Learning: Incomplete data

Probabilistic Graphical Models
Sharif University of Technology
Spring 2017

Soleymani
Parameter learning for incomplete data

- We now assume that the structure of the model is known and consider learning parameters when the data is incomplete.

- **Goal:** estimate parameters from a dataset \( \mathcal{D} = \{x^{(1)}, \ldots, x^{(N)}\} \) of \( N \) independent, identically distributed (i.i.d.) training samples.
  - Each training sample \( x^{(n)} = [x_O^{(n)}, x_H^{(n)}] \) is a vector that some elements \( x_i^{(n)} \) may be unknown (missing values or hidden variables).

Incomplete data

- **Hidden or latent variables**
  - The variables that are unobserved for all data samples
- **Missing values**
  - For each data sample some of the variables may be missing
Missing At Random (MAR)

- We ignore the missing data mechanism and focus only on the likelihood.
  - We assume that data have been missing at random
    - the mechanism causing values to be missing does not depend on the unobserved values.
Latent variables

Examples of hidden or latent variables:

- In unsupervised learning,
  - Clustering
  - Dimensionality reduction
    - simplified and abstractive view of the data generation process
  - Multi-modal density estimation
- For real variables that we cannot know their values
Incorporate likelihood

- **Complete likelihood**
  - Maximizing likelihood $P(\mathcal{D}|\theta)$ for labeled data is straightforward
  - In directed models (with i.i.d. settings) the likelihood decomposes:
    $$P(\mathcal{D}|\theta) = \prod_{n=1}^{N} P(x^{(n)}|y^{(n)}, \theta_{X|Y}) \prod_{n=1}^{N} P(y^{(n)}|\theta_Y)$$

- **Incomplete likelihood**
  - With $Z$ unobserved, our objective becomes $P(\mathcal{D}|\theta) = \sum_{\mathcal{H}} P(\mathcal{D}, \mathcal{H}|\theta)$
  - All the parameters become coupled together via marginalization:
    $$P(\mathcal{D}|\theta) = \sum_{\mathcal{H}} P(\mathcal{D}, \mathcal{H}|\theta) = \sum_{\mathcal{H}} \prod_{n=1}^{N} P(x^{(n)}, z^{(n)}|\theta)$$
    $$= \prod_{n=1}^{N} \sum_{z^{(n)}} P(x^{(n)}|z^{(n)}, \theta_{X|Z}) P(z^{(n)}|\theta_Z)$$
Complexity of the incomplete likelihood function

\[ P(\mathcal{D}|\theta) = \sum_{\mathcal{H}} P(\mathcal{D}, \mathcal{H}|\theta) \]

- Incomplete likelihood is the sum of likelihood functions, one for each possible joint assignment of the missing values.
- If each one of the complete likelihood is a unimodal function. Their sum, however, can be multimodal.
  - In the worst case, the likelihood of each of the possible assignments to missing values contributes to a different peak in the likelihood function.
    - The number of possible assignments is exponential in the total number of missing values.
Optimization methods to maximize incomplete likelihood

- Gradient-based methods
  - need to ensure that parameters in each iteration define legal CPDs
  - For reasonable convergence, need to combine with advanced methods
    - Line-search, conjugate gradient

- Expectation Maximization (EM)
  - Special-purpose algorithm designed for optimizing likelihood functions
  - no learning rate (i.e., step-size) parameter is required
  - automatically enforces parameter constraints
EM algorithm

- General algorithm for finding ML estimation when the data is incomplete (missing or unobserved data).

- Assumptions: \( D \) (observed or known variables), \( H \) (unobserved or latent variables), data come from a specific model with unknown parameters \( \theta \)
  - If \( H \) is relevant to \( D \) (in any way), we can hope to extract information about it from \( D \) assuming a specific parametric model on the data.

