Maximum likelihood learning with complete data
Maximum likelihood estimation

• Suppose we observe $K$ IID training cases $\mathbf{o}^{(1)} \ldots \mathbf{o}^{(K)}$

• $\mathbf{o}^{(k)}$ is the $k$th training case

• Let $\theta$ be the parameters of the local functions in a graphical model

• Maximum likelihood estimate of $\theta$:

$$\theta_{\text{ML}} = \arg\max_{\theta} \prod_{k} P(\mathbf{o}^{(k)}|\theta)$$
Why maximum-likelihood?

• If $\theta$ spans all pdfs and if $K \to \infty$, then the ML estimate matches the true data distribution.

• Decisions based on $P$ maximize utility (are Bayes-optimal).

• …works well in many applications.
Complete data in Bayes nets

• All variables are observed, so

\[ P(o|\theta) = \Pi_i P(o_i|pa_i, \theta), \quad pa_i = \text{parents of } o_i \]

• Since \( \arg\max () = \arg\max \log () \),

\[ \theta_{\text{ML}} = \arg\max_\theta \log \Pi_k P(o^{(k)}|\theta) \]

\[ = \arg\max_\theta \sum_k \log P(o^{(k)}|\theta) \]

\[ \theta_{\text{ML}} = \arg\max_\theta \sum_k \sum_i \log P(o^{(k)}_i|pa^{(k)}_i, \theta) \]
• Let $\theta_i \subseteq \theta$ parameterize $P(o_i|pa_i,\theta_i)$

• Then,

$$\theta_i^{ML} = \arg\max_{\theta_i} \sum_k \log P(o_i^{(k)}|pa_i^{(k)},\theta_i)$$

• Learning the entire Bayes net decouples into learning individual conditional distributions

• The sufficient statistics for learning a Bayes net are the sufficient statistics of the individual conditional distributions
Learning using complete data in occlusion model

- For $j = 1, \ldots, J$

\[
\pi_j \leftarrow \frac{(\sum_{t=1}^{T} [f(t) = j] + \sum_{t=1}^{T} [b(t) = j])}{2T}
\]

- For $j = 1, \ldots, J$, for $i = 1, \ldots, K$

\[
\begin{align*}
\alpha_{ji} & \leftarrow \frac{(\sum_{t=1}^{T} [f(t) = j] m_i^{(t)})}{(\sum_{t=1}^{T} [f(t) = j])} \\
\mu_{ji} & \leftarrow \frac{(\sum_{t=1}^{T} [f(t) = j \text{ or } b(t) = j] z_i^{(t)})}{(\sum_{t=1}^{T} [f(t) = j \text{ or } b(t) = j])} \\
\psi_{ji} & \leftarrow \frac{(\sum_{t=1}^{T} [f(t) = j \text{ or } b(t) = j] (z_i^{(t)} - \mu_{ji})^2)}{(\sum_{t=1}^{T} [f(t) = j \text{ or } b(t) = j])}
\end{align*}
\]

Here, the Iverson notation is used where $[\text{True}] = 1$ and $[\text{False}] = 0.$
Continuous child with continuous parents

- Estimation becomes a regression-type problem

- Eg, linear Gaussian model:

\[ P(o_i|pa_i, \theta_i) = \mathcal{N}(o_i; \theta_i + \sum_{n:o_n \in pa_i} \theta_{in} o_n, C_i), \]

- mean = linear function of parents

- Estimation = linear regression
Complete data in MRFs

- All variables are observed, so

$$P(o|\theta) = \left[ \prod_i \phi_i(c_i|\theta) \right] / Z(\theta)$$

Partition function: Generally intractable!
Coping with the partition function

Use approximate inference techniques

- Variational methods
- Markov chain Monte Carlo
- ...
Complete data in factor graphs

• If factor graph is “directed”, learning proceeds as for Bayes nets

• If factor graph is “undirected”, we get a partition function, just like for MRFs
Incomplete data in Bayes nets

\[ \theta_{\text{ML}} = \arg\max_{\theta} \log \prod_k P(o^{(k)}|\theta) \]

\[ = \arg\max_{\theta} \sum_k \log \left( \sum_{u^{(k)}} P(\mathbf{o}^{(k)}, \mathbf{u}^{(k)}|\theta) \right) \]

\[ = \arg\max_{\theta} \sum_k \log \left( \sum_{u^{(k)}} \prod_i P(x_i^{(k)}|\text{pa}_i, \theta) \right) \]

Problem! Summation gets in the way of \( \log \Pi_i \)
Example: Mixture of 2 unit-variance Gaussians

\[ P(z) = \pi_1 \alpha \exp[-(z-\mu_1)^2/2] + \pi_2 \alpha \exp[-(z-\mu_2)^2/2] \]

where \( \alpha = (2\pi)^{-1/2} \)

