ECE276A: Sensing & Estimation in Robotics
Lecture 4: Expectation Maximization

Lecturer:

Nikolay Atanasov: natanasov@ucsd.edu

Teaching Assistants:

Siwei Guo: s9guo@eng.ucsd.edu

Anwesan Pal: a2pal@eng.ucsd.edu

UC San Diego

JACOBS SCHOOL OF ENGINEERING

Electrical and Computer Engineering

Gaussian Discriminant Analysis

- **Generative model**: $h(\mathbf{x}) := \arg\max_y p(y, \mathbf{x})$
- **Maximum Likelihood Estimation (MLE)**: $\max_{\theta,\omega} p(\mathbf{y}, X \mid \theta, \omega)$
- Gaussian (Mixture) Discriminant Analysis: uses a **Gaussian Mixture** with J components to model $p(\mathbf{x}_i \mid y_i, \omega)$:

$$p(\mathbf{y}, X \mid \omega, \theta) = p(\mathbf{y} \mid \theta) p(X \mid \mathbf{y}, \omega) = p(\mathbf{y} \mid \theta) \prod_{i=1}^{n} p(\mathbf{x}_i \mid y_i, \omega)$$

$$p(\mathbf{y} \mid \theta) := \prod_{i=1}^{n} \prod_{k=1}^{K} \theta_k^{\mathbb{1}\{y_i = k\}} \quad p(\mathbf{x}_i \mid y_i = k, \omega) := \sum_{j=1}^{J} \alpha_{kj} \phi(\mathbf{x}_i; \mu_{kj}, \Sigma_{kj})$$

- The MLE of θ can be obtained via the softmax trick and differentiation
- Obtaining MLE estimates for $\omega := \{\alpha_{kj}, \mu_{kj}, \Sigma_{kj}\}$ is no longer straight forward because $\log \sum_{j=1}^{J} \alpha_{kj} \phi(\mathbf{x}_i; \mu_{kj}, \Sigma_{kj})$ is not convex/concave
- Also, need to ensure that $\sum_{j=1}^{J} \alpha_{kj} = 1, \ \forall k$.

Data Log Likelihood

- $\log p(\mathbf{y}, X \mid \omega, \theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}\{y_i = k\} \log \theta_k$
  $$+ \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}\{y_i = k\} \log \left(\sum_{j=1}^{J} \alpha_{kj} \phi(\mathbf{x}_i; \mu_{kj}, \Sigma_{kj}) \right)$$
- Focus on max wrt $\omega := \{\alpha_{kj}, \mu_{kj}, \Sigma_{kj}\}$; the first term can be ignored
- To simplify notation, let $D_k := \{\mathbf{x}_i, y_i \mid y_i = k\} \subseteq D$ and define:

$$\lambda(X, \omega) := \sum_{k=1}^{K} \sum_{\mathbf{x} \in D_k} \log \left(\sum_{j=1}^{J} \alpha_{kj} \phi(\mathbf{x}; \mu_{kj}, \Sigma_{kj}) \right)$$

Membership Probabilities

- Gaussian Mixtures are well suited for modeling clusters of points:
  - each cluster is assigned a Gaussian
  - the mean is somewhere in the middle of the cluster
  - the covariance measures the cluster spread
- **Sampling**
  - Draw an integer between 1 and J with probability α_{kj}
  - Draw a vector $\mathbf{x}$ from the j-th Gaussian pdf $\phi(\mathbf{x}; \mu_{kj}, \Sigma_{kj})$
- It is useful to understand the meaning of $q_k(j, \mathbf{x}) := \alpha_{kj}\phi(\mathbf{x}; \mu_{kj}, \Sigma_{kj})$
- Given class k, $q_k(j, \mathbf{x})d\mathbf{x}$ is the joint probability of drawing component j and data point $\mathbf{x}$ in a volume $d\mathbf{x}$ around it
- **Membership probabilities** the conditional probability of having selected component j given data point $\mathbf{x}$:

$$r_k(j \mid \mathbf{x}) := \frac{q_k(j, \mathbf{x})}{\sum_{l=1}^{J} q_k(l, \mathbf{x})} \qquad \sum_{j=1}^{J} r_k(j \mid \mathbf{x}) = 1$$