- An iterative algorithm in which each iteration is guaranteed to improve the log-likelihood function
EM intuition

- When learning with incomplete data, we are trying to solve two problems at once:
  - hypothesizing values for the unobserved variables in each data sample
  - learning the parameters

- Each of these tasks is fairly easy when we have the solution to the other:
  - Given complete data, we have the statistics, and we can estimate parameters using the MLE formulas.
  - Conversely, computing probability of unobserved variables given the parameters is an inference problem
**EM algorithm**

\[ \mathcal{D}: \text{observed variables} \]
\[ \mathcal{H}: \text{Unobserved variables} \]
\[ \theta: \text{parameters} \]

**Expectation step (E-step):** given the current parameters, find soft completion of the data, using probabilistic inference.

**Maximization step (M-step):** We then treat the soft completed data as if it were observed and learn a new set of parameters.

Choose an initial parameters \( \theta^1 \)
\[ t \leftarrow 1 \]

Iterate until convergence:

**E Step:** Calculate \( P(\mathcal{H}|\mathcal{D}, \theta^t) \)

**M Step:** \( \theta^{t+1} = \arg \max \theta E_{P(\mathcal{H}|\mathcal{D}, \theta^t)}[\log P(\mathcal{D}, \mathcal{H}|\theta)] \)
\[ t \leftarrow t + 1 \]

Guaranteed to improve \( \ln P(\mathcal{D}|\theta) \) at each iteration
EM algorithm objective function

- $\ln P(\mathcal{D}, \mathcal{H} | \theta)$ is a random quantity and it cannot be maximized directly.

- EM intends to maximize $E_{P(\mathcal{H} | \mathcal{D}, \theta)}[\ln P(\mathcal{D}, \mathcal{H} | \theta)]$, i.e., expected log complete likelihood:

$$\hat{\theta} = \arg\max_{\theta} E_{P(\mathcal{H} | \mathcal{D}, \theta)}[\ln P(\mathcal{D}, \mathcal{H} | \theta)]$$

- Expected log complete likelihood
  - Linear in $\ln P(\mathcal{D}, \mathcal{H} | \theta)$ and inherit its factorizability
Mixture models: definition

- Mixture models: Linear supper-position of mixtures or components
  - $Z$: latent or hidden variable specifies the mixture component

$$P(x|\theta) = \sum_{k=1}^{K} P(Z_k = 1) P(x|Z_k = 1, \theta_k)$$

- $\sum_{k=1}^{K} P(Z_k = 1) = 1$
- $P(Z_k = 1)$: the prior probability of $k$-th mixture
- $\theta_k$: the parameters of $k$-th mixture
- $P(x|Z_k = 1, \theta_k)$: the probability of $x$ in the $k$-th mixture

Multi-modal distribution
Gaussian Mixture Models (GMMs)

- **Gaussian Mixture Models:**
  - \( P(x|Z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k) \)
  - \( Z \sim \text{Mult}(\pi) \)
    - \( P(Z_k = 1) = \pi_k \)
    - \( 0 \leq \pi_k \leq 1 \)
    - \( \sum_{k=1}^{K} \pi_k = 1 \)
  - \( P(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \)

- **Fitting Gaussian mixture model**
  - Input: data points \( \{x^{(i)}\}_{i=1}^{N} \)
  - Goal: find the parameters of GMM \( (\pi_k, \mu_k, \Sigma_k, k = 1, ..., K) \)
GMM: 1-D example

\[ \begin{align*} 
\pi_1 &= 0.6 \\
\pi_2 &= 0.3 \\
\pi_3 &= 0.1 \\
\mu_1 &= -2 \\
\sigma_1 &= 2 \\
\mu_2 &= 4 \\
\sigma_2 &= 1 \\
\mu_3 &= 8 \\
\sigma_3 &= 0.2
\end{align*} \]
EM for GMM