The log-likelihood to be maximized is

\[ \log(\pi_1 \alpha \exp[-(z-\mu_1)^2/2] + \pi_2 \alpha \exp[-(z-\mu_2)^2/2]) \]

The parameters \((\pi_1, \mu_1, \pi_2, \mu_2)\) that maximize this do not have a simple, closed form solution

- One approach: Use nonlinear optimizer (eg, Newton-type method). OR…
Expectation Maximization Algorithm
(Dempster, Laird and Rubin 1977)

• Learning was more straightforward when the data was complete

• Can we use probabilistic inference (compute \( P(u|o, \theta) \)) to “fill in” the missing data and then use the learning rules for complete data?

• YES: EM algorithm
The EM algorithm
Expectation Maximization (EM) Algorithm

- Initialize $\theta$ (eg, using 2nd order statistics)
- E-Step: Compute $Q(u) = P(u|o,\theta)$ for all $u$
- M-Step: Holding $Q(u)$ constant maximize $\Sigma_u Q(u) \log P(o,u|\theta)$ wrt $\theta$
- Repeat E and M steps until convergence
- EM consistently increases $\log(\Sigma_u P(o,u|\theta))$

"Ensemble completion"
EM in Bayes nets

- Recall $P(o, u|\theta) = \prod_i P(x_i|pa_i, \theta)$, $x = (o, u)$
- Let $\theta_i \subseteq \theta$ parameterize $P(x_i|pa_i, \theta_i)$
- Then, maximizing

$$\sum_u Q(u) \log P(o, u|\theta)$$

wrt $\theta_i$ becomes equivalent to maximizing

$$\sum_{x_i, pa_i} Q(x_i, pa_i) \log P(x_i|pa_i, \theta_i)$$

- Just like for complete data, learning is decoupled
EM for mixture of Gaussians: E step

\[ \pi_1 = 0.5, \quad \mu_1 = \Phi_1 = \]

\[ \pi_2 = 0.5, \quad \mu_2 = \Phi_2 = \]

Images from data set
EM for mixture of Gaussians: E step

\[ P(c|z) \]

\[ \pi_1 = 0.5, \quad \mu_1 = \text{image} \]

\[ \Phi_1 = \text{rectangle} \quad c = 1 \quad 0.52 \]

\[ \pi_2 = 0.5, \quad \mu_2 = \text{image} \]

\[ \Phi_2 = \text{rectangle} \quad c = 2 \quad 0.48 \]

Images from data set
EM for mixture of Gaussians: E step

\[
P(c|z)\]

\[
\pi_1 = 0.5, \quad \mu_1 = \begin{array}{c}
\text{Image 1}
\end{array}
\]

\[
\phi_1 = \begin{array}{c}
\text{Gaussian 1}
\end{array}
\]

\[
c = 1 \quad 0.51
\]

\[
\pi_2 = 0.5, \quad \mu_2 = \begin{array}{c}
\text{Image 2}
\end{array}
\]

\[
\phi_2 = \begin{array}{c}
\text{Gaussian 2}
\end{array}
\]

\[
c = 2 \quad 0.49
\]
EM for mixture of Gaussians: E step

\[ P(c|z) \]

\[ \pi_1 = 0.5, \quad \mu_1 = \]  
\[ \Phi_1 = \]  
\[ c = 1 \quad 0.48 \]

\[ \pi_2 = 0.5, \quad \mu_2 = \]  
\[ \Phi_2 = \]  
\[ c = 2 \quad 0.52 \]

Images from data set
EM for mixture of Gaussians: E step

\[ P(c|z) \]

\[
\begin{align*}
\pi_1 &= 0.5, \quad \mu_1 = \begin{array}{c}
\end{array} \\
\pi_2 &= 0.5, \quad \mu_2 = \begin{array}{c}
\end{array} \\
\Phi_1 &= \begin{array}{c}
\end{array} \\
\Phi_2 &= \begin{array}{c}
\end{array}
\end{align*}
\]

Images from data set

\begin{align*}
P(c | z) &
\begin{array}{c}
\end{array}
\end{align*}
EM for mixture of Gaussians: M step

\[ \pi_1 = 0.5, \quad \mu_1 = \Phi_1 = \]
\[ \pi_2 = 0.5, \quad \mu_2 = \Phi_2 = \]

Set \( \mu_1 \) to the average of \( z \mathbb{P}(c=1|z) \)
Set \( \mu_2 \) to the average of \( z \mathbb{P}(c=2|z) \)
EM for mixture of Gaussians: M step

\[ \pi_1 = 0.5, \quad \mu_1, \quad \Phi_1 \]
\[ \pi_2 = 0.5, \quad \mu_2, \quad \Phi_2 \]