Local maxima of $\lambda(X, \omega)$

- Maxima of $\sum_{k=1}^{K} \sum_{\mathbf{x} \in D_k} \log \left(\sum_{j=1}^{J} \alpha_{kj} \phi(\mathbf{x}; \mu_{kj}, \Sigma_{kj}) \right)$ occur at critical points
- $$\frac{d}{d\mu_{lm}} \lambda(X, \omega) = \sum_{\mathbf{x} \in D_l} \frac{\alpha_{lm}}{\sum_{j=1}^{J} \alpha_{lj} \phi(\mathbf{x}; \mu_{lj}, \Sigma_{lj})} \frac{d}{d\mu_{lm}} \phi(\mathbf{x}; \mu_{lm}, \Sigma_{lm})$$
$$= \sum_{\mathbf{x} \in D_l} r_l(m \mid \mathbf{x}) (\mu_{lm} - \mathbf{x})^T \Sigma_{lm}^{-1}$$
- $$\frac{d}{d\Sigma_{lm}} \lambda(X, \omega) = \frac{1}{2} \sum_{\mathbf{x} \in D_l} r_l(m \mid \mathbf{x}) \left(\Sigma_{lm}^{-1} (\mu_{lm} - \mathbf{x})(\mu_{lm} - \mathbf{x})^T \Sigma_{lm}^{-1} - \Sigma_{lm}^{-1} \right)$$
- Use **softmax trick** for α_{kj} to handle simplex constraints

$$\frac{d}{d\gamma_{lm}} \lambda(X, \omega) = \sum_{\mathbf{x} \in D_l} \frac{1}{\sum_{j=1}^{J} \alpha_{lj} \phi(\mathbf{x}; \mu_{lj}, \Sigma_{lj})} \sum_{j=1}^{J} \frac{d\alpha_{lj}}{d\gamma_{lm}} \phi(\mathbf{x}; \mu_{lj}, \Sigma_{lj})$$
$$= \sum_{\mathbf{x} \in D_l} (r_l(m \mid \mathbf{x}) - \alpha_{lm})$$

Local maxima of $\lambda(X, \omega)$

- Setting the previous derivatives to zero, we obtain:

$$\boxed{\alpha_{kj} = \frac{\sum_{i=1}^n \mathbb{1}\{y_i = k\} r_k(j \mid \mathbf{x}_i)}{\sum_{i=1}^n \mathbb{1}\{y_i = k\}}} \qquad \boxed{\mu_{kj} = \frac{\sum_{i=1}^n \mathbb{1}\{y_i = k\} r_k(j \mid \mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^n \mathbb{1}\{y_i = k\} r_k(j \mid \mathbf{x}_i)}}$$

$$\boxed{\Sigma_{kj} = \frac{\sum_{i=1}^n \mathbb{1}\{y_i = k\} r_k(j \mid \mathbf{x}_i)(\mathbf{x}_i - \mu_{kj})(\mathbf{x}_i - \mu_{kj})^T}{\sum_{i=1}^n \mathbb{1}\{y_i = k\} r_k(j \mid \mathbf{x}_i)}}$$

- The mixture weights are equal to the sample mean of the membership probabilities $r_k(j \mid \mathbf{x}_i)$ assuming a uniform distribution over D_k
- The latter are the sample mean and covariance of the data, weighted by the membership probabilities
- The three equations are coupled through $r_k(j \mid \mathbf{x})$ and hence are hard to solve directly
- **Idea**: start with a guess $\omega^{(0)}$ and iterate between updating $r_k(j \mid \mathbf{x}_i)$ and updating $\omega^{(t)}$

Clustering

- How do we obtain an initial guess $\omega^{(0)} := \left\{\alpha_{kj}^{(0)}, \mu_{kj}^{(0)}, \Sigma_{kj}^{(0)}\right\}$?
- **Clustering** (or vector quantization) is the task of grouping objects in a way that those in the same group (a **cluster**) are more similar (according to a distance metric) to each other than to those in other groups.
- **Unsupervised Learning**: given an *unlabeled* dataset $D = \{\mathbf{x}_i\}_{i=1}^{n}$, the goal is to partition it into J clusters

k-means Algorithm

- The k-**means algorithm** is an iterative clustering algorithm that uses **coordinate descent** to solve the following optimization:

$$\min_{\mu, r} C(\mu, r) := \sum_{i=1}^{n} \sum_{j=1}^{J} r_{ij} \|\mu_j - \mathbf{x}_i\|_2^2$$