- Initialize $\theta^{old}: \mu_k, \Sigma_k, \pi_k \quad k = 1, \ldots, K$
- **E step:** $n = 1, \ldots, N, k = 1, \ldots, K$
  \[
  \gamma_k^n = P \left( Z_k^n = 1 | x^{(n)}, \theta^{old} \right) = \frac{\pi_k^{old} \mathcal{N}(x^{(n)} | \mu_k^{old}, \Sigma_k^{old})}{\sum_{j=1}^{K} \pi_j^{old} \mathcal{N}(x^{(n)} | \mu_j^{old}, \Sigma_j^{old})}
  \]
- **M Step:** $k = 1, \ldots, K$
  \[
  \mu_k^{new} = \frac{\sum_{n=1}^{N} \gamma_k^n x^{(n)}}{\sum_{n=1}^{N} \gamma_k^n}
  \]
  \[
  \Sigma_k^{new} = \frac{1}{\sum_{n=1}^{N} \gamma_k^n} \sum_{n=1}^{N} \gamma_k^n (x^{(n)} - \mu_k^{new})(x^{(n)} - \mu_k^{new})^T
  \]
  \[
  \pi_k^{new} = \frac{\sum_{n=1}^{N} \gamma_k^n}{N}
  \]
- Repeat E and M steps until convergence

$\theta = [\pi, \mu, \Sigma]$ 

$z^{(n)} \in \{1, 2, \ldots, K\}$ shows the mixture from which $x^{(n)}$ is generated
EM & GMM: example
EM & GMM: Example

(a) $L = 1$

(b) $L = 2$

(c) $L = 5$

(d) $L = 20$

[Bishop]
EM theoretical analysis

- What is the underlying theory for the use of the expected complete log likelihood in the M-step?

\[ E_{P(\mathcal{H}|\mathcal{D},\theta)}[\log P(\mathcal{D},\mathcal{H}|\theta)] \]

- Now, we show that maximizing this function also maximizes the likelihood
EM theoretical foundation:
Objective function

\[ \ell(\theta; D) = \log P(D|\theta) = \log \sum_{H} P(D, H|\theta) \]

\[ = \log \sum_{H} Q(H) \frac{P(D, H|\theta)}{Q(H)} \geq \sum_{H} Q(H) \log \frac{P(D, H|\theta)}{Q(H)} \]

\[ F_D[\theta, Q] \]

\( F_D[\theta, Q] \) is a lower bound on \( \ell(\theta; D) \)

EM optimizes \( F_D[\theta, Q] \)
EM theoretical foundation: Algorithm in general form

- EM is a coordinate ascent algorithm on $F_D[\theta, Q]$. In the $t$-th iteration,
  - E-step: maximize $F_D[\theta, Q]$ w.r.t. $Q$
    $$Q^t = \arg\max_Q F_D[\theta^t, Q]$$
  - M-step:
    $$\theta^{t+1} = \arg\max_{\theta} F_D[\theta, Q^t]$$

We will show that each iteration improves the log-likelihood
EM theoretical foundation: E-step

\[ Q^t = P(\mathcal{H}|\mathcal{D}, \theta^t) \implies Q^t = \arg\max_Q F_\mathcal{D}[\theta^t, Q] \]

Proof:

\[
F_\mathcal{D}[\theta^t, P(\mathcal{H}|\mathcal{D}, \theta^t)] = \sum_{\mathcal{H}} P(\mathcal{H}|\mathcal{D}, \theta^t) \log \frac{P(\mathcal{D}, \mathcal{H}|\theta^t)}{P(\mathcal{H}|\mathcal{D}, \theta^t)} \\
= \sum_{\mathcal{H}} P(\mathcal{H}|\mathcal{D}, \theta^t) \log P(\mathcal{D}|\theta^t) = \log P(\mathcal{D}|\theta^t) = \ell(\theta^t; \mathcal{D})
\]

