} = \{v^{(m)}\}_m \]

sufficient statistics:  \[ \phi(v_i, h_j) = v_i h_j \]

we want to optimize:  \[ \ell(\mathcal{D}; \theta) = \sum_{v \in \mathcal{D}} \log \sum_h \frac{1}{Z(\theta)} \exp(\sum_{i,j} \theta_{i,j} v_i h_j) \]
**Example:** Restricted Boltzmann Machine (RBM)

binary RBM: \[ p(h, v) = \frac{1}{Z(\theta)} \exp(\sum_{i,j} \theta_{i,j} v_i h_j) \]

for \( v_i, h_j \in \{0, 1\} \)

data: \( D = \{ v^{(m)} \}_m \)

sufficient statistics: \( \phi(v_i, h_j) = v_i h_j \)

we want to optimize: \( \ell(D; \theta) = \sum_{v \in D} \log \sum_h \frac{1}{Z(\theta)} \exp(\sum_{i,j} \theta_{i,j} v_i h_j) \)

gradient: \( \frac{\partial}{\partial \theta_{i,j}} \ell(D; \theta) \propto \mathbb{E}_{D,\theta}[v_i h_j] - \mathbb{E}_{p_\theta}[v_i h_j] \)

\[ = \left( \frac{1}{M} \sum_{v_i' \in D} v_i' \mathbb{E}_{p_\theta} [h_j | v_i'] - \mathbb{E}_{p_\theta} [v_i h_j] \right) \]
Example: Restricted Boltzmann Machine (RBM)

Binary RBM: \[ p(h, v) = \frac{1}{Z(\theta)} \exp(\sum_{i,j} \theta_{i,j} v_i h_j) \]

data: \[ D = \{v^{(m)}\}_m \]

sufficient statistics: \[ \phi(v_i, h_j) = v_i h_j \]

we want to optimize: \[ \ell(D; \theta) = \sum_{v \in D} \log \sum_h \frac{1}{Z(\theta)} \exp(\sum_{i,j} \theta_{i,j} v_i h_j) \]

gradient: \[ \frac{\partial}{\partial \theta_{i,j}} \ell(D; \theta) \propto \mathbb{E}_{D,\theta} [v_i h_j] - \mathbb{E}_{p_{\theta}} [v_i h_j] \]
\[= \left( \frac{1}{M} \sum_{v'_i \in D} v'_i \mathbb{E}_{p_{\theta}} [h_j | v'_i] \right) - \mathbb{E}_{p_{\theta}} [v_i h_j] \]

sampling-based inference: sample \( h \mid v \)

use Gibbs sampling: sample both \( h,v \) using current parameters
learning with partial observations:
- missing data
  - optimize the likelihood when missing at random
- latent variables
  - can produce expressive probabilistic models

problem is not convex

how to learn the model?
- directly estimate the gradient \((\text{directed and undirected})\)
- use EM \((\text{directed models})\)
  - variational interpretation + relation to ELBO
Example: Gaussian mixture model

- Model parameters: \( \theta = [\pi, \{\mu_k, \Sigma_k\}] \)
- Probability density function:
  - \( p(x; \pi) = \prod_k \pi_k \mathbb{1}(x=k) \)
  - \( p(y|x; \{\mu_k, \Sigma_k\}) = \frac{1}{\sqrt{|2\pi \Sigma|}} \exp\left(-\frac{1}{2}(y - \mu_x)^T \Sigma_x^{-1} (y - \mu_x)\right) \)
Example: Gaussian mixture model

\[
p(x; \pi) = \prod_k \pi_k \mathbb{I}(x=k)
\]

\[
p(y|x; \{\mu_k, \Sigma_k\}) = \frac{1}{\sqrt{|2\pi\Sigma_x|}} \exp\left(-\frac{1}{2}(y - \mu_x)^T \Sigma_x^{-1} (y - \mu_x)\right)
\]

**E-step:** calculate \( p(x|y) \) for each \( y \in \mathcal{D} \)

\[
p(x|y) \propto p(x; \pi)p(y|x; \mu, \Sigma) = \pi_k \mathcal{N}(y; \mu_k, \Sigma_k)
\]

- now we have "probabilistically completed" instances
- update the parameters (easy in a Bayes-net)
Example: Gaussian mixture model

$$p(x; \pi) = \prod_k \pi_k \mathbb{I}(x=k)$$

$$p(y|x; \{\mu_k, \Sigma_k\}) = \frac{1}{\sqrt{2\pi \Sigma_x}} \exp\left(-\frac{1}{2} (y - \mu_x)^T \Sigma_x^{-1} (y - \mu_x) \right)$$

**M-step:** estimate $\pi, \mu_k, \Sigma_k \forall k$

$$\pi_k^{\text{new}} = \frac{1}{N} \sum_{y \in D} \frac{p(x=k|y)}{\sum_{k'} p(x=k'|y)}$$

$$\mu_k = \frac{\sum_{y \in D} p(x=k|y) y}{\sum_{y \in D} p(x=k|y)} \quad \text{mean of a weighted set of instances}$$

$$\Sigma_k = \frac{\sum_{y \in D} p(x=k|y)(y-\mu_k)(y-\mu_k)^T}{\sum_{y \in D} p(x=k|y)} \quad \text{covariance of a weighted set of instances}$$
**Example: Gaussian mixture model**

\[
p(x; \pi) = \prod_k \pi_k \mathbb{I}(x=k)
\]

**model parameters**

\[
p(y|x; \{\mu_k, \Sigma_k\}) = \frac{1}{\sqrt{2\pi|\Sigma_x|}} \exp\left(-\frac{1}{2}(y - \mu_x)^T \Sigma_x^{-1}(y - \mu_x)\right)
\]

**M-step:** estimate \(\pi, \mu_k, \Sigma_k \forall k\)

\[
\pi_k^{\text{new}} = \frac{1}{N} \sum_{y \in D} \frac{p(x=k|y)}{\sum_{k'} p(x=k'|y)}
\]

\[
\mu_k = \frac{\sum_{y \in D} p(x=k|y)y}{\sum_{y \in D} p(x=k|y)} \quad \text{mean of a weighted set of instances}
\]

\[
\Sigma_k = \frac{\sum_{y \in D} p(x=k|y)(y-\mu_k)(y-\mu_k)^T}{\sum_{y \in D} p(x=k|y)} \quad \text{covariance of a weighted set of instances}
\]