- μ_j are cluster centroids, $r_{ij} := \mathbb{1}_{\{\mathbf{x}_i \text{ is closest to } \mu_j\}}$ are cluster membership indicators

- It is common to repeat the algorithm several times with different initialization of μ_j

- Since k-means is optimizing $\|\cdot\|_2$, it implicitly makes a spherical assumption on the shape of the clusters.

k-means Algorithm

Algorithm 1 k-means clustering

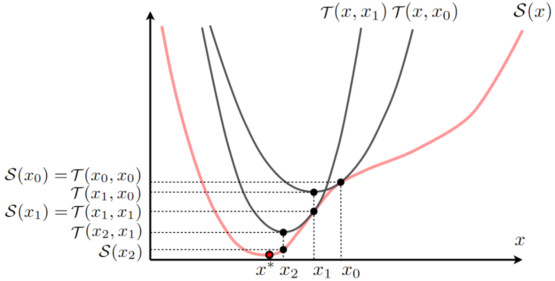
Input: unlabeled dataset $D = \{\mathbf{x}_i\}_{i=1}^n$, number of clusters k
Output: cluster centroids μ_j, cluster assignments $\{r_{ij}\}$
Init: pick k cluster centroids $\mu_1, \ldots, \mu_k$
repeat
  # *Assign examples to the nearest centroid*:
    $r_{ij} = 1$, if $j = \arg\min_l \|\mu_l - x_i\|_2^2$, and $r_{ij} = 0$, otherwise.
  # *Set each centroid to the mean of the examples assigned to it*:
    $\mu_j = \arg\min_\mu C(\mu, r) = \frac{\sum_{i=1}^n r_{ij} x_i}{\sum_{i=1}^n r_{ij}}$
until convergence

Expectation Maximization

- Iterative maximization technique based on auxiliary lower bounds
  - Old idea (late 50's) but formalized by Dempster, Laird and Rubin in 1977
  - Subject of much investigation. See McLachlan & Krishnan book, 1997.
  - Has two steps: Expectation (E) and Maximization (M)
  - Generalizes k-means to soft cluster assignments
- Applicable to a wide range of problems:
  - Fitting mixture models
  - Probabilistic latent semantic analysis: produce concepts related to documents and terms (NLP)
  - Learning parts and structure models (vision)
  - Segmentation of layers in video (vision)

Expectation Maximization

- **Goal**: $\max_\omega \mathcal{S}(\omega)$
- $\mathcal{S}(\omega)$ is not necessarily concave



- Initialize $\omega^{(0)}$ and iterate the following:
  E. Construct an auxiliary lower-bound function $\mathcal{T}$ at $\omega^{(t)}$ such that:
  $$\mathcal{S}(\omega^{(t)}) = \mathcal{T}(\omega^{(t)}, \omega^{(t)}) \geq \mathcal{T}(\omega, \omega^{(t)})$$
  M. Solve the easier auxiliary maximization to obtain the next point:
  $$\omega^{(t+1)} = \arg\max_\omega \mathcal{T}(\omega, \omega^{(t)})$$
- The properties of $\mathcal{T}$ guarantee that each step gets closer to a local max:
$$\mathcal{S}(\omega^{(t)}) = \mathcal{T}(\omega^{(t)}, \omega^{(t)}) \leq \max_\omega \mathcal{T}(\omega, \omega^{(t)}) \leq S(\omega^{(t+1)})$$

Auxiliary Function

- EM is related to MLE since it can be used to solve a problem of the form: $\max_\omega \log p(D; \omega)$, which might be too hard to solve by simply setting the gradient to zero.
- In the context of MLE, EM uses **latent/hidden variables** to construct an auxiliary lower-bound to the data log likelihood via:
  - **Jensen's Inequality**: $f(\mathbb{E}[Z]) \leq \mathbb{E}[f(Z)]$ for convex f
  - e.g.: $\log\left(\sum_j z_j\right) = \log\left(\sum_j r_j \frac{z_j}{r_j}\right) \geq \sum_j r_j \log\left(\frac{z_j}{r_j}\right)$ for $\sum_j r_j = 1$ and $r_j \geq 0$