- \( F_\mathcal{D}[\theta, Q] \) is a lower bound on \( \ell(\theta; \mathcal{D}) \). Thus, \( F_\mathcal{D}[\theta^t, Q] \) has been maximized by setting \( Q \) to \( P(\mathcal{H}|\mathcal{D}, \theta^t) \):

\[
F_\mathcal{D}[\theta^t, P(\mathcal{H}|\mathcal{D}, \theta^t)] = \ell(\theta^t; \mathcal{D})
\]

\[
\implies P(\mathcal{H}|\mathcal{D}, \theta^t) = \arg\max_Q F_\mathcal{D}[\theta^t, Q]
\]
EM theoretical foundation:
M-step

M-step can be equivalently viewed as maximizing the expected complete log-likelihood:

\[ \theta^{t+1} = \arg\max_{\theta} F_D[\theta, Q^t] = \arg\max_{\theta} E_{Q^t} [\log P(\mathcal{D}, \mathcal{H} | \theta)] \]

Proof:

\[ F_D[\theta, Q^t] = \sum_{\mathcal{H}} Q^t(\mathcal{H}) \log \frac{P(\mathcal{D}, \mathcal{H} | \theta)}{Q^t(\mathcal{H})} \]

\[ = \sum_{\mathcal{H}} Q^t(\mathcal{H}) \log P(\mathcal{D}, \mathcal{H} | \theta) - \sum_{\mathcal{H}} Q^t(\mathcal{H}) \log Q^t(\mathcal{H}) \]

\[ \Rightarrow F_D[\theta, Q^t] = E_{Q^t} [\log P(\mathcal{D}, \mathcal{H} | \theta)] + H_{Q^t(\mathcal{H})} \]

Independent of \( \theta \)
EM algorithm

$D$: observed variables
$H$: Unobserved variables
$\theta$: parameters

Expectation Maximization (EM) seeks to estimate:

$$\hat{\theta} = \arg \max_{\theta} E_{P(H|D,\theta)} [\log P(D, H|\theta)]$$

Choose an initial parameters $\theta^1$
$t \leftarrow 1$

Iterate until convergence:

**E Step:** Calculate $P(H|D, \theta^t)$

**M Step:** $\theta^{t+1} = \arg \max_{\theta} E_{P(H|D,\theta^t)} [\log P(D, H|\theta)]$

$t \leftarrow t + 1$
EM algorithm: illustration
EM theoretical foundation

\( \ell(\theta; \mathcal{D}) = F_{\mathcal{D}}[\theta, Q] + D(Q(\mathcal{H}) || P(\mathcal{H}|\mathcal{D}, \theta)) \)

Proof:

\[
\ell(\theta; \mathcal{D}) - F_{\mathcal{D}}[\theta, Q] = \log P(\mathcal{D}|\theta) - \sum_{\mathcal{H}} Q(\mathcal{H}) \log \frac{P(\mathcal{D}, \mathcal{H}|\theta)}{Q(\mathcal{H})} \\
= \sum_{\mathcal{H}} Q(\mathcal{H}) \left[ \log P(\mathcal{D}|\theta) - \log \frac{P(\mathcal{D}, \mathcal{H}|\theta)}{Q(\mathcal{H})} \right] \\
= \sum_{\mathcal{H}} Q(\mathcal{H}) \left[ \log Q(\mathcal{H}) - \log \frac{P(\mathcal{D}, \mathcal{H}|\theta)}{P(\mathcal{D}|\theta)} \right] \\
= \sum_{\mathcal{H}} Q(\mathcal{H}) \log \frac{Q(\mathcal{H})}{P(\mathcal{H}|\mathcal{D}, \theta)} = D(Q(\mathcal{H}) || P(\mathcal{H}|\mathcal{D}, \theta))
\]
KL divergence

- Kullback-Leibler divergence between $P$ and $Q$:
  \[ D(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} \, dx \]

- A result from information theory: For any $P$ and $Q$
  \[ D(P||Q) \geq 0 \]

- $D(P||Q) = 0$ if and only if $P \equiv Q$
- $D$ is asymmetric
EM theoretical foundation
E-step illustration

\[ \ell(\theta; D) = F_D[\theta, Q] + D(Q(\mathcal{H}) \parallel P(\mathcal{H}|D, \theta)) \]