Set \( \mu_1 \) to the average of \( z P(c=1|z) \)

Set \( \mu_2 \) to the average of \( z P(c=2|z) \)
EM for mixture of Gaussians: M step

\[ \pi_1 = 0.5, \quad \mu_1 = \begin{array}{c}
\end{array} \quad \Phi_1 = \begin{array}{c}
\end{array} \]

\[ \pi_2 = 0.5, \quad \mu_2 = \begin{array}{c}
\end{array} \quad \Phi_2 = \begin{array}{c}
\end{array} \]

Set \( \Phi_1 \) to the average of 
\[ \text{diag}((z-\mu_1)^T (z-\mu_1))P(c=1|z) \]

Set \( \Phi_2 \) to the average of 
\[ \text{diag}((z-\mu_2)^T (z-\mu_2))P(c=2|z) \]
EM for mixture of Gaussians: M step

\[ \pi_1 = 0.5, \quad \mu_1 = \begin{array} \end{array}, \Phi_1 = \begin{array} \end{array} \]
\[ \pi_2 = 0.5, \quad \mu_2 = \begin{array} \end{array}, \Phi_2 = \begin{array} \end{array} \]

Set \( \Phi_1 \) to the average of
\[ \text{diag}((z - \mu_1)^T (z - \mu_1))P(c=1|z) \]

Set \( \Phi_2 \) to the average of
\[ \text{diag}((z - \mu_2)^T (z - \mu_2))P(c=2|z) \]
\[ \pi_1 = 0.6, \quad \mu_1 = \begin{array}{c} \text{Image 1} \\ \text{Image 2} \end{array}, \quad \Phi_1 = \begin{array}{c} \text{Image 3} \\ \text{Image 4} \end{array} \]

\[ \pi_2 = 0.4, \quad \mu_2 = \begin{array}{c} \text{Image 5} \\ \text{Image 6} \end{array}, \quad \Phi_2 = \begin{array}{c} \text{Image 7} \\ \text{Image 8} \end{array} \]
EM vs k-means for clustering

MATLAB demo
Cluster video frames using EM in a mixture of Gaussians

<table>
<thead>
<tr>
<th>Model size</th>
<th># classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 class</td>
</tr>
<tr>
<td>2</td>
<td>2 classes</td>
</tr>
<tr>
<td>3</td>
<td>3 classes</td>
</tr>
<tr>
<td>4</td>
<td>4 classes</td>
</tr>
</tbody>
</table>

Input video

Mixture of Gaussians

Learned cluster centers
Transformed mixture of Gaussians (TMG)

\[ P(c) = \pi_c \]

\[ \pi_1 = 0.6, \quad \mu_1 = \text{Diag}(\Phi_1) = \text{Image} \]

\[ \pi_2 = 0.4, \quad \mu_2 = \text{Diag}(\Phi_2) = \text{Image} \]

\[ P(s) \times T \]

\[ z = \text{Image} \]

\[ P(x|z, T) = N(x; Tz, s), \quad \Psi \]

\[ \text{Diagonal} \]

B. J. FREY
EM for TMG

- E step: Compute $P(T|x)$, $P(c|x)$ and $p(z|T,x)$ for each $x$ in data

- M step: Set
  - $\pi_c = \text{avg of } P(c|x)$
  - $\rho_T = \text{avg of } P(T|x)$
  - $\mu_c = \text{avg mean of } p(z|c,x)$
  - $\Phi_c = \text{avg variance of } p(z|c,x)$
  - $\Psi = \text{avg var of } p(x-Tz|x)$
Inference AFTER learning TMG

\[
P(c|\mathbf{x}) = \mu_{\text{argmax}_c P(c|\mathbf{x})}
\]

\[
\text{E}[\mathbf{z}|\mathbf{x}]
\]

\[
\text{E}[T \mathbf{z}|\mathbf{x}]
\]

\[
\text{argmax}_T P(T|\mathbf{x})
\]
Inferred appearance means, $\mu$, in transformed mixture of Gaussians

TMG - Frey & Jojic, 1999-2001

Input video

Learned cluster centers

Model size

Random initialization

# classes

- 1 class
- 2 classes
- 3 classes
- 4 classes

B. J. FREY
Peering inside the Bayes net:
A look at the expected transformed latent image

\[ x \quad E[Ts|x] \]
tmgEM.m and tmglPx.m on the web
How to pick the number of classes

- Cost function (MAP)
- Validation set
- Bayesian methods
- ...