Auxiliary Function

- Introduce a latent random variable Z with pdf $r(z \mid D)$:

$$
\begin{aligned}
\log p(D;\omega) &\overset{\text{Total law}}{\underset{\text{of prob.}}{=\!=\!=}} \log \int p(D,z;\omega)dz = \log \int r(z|D)\frac{p(D,z;\omega)}{r(z|D)}dz \\
&\underset{\text{inequality}}{\overset{\text{Jensen's}}{\geq}} \int r(z|D) \log \frac{p(D,z;\omega)}{r(z|D)}dz \overset{\text{Auxiliary}}{\underset{\text{function}}{=\!=\!=}} \mathcal{T}(\omega, r)
\end{aligned}
$$

- Assuming that $\log p(D,z;\omega)$ is concave in ω, the **auxiliary function** is concave in ω for a fixed r and concave in r for a fixed ω (but **not jointly** concave)

- The local maxima of $\mathcal{T}(\omega, r)$ are local maxima of $\log p(D;\omega)$

$$
\begin{aligned}
&\text{(E step)} \qquad & r(\cdot \mid D) &= \arg\max_{s(\cdot|D)} \mathcal{T}(\omega, s) \\
&\text{(M step)} \qquad & \omega' &= \arg\max_{\omega} \mathcal{T}(\omega, r)
\end{aligned}
$$

E Step Details

- $r(\cdot \mid D) \xlongequal{why?} \underset{s(\cdot|D)}{\arg\max}\, \mathcal{T}(\omega, s)$

- $$\begin{aligned} \log p(D; \omega) \geq \mathcal{T}(\omega, s) &= \int s(z|D) \log \frac{r(z|D)p(D;\omega)}{s(z|D)} dz \\ &= \log p(D; \omega) - d_{\mathcal{KL}}(r(\cdot \mid D) || s(\cdot \mid D)) \end{aligned}$$

- When maximizing the lower bound $\mathcal{T}(\omega, s)$ with respect to s, we are maximizing the similarity between $s(\cdot \mid D)$ and the conditional pdf $r(\cdot \mid D)$ of the latent variable Z

- Choosing the optimal $s^*(\cdot \mid D) \equiv r(\cdot \mid D)$ makes the lower bound $\mathcal{T}(\omega, s^*)$ **tight**, i.e., it touches the log-likelihood function at ω:

$$\mathcal{T}(\omega, s^*) = \mathcal{T}(\omega, r) = \int r(z \mid D) \log p(D; \omega) dz = \log p(D; \omega)$$

M Step Details

- $\max_\omega \mathcal{T}(\omega, r) = \int r(z|D) \log \frac{p(D,z;\omega)}{r(z|D)} dz$
$$= \underbrace{h(r(\cdot \mid D))}_{\substack{\text{Entropy of } r; \\ \text{does not depend on } \omega}} + \underbrace{\int r(z|D) \log p(D,z;\omega) dz}_{\substack{\text{Weighted MLE where labeled examples} \\ \{(\mathbf{x}_i, y_i, z_i)\} \text{ are weighted by } r(z_i \mid D)}}$$

Auxiliary Function for GM Log Likelihood

- **Latent variable**: soft cluster assignment Z with pdf $r_k^{(t)}(\cdot \mid \mathbf{x})$
- Lower-bound the Gaussian Mixture log likelihood via Jensen's:

$$\begin{aligned}\lambda(X,\omega) &:= \sum_{k=1}^{K} \sum_{\mathbf{x}\in D_k} \log\left(\sum_{j=1}^{J} q_k(j,\mathbf{x})\right) \\ &\geq \sum_{k=1}^{K} \sum_{\mathbf{x}\in D_k} \sum_{j=1}^{J} r_k^{(t)}(j \mid \mathbf{x}) \log \frac{q_k(j,\mathbf{x})}{r_k^{(t)}(j \mid \mathbf{x})} =: \mathcal{T}(\omega,\omega^{(t)})\end{aligned}$$