- Thus, since \( D(Q||P) \geq 0 \) and \( D(Q||P) = 0 \) iff \( Q(\mathcal{H}) = P(\mathcal{H}|D, \theta) \) then:
  - indeed we select \( Q \) in the E-step such that the gap \( \ell(\theta^{old}; D) - F_D[\theta^{old}, Q] \) removes.
EM iteration increases $\ell(\theta; D)$

\[
\ell(\theta^{t+1}; D) \geq E_{Q_t}[\log P(D, H | \theta^{t+1})] + H_{Q_t}(H)
\]

\[
\ell(\theta^t; D) = E_{Q_t}[\log P(D, H | \theta^t)] + H_{Q_t}(H)
\]

\[
\ell(\theta^{t+1}; D) - \ell(\theta^t; D) \geq E_{Q_t}[\log P(D, H | \theta^{t+1})] - E_{Q_t}[\log P(D, H | \theta^t)]
\]

Moreover, we have:

\[
\theta^{t+1} = \arg\max_{\theta} E_{Q_t}[\log P(D, H | \theta)]
\]

\[
\Rightarrow E_{Q_t}[\log P(D, H | \theta^{t+1})] \geq E_{Q_t}[\log P(D, H | \theta^t)]
\]

\[
\Rightarrow \ell(\theta^{t+1}; D) - \ell(\theta^t; D) \geq 0
\]
EM for exponential family

- We generally have:

\[
E_{P(\mathcal{H}|\mathcal{D},\theta^t)}[\ln P(\mathcal{D}, \mathcal{H}|\theta)] = \sum_{n=1}^{N} \sum_{x_h^{(n)} \in \text{Val}(X_h)} P \left( x_h^{(n)} | x_o^{(n)}, \theta \right) \ln P \left( x_o^{(n)}, x_h^{(n)} | \theta \right)
\]

- For exponential family:

\[
P \left( x_o^{(n)}, x_h^{(n)} | \theta \right) = h \left( x_o^{(n)}, x_h^{(n)} \right) \exp \left\{ \eta^T T \left( x_o^{(n)}, x_h^{(n)} \right) - A(\eta) \right\}
\]

\[x^{(n)} = [x_o^{(n)}, x_h^{(n)}]\]
EM for exponential family

\[ E_{P(\mathcal{H}|D,\theta^t)}[\ln P(D, \mathcal{H}|\theta)] = \eta^T \sum_{n=1}^{N} \sum_{x_h^{(n)} \in Val(X_h)} P(x_h^{(n)}|x_o^{(n)}, \theta^t) T(x_o^{(n)}, x_h^{(n)}) - NA(\eta) + C \]

Expected sufficient statistics

**E-step:** Find expected sufficient statistics

\[ \sum_{n=1}^{N} E_{P(x_h^{(n)}|x_o^{(n)}, \theta^t)} [T(x_o^{(n)}, x_h^{(n)})] \]

**M-step:**

ML estimator uses the above expected sufficient statistics instead of sufficient statistics \( T(D) = \sum_{n=1}^{N} T(x^{(n)}) \) that is used for complete data
EM algorithm: expected sufficient statistics

- **E-step**: computing the expected value of the sufficient statistics

- **M-step**: This is isomorphic to MLE except that we use “expected sufficient statistics” instead of “sufficient statistics”

- **Example**: tabular CPDs in Bayesian networks for the i-th node:

\[
Ess_{\theta_t}[x,u] = \sum_{n=1}^N P(X_i = x, P_a x_i = u|x_0^{(n)}, \theta^t)
\]

\[
\theta_{i,x,u} = \frac{Ess_{i,\theta_t}[x,u]}{\sum_x Ess_{i,\theta_t}[x,u]}
\]
EM for Bayesian networks

