Note from Prof. Miguel Carreira-Perpiñán

Mixture densities and the EM algorithm

- *Mixture density* with K components: $p(\mathbf{x};\boldsymbol{\Theta}) = \sum_{k=1}^{K} p(\mathbf{x}|k)p(k)$ $\begin{cases} p(\mathbf{x}|k) & \textit{component densities} \\ p(k)=\pi_k & \textit{mixture proportions.} \end{cases}$
- Ex: Gaussian mixture: $\mathbf{x}|k \sim \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$. Mixture parameters: $\boldsymbol{\Theta} = \{\pi_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\}_{k=1}^K$.
- *Maximum likelihood* estimation of Gaussian mixture parameters: given a sample $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$:
$$\max_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\Theta};\mathcal{X}) = \sum_{n=1}^N \log p(\mathbf{x}_n;\boldsymbol{\Theta}) = \sum_{n=1}^N \log\left(\sum_{k=1}^K p(\mathbf{x}|k)p(k)\right).$$
- $\mathcal{L}$ cannot be maximized in closed form over $\boldsymbol{\Theta}$; it needs an iterative optimization algorithm. Many such algorithms exist (such as gradient descent), but there is a specially convenient one for mixture models (and more generally, for maximum likelihood with missing data).
- *Expectation-Maximization (EM) algorithm*: for Gaussian mixtures:
  - *E step*: given the current parameter values $\boldsymbol{\Theta}$, compute the posterior probability of component k given data point $\mathbf{x}_n$ (for each $k=1,\dots,K$ and $n=1,\dots,N$):

$$z_{nk} = p(k|\mathbf{x}_n;\boldsymbol{\Theta}) = \frac{p(\mathbf{x}_n|k)p(k)}{p(\mathbf{x}_n;\boldsymbol{\Theta})} = \frac{\pi_k|2\pi\boldsymbol{\Sigma}_k|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_k)\right)}{\sum_{k'=1}^K \pi_{k'}|2\pi\boldsymbol{\Sigma}_{k'}|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}_n-\boldsymbol{\mu}_{k'})^T\boldsymbol{\Sigma}_{k'}^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_{k'})\right)} \in (0,1).$$

  - *M step*: given the posterior probabilities, estimate the parameters $\boldsymbol{\Theta}$: for $k=1,\dots,K$:

$$\pi_k = \frac{1}{N}\sum_{n=1}^N z_{nk} \qquad \boldsymbol{\mu}_k = \frac{\sum_{n=1}^N z_{nk}\mathbf{x}_n}{\sum_{n=1}^N z_{nk}} \qquad \boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^N z_{nk}(\mathbf{x}_n-\boldsymbol{\mu}_k)(\mathbf{x}_n-\boldsymbol{\mu}_k)^T}{\sum_{n=1}^N z_{nk}}.$$

  Similar to k-means, where the assignment and centroid steps correspond to the E and M steps. But in EM the assignments are *soft* ($z_{nk} \in [0,1]$), while in k-means they are *hard* ($z_{nk} \in \{0,1\}$).
- If we knew which component $\mathbf{x}_n$ came from for each $n=1,\dots,N$, we'd not need the E step: a single M step that estimates each component's parameters on its set of points would suffice. This was the case in classification (where $\mathbf{x}|C_k$ is Gaussian): we were given $(\mathbf{x}_n, y_n)$.
- Each EM step increases $\mathcal{L}$ or leaves it unchanged, but it takes an infinite number of iterations to converge. In practice, we stop when the parameters don't change much, or when the number of iterations reaches a limit.
- EM converges to a local optimum that depends on the initial value of $\boldsymbol{\Theta}$. Usually from k-means[?].
- User parameter: number of clusters K. Output: posterior probabilities $\{p(k|\mathbf{x}_n)\}$ and $\{\pi_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\}_{k=1}^K$.
- *Parametric clustering*: K clusters, assumed Gaussian.
- The fundamental advantage of Gaussian mixtures over k-means for clustering is that we can model the uncertainty in the assignments (particularly useful for points near cluster boundaries), and the clusters can be elliptical and have different proportions.

	k-means	EM for Gaussian mixtures
assignments z_{nk}	hard	soft, $p(k\|\mathbf{x}_n)$
probability model?	no	yes
number of iterations	finite	infinite
parameters	centroids $\{\boldsymbol{\mu}_k\}_{k=1}^K$	$\{\pi_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\}_{k=1}^K$