- A theoretical construction only since we already know that the maximum of $\mathcal{T}(\omega^{(t)}, s)$ occurs at $r_1^{(t)}(\cdot \mid x), \ldots, r_K^{(t)}(\cdot \mid x)$

Gaussian Mixture MLE via EM (summary)

- Start with initial guess $\omega^{(t)} := \left\{\alpha_{kj}^{(t)}, \mu_{kj}^{(t)}, \Sigma_{kj}^{(t)}\right\}$ for $t = 0$, $k = 1, \ldots, K$, $j = 1, \ldots, J$ and iterate:

(E step)
$$\boxed{r_k^{(t)}(j \mid \mathbf{x}_i) = \frac{\alpha_{kj}^{(t)} \phi\left(\mathbf{x}_i; \mu_{kj}^{(t)}, \Sigma_{kj}^{(t)}\right)}{\sum_{l=1}^{J} \alpha_{kl}^{(t)} \phi\left(\mathbf{x}_i; \mu_{kl}^{(t)}, \Sigma_{kl}^{(t)}\right)}}$$

(M step)
$$\boxed{\alpha_{kj}^{(t+1)} = \frac{\sum_{i=1}^{n} \mathbb{1}\{y_i = k\} r_k^{(t)}(j \mid \mathbf{x}_i)}{\sum_{i=1}^{n} \mathbb{1}\{y_i = k\}}}$$

$$\boxed{\mu_{kj}^{(t+1)} = \frac{\sum_{i=1}^{n} \mathbb{1}\{y_i = k\} r_k^{(t)}(j \mid \mathbf{x}_i) \mathbf{x}_i}{\sum_{i=1}^{n} \mathbb{1}\{y_i = k\} r_k^{(t)}(j \mid \mathbf{x}_i)}}$$

$$\boxed{\Sigma_{kj}^{(t+1)} = \frac{\sum_{i=1}^{n} \mathbb{1}\{y_i = k\} r_k^{(t)}(j \mid \mathbf{x}_i) \left(\mathbf{x}_i - \mu_{kj}^{(t+1)}\right)\left(\mathbf{x}_i - \mu_{kj}^{(t+1)}\right)^T}{\sum_{i=1}^{n} \mathbb{1}\{y_i = k\} r_k^{(t)}(j \mid \mathbf{x}_i)}}$$

Gaussian Mixture MLE via EM (comments)

- Sometimes the data is not enough to estimate all these parameters:
  - Fix the weights $\alpha_{kj} = \frac{1}{J}$
  - Fix diagonal $\Sigma_{kj} = \textbf{diag}\left([\sigma_{kj1}^2, \ldots, \sigma_{kjn}^2]^T\right)$ or spherical $\Sigma_{kj} = \sigma_{kj}^2 I_n$
  - Estimate a **diagonal covariance**:

$$\Sigma_{kj}^{(t+1)} = \frac{\sum_{i=1}^n \mathbb{1}\{y_i = k\} r_k^{(t)}(j \mid \mathbf{x}_i) \textbf{diag}\left(\mathbf{x}_i - \mu_{kj}^{(t+1)}\right)^2}{\sum_{i=1}^n \mathbb{1}\{y_i = k\} r_k^{(t)}(j \mid \mathbf{x}_i)}$$

  - Estimate a **spherical covariance**:

$$\sigma_{kj}^{2,(t+1)} = \frac{1}{d} \frac{\sum_{i=1}^n \mathbb{1}\{y_i = k\} r_k^{(t)}(j \mid \mathbf{x}_i) \left\|\mathbf{x}_i - \mu_{kj}^{(t+1)}\right\|^2}{\sum_{i=1}^n \mathbb{1}\{y_i = k\} r_k^{(t)}(j \mid \mathbf{x}_i)}, \qquad \mathbf{x}_i \in \mathbb{R}^d$$

- How should we initialize $\omega^{(0)}$? Use k-**means++**! If $\sigma_{kj} \to 0$, the GM component assignments of EM become hard and EM works like k-means.