- Initialize parameters $\theta^1$
- while not converged
  - Set $E$ to zero for all nodes
- %E-step:
  - For each data sample $n$
    - Run inference $P(x^{(n)}_h | x^{(n)}_o, \theta^t)$
    - For each node $i$
      - For each $x, u \in Val(X_i, P_{aX_i})$
        - $M_i[x, u] = M_i[x, u] + P(X_i = x, P_{aX_i} = u | x^{(n)}_o, \theta^t)$
  - %M-step:
    - For each node $i$
      - For each $x, u \in Val(X_i, P_{aX_i})$
        - $\theta_{i,x,u}^{t+1} = \frac{M_i[x,u]}{M_i[u]}$
        - $t \leftarrow t + 1$
Conditional mixture models: Mixture of linear regression

\[
P(y|x, \theta) = \sum_{k=1}^{K} P(z_k = 1) P(y|z_k = 1, x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y|w_k^T x, \sigma^2)
\]

\[
\mathcal{D}^X = \{x^{(1)}, \ldots, x^{(N)}\}
\]
\[
\mathcal{D}^Y = \{y^{(1)}, \ldots, y^{(N)}\}
\]

\[
\ln P(\mathcal{D}^Y, \mathcal{H} | \mathcal{D}^X, \theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_k^{(n)} \ln(\pi_k \mathcal{N}(y^{(n)}|w_k^T x^{(n)}, \sigma^2))
\]

\[
E_{P(\mathcal{H} | \mathcal{D}, \theta^t)}[\ln P(\mathcal{D}^Y, \mathcal{H} | \mathcal{D}^X, \theta)]
\]
\[
= \sum_{n=1}^{N} \sum_{k=1}^{K} P(z^{(n)} = k | x^{(n)}, y^{(n)}, \theta^t) \ln(\pi_k \mathcal{N}(y^{(n)}|w_k^T x^{(n)}, \sigma^2))
\]
EM algorithm for mixture of linear regression

- **E step**: $n = 1, \ldots, N, k = 1, \ldots, K$

$$
\gamma^n_k = P \left( z^{(n)} = k | x^{(n)}, y^{(n)}, \theta^{\text{old}} \right) = \frac{\pi^\text{old}_k \mathcal{N} \left( y^{(n)} \mid w^\text{old}_k^T x^{(n)}, \sigma^2 \right)}{\sum_{j=1}^K \pi^\text{old}_j \mathcal{N} \left( y^{(n)} \mid w^\text{old}_j^T x^{(n)}, \sigma^2 \right)}
$$

- **M Step**: $k = 1, \ldots, K$

$$
\pi^\text{new}_k = \frac{\sum_{n=1}^N \gamma^n_k}{N}
$$

\[
\nabla_{w_k} E_{P(\mathcal{H}|\mathcal{D},\theta^t)} \left[ \ln P(\mathcal{D}^Y, \mathcal{H}|\mathcal{D}^X, \theta) \right] = \nabla_{w_k} \sum_{n=1}^N \gamma^n_k \left\{ -\frac{1}{2\sigma^2} \left( y^{(n)} - w_k^T x^{(n)} \right)^2 \right\}
\]

\[
\nabla_{w_k} E_{P(\mathcal{H}|\mathcal{D},\theta^t)} \left[ \ln P(\mathcal{D}^Y, \mathcal{H}|\mathcal{D}^X, \theta) \right] = 0
\]

$$
\Rightarrow w_{k}^{\text{new}} = (X^T R_k X)^{-1} X^T R_k y
$$

$$
R_k = \text{diag}(\gamma^n_k)
$$
Example

$t = 1$

$t = 30$

$t = 50$

[Bishop]
Example

\[ P(y|x, \theta) \]
\[ k = 2 \]

\[ P(y|x, \theta) \]
\[ k = 1 \]
Conditional mixture models: Non-overlapping experts

- We intend to model \( P(y|x) \) using different experts responsible for different regions of the input space.

