STAT 339
Clustering III and IV

April 29 and May 1, 2020

A Closer Look at EM
Regularized and Bayesian Clustering

Outline

Mixture Model

Our generative model for data that we think has clusters is a **mixture model**:

$$p(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \prod_{n=1}^{N} p(z_n \mid \boldsymbol{\pi}) p(\mathbf{x}_n \mid z_n, \boldsymbol{\theta})$$
$$= \prod_{n=1}^{N} \pi_{z_n} p_{z_n}(\mathbf{x}_n \mid \boldsymbol{\theta}_{z_n})$$

where, marginalizing out clusters we have

$$p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \left(\sum_{z_n=1}^{K} \pi_{z_n} p_{z_n}(\mathbf{x}_n \mid \boldsymbol{\theta}_{z_n}) \right)$$

MLE for the Mixture Model

Direct analytic maximization of the likelihood for $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ is intractable, but we can use **iterative optimization** in the form of the **Expectation-Maximization (EM) algorithm**

0. Initialize parameters $\hat{\boldsymbol{\pi}}^{(0)}$ and $\hat{\boldsymbol{\theta}}^{(0)}$
1. Until converged, iterate, alternating between:
   (a) (E-step) Set $\mathbf{Q}^{(m)} = (q_{nk}^{(m)})$ using

$$q_{nk}^{(m)} \leftarrow p(z_n = k \mid \mathbf{x}_n, \hat{\boldsymbol{\pi}}^{(m-1)}, \hat{\boldsymbol{\theta}}^{(m-1)})$$

   The q_{nk}s are called **responsibilities** and are computed using Bayes' rule as for a Bayes' classifier

   (b) (M-step) Set $\hat{\boldsymbol{\pi}}^{(m)}$ and $\hat{\boldsymbol{\theta}}^{(m)}$ by conditional MLE:

$$\boldsymbol{\pi}_k^{(m)} \leftarrow \operatorname*{argmax}_{\boldsymbol{\pi}} p(\mathbf{Q}^{(m)} \mid \boldsymbol{\pi})$$

$$\hat{\boldsymbol{\mu}}_k^{(m)}, \boldsymbol{\Sigma}_k^{(m)} \leftarrow \operatorname*{argmax}_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k} p(\tilde{\mathbf{X}}_k^{(m)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \mathbf{Q}^{(m)})$$

Goals

- Why **Expectation**-Maximization?
- Proof that EM finds a local maximum
- Some practical issues implementing EM
- Regularized/Bayesian Mixture Modeling
- Model Selection

Outline

Outline

A Closer Look at the E-step

- The "E" stands for "Expectation". What expectation are we taking?

A Closer Look at the E-step

- The "E" stands for "Expectation". What expectation are we taking?
- Consider an "vector of indicator variables" representation of z_n: a binary vector of all zeroes, except for a 1 in position k. For example:

$$\tilde{\mathbf{z}}_n = (0, 0, 1, 0), \text{ equivalent to } z_n = 3 \text{ with } K = 4$$

A Closer Look at the E-step

- The "E" stands for "Expectation". What expectation are we taking?
- Consider an "vector of indicator variables" representation of z_n: a binary vector of all zeroes, except for a 1 in position k. For example:

$$\tilde{\mathbf{z}}_n = (0, 0, 1, 0), \text{ equivalent to } z_n = 3 \text{ with } K = 4$$

- Then q_{nk} is the conditional mean of $\tilde{z}_{nk}$:

$$\begin{aligned}
&\mathbb{E}\left[\tilde{z}_{nk} \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}, \mathbf{X}\right] \\
&= 0 \cdot p(\tilde{z}_{nk} = 0 \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}, \mathbf{X}) + 1 \cdot p(\tilde{z}_{nk} = 1 \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}, \mathbf{X}) \\
&= p(z_n = k \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}, \mathbf{X}) \\
&=: q_{nk}
\end{aligned}$$

Outline

Technical Digression: Local Maximum Property

- EM finds a **local maximum** of the likelihood function

Technical Digression: Local Maximum Property

- EM finds a **local maximum** of the likelihood function
- How do we know?

Technical Digression: Local Maximum Property

- EM finds a **local maximum** of the likelihood function
- How do we know?
- We can write the log likelihood as

$$\begin{aligned}
\log p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) &= \sum_{n=1}^{N} \log \left(p(\mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\theta}) \right) \\
&= \sum_{n=1}^{N} \log \left(\sum_{z_n=1}^{K} p(z_n, \mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\theta}) \right) \\
&= \sum_{n=1}^{N} \log \left(\sum_{z_n=1}^{K} q_{z_n k} \frac{p(z_n, \mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\theta})}{q_{z_n k}} \right) \\
&= \sum_{n=1}^{N} \log \mathbb{E}_{\mathbf{Q}} \left[\frac{p(z_n, \mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\theta})}{q_{z_n k}} \right]
\end{aligned}$$

where the expectation in the last line is with respect to the distribution on z_n encoded by $\mathbf{Q}$