\[
P(y|x, \theta) = \sum_{k=1}^{K} P(Z = k|x, \xi)P(y|x, Z = k, w, \sigma^2)
\]

Mixing coefficients \( P(Z = k|x, \xi) \) as gating functions
Conditional mixture models: Non-overlapping experts

- Gating functions $\pi_k(x) = P(Z_k = 1|x, \xi)$ must satisfy the usual constraints for mixing coefficients:
  - $0 \leq \pi_k(x) \leq 1$
  - $\sum_{k=1}^{K} \pi_k(x) = 1$

$$P(Z = k|x, \xi) = \text{softmax}(\xi_k^T x) = \frac{e^{\xi_k^T x}}{\sum_{j=1}^{K} e^{\xi_j^T x}}$$
Conditional mixture models: Non-overlapping experts (EM algorithm)

- **E step:** \( n = 1, \ldots, N, k = 1, \ldots, K \)

\[
\gamma^*_k = P \left( z^{(n)} = k | x^{(n)}, y^{(n)}, \theta^{old} \right) = \frac{\pi_k(x^{(n)}, \xi^{old}) \mathcal{N} \left( y^{(n)} \left| w_k^{old} x^{(n)}, \sigma^2 \right. \right)}{\sum_{j=1}^{K} \pi_j(x^{(n)}, \xi^{old}) \mathcal{N} \left( y^{(n)} \left| w_j^{old} x^{(n)}, \sigma^2 \right. \right)}
\]

- **M Step:** \( k = 1, \ldots, K \)

\[
\mathbb{E}_{P(\mathcal{H} | \mathcal{D}, \theta^t)} \left[ \ln P(\mathcal{D}^Y, \mathcal{H} | \mathcal{D}^X, \theta) \right] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma^*_k \ln(\pi_k(x^{(n)}, \xi^{old}) \mathcal{N}(y^{(n)} | w_k^{T} x^{(n)}, \sigma^2))
\]

\[
\nabla_{\xi_k} \mathbb{E}_{P(\mathcal{H} | \mathcal{D}, \theta^t)} \left[ \ln P(\mathcal{D}^Y, \mathcal{H} | \mathcal{D}^X, \theta) \right] = 0
\]

It does not lead to closed form solution for \( \xi_k \) and so needs a gradient-based optimizer similar to LR

\[
\nabla_{w_k} \mathbb{E}_{P(\mathcal{H} | \mathcal{D}, \theta^t)} \left[ \ln P(\mathcal{D}^Y, \mathcal{H} | \mathcal{D}^X, \theta) \right] = \nabla_{w_k} \sum_{n=1}^{N} \gamma^*_k \left\{ -\frac{1}{2\sigma^2} \left( y^{(n)} - w_k^{T} x^{(n)} \right)^2 \right\} = 0
\]

\[
\Rightarrow w_k^{\text{new}} = (X^T R_k X)^{-1} X^T R_k y \quad R_k = \text{diag}(\gamma^*_k)
\]
Undirected models: Learning with incomplete data

- In the incomplete data case, we can use EM algorithm (according to the properties of EM on the exponential family):
  - E step: $M^t_\theta[f_i] = \sum_{n=1}^{N} E_{P(x_h^{(n)}|x_o^{(n)},\theta^t)}[f_i(x_i^{(n)})]$
  - M Step: similar to ML on undirected models with complete data (expected sufficient statistics are used instead of sufficient statistics)

- Or use gradient ascent on $\ell(\theta; D)$ directly:
  - $\nabla_{\theta_i} \ell(\theta; D) = \sum_{n=1}^{N} E_{P(x_h^{(n)}|x_o^{(n)},\theta)}[f_i(x_i^{(n)})] - N \times E_{P(x|\theta)}[f_i(x_i)]$

  $x_i$: shows the scope of the i-th feature
Reference

- Koller & Friedman, Chapter 19.1-19.2.3 and 20.3.3.
- Jordan, Chapters 10-11.