Technical Digression: Local Maximum Property

The log likelihood is

$$\log L(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}) = \sum_{n=1}^{N} \log \mathbb{E}_{\mathbf{Q}} \left[\frac{p(z_n, \mathbf{x}_n \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}})}{q_{z_n k}} \right]$$

Since $\log$ is a concave function, the log of each expected value is at least as large as the expected value of the log (by Jensen's inequality). So, we have a lower bound on the log likelihood:

$$\log L(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}) \geq \sum_{n=1}^{N} \mathbb{E}_{\mathbf{Q}} \left[\log \left(\frac{p(z_n, \mathbf{x}_n \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}})}{q_{z_n k}} \right) \right]$$

Technical Digression: Local Maximum Property

Examining this lower bound:

$$\begin{aligned}
\log L(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}) &\geq \sum_{n=1}^{N} \mathbb{E}_{\mathbf{Q}} \left[\log \left(\frac{p(z_n, \mathbf{x}_n \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}})}{q_{z_n k}} \right) \right] \\
&= \sum_{n=1}^{N} \mathbb{E}_{\mathbf{Q}} \left[\log p(z_n, \mathbf{x}_n \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}) \right] - \sum_{n=1}^{N} \mathbb{E}_{\mathbf{Q}} \left[\log(q_{nk}) \right] \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} q_{nk} \log(\hat{\pi}_k p_k(\mathbf{x}_n \mid \hat{\boldsymbol{\theta}}_k)) - \sum_{n} \mathbb{E}_{\mathbf{Q}} \left[\log(q_{nk}) \right]
\end{aligned}$$

Technical Digression: Local Maximum Property

The log likelihood is bounded below by

$$\log L(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}) \geq \sum_{n=1}^{N} \sum_{k=1}^{K} q_{nk} \log(\hat{\pi}_k p_k(\mathbf{x}_n \mid \hat{\boldsymbol{\theta}}_k)) - \sum_n \mathbb{E}_{\mathbf{Q}} [\log(q_{nk})]$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \log(\hat{\pi}_k^{q_{nk}}) + \sum_{n=1}^{N} \sum_{k=1}^{K} \log(p_k(\mathbf{x}_n \mid \hat{\boldsymbol{\theta}}_k)^{q_{nk}})$$

$$- \sum_n \mathbb{E}_{\mathbf{Q}} [\log(q_{nk})]$$

- The M step in EM involves maximizing the first two terms w.r.t. $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, respectively.
- Since the last term does not depend on $\boldsymbol{\pi}$ or $\boldsymbol{\theta}$, the M step does not affect it
- Therefore, the M step increases the lower bound on the log likelihood.

Technical Digression: Local Maximum Property

What about the E step? Rewriting the bound on the log likelihood:

$$
\begin{aligned}
&\log L(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}) \\
&\geq \sum_{n=1}^{N} \mathbb{E}_{\mathbf{Q}} \left[\log \left(\frac{p(z_n, \mathbf{x}_n \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}})}{q_{nk}} \right) \right] \\
&= \sum_{n=1}^{N} \mathbb{E}_{\mathbf{Q}} \left[\log \left(\frac{p(\mathbf{x}_n \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}) p(z_n \mid \mathbf{x}_n, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}})}{q_{nk}} \right) \right] \\
&= \sum_{n=1}^{N} \mathbb{E}_{\mathbf{Q}} \left[\log p(\mathbf{x}_n \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}) \right] + \sum_{n=1}^{N} \mathbb{E}_{\mathbf{Q}} \left[\log \frac{p(z_n \mid \mathbf{x}_n, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}})}{q_{nk}} \right] \\
&= \sum_{n=1}^{N} \log p(\mathbf{x}_n \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}) - \sum_{n=1}^{N} \sum_{k} q_{nk} \log \left(\frac{q_{nk}}{p(z_n \mid \mathbf{x}_n, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}})} \right)
\end{aligned}
$$

- The first term is the actual log likelihood, and the second is the "slack" in the lower bound
- By setting $q_{nk} = p(z_n = k \mid \mathbf{x}_n, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}})$ the "slack" is zero

Technical Digression: Local Maximum Property

- We have just proved the following:

Technical Digression: Local Maximum Property

- We have just proved the following:
  1. During the M-step, the "floor" for our current log likelihood (evaluated at $\hat{\boldsymbol{\pi}}^{(m)}$ and $\boldsymbol{\theta}^{(m)}$) increases

Technical Digression: Local Maximum Property

- We have just proved the following:
  1. During the M-step, the "floor" for our current log likelihood (evaluated at $\hat{\boldsymbol{\pi}}^{(m)}$ and $\boldsymbol{\theta}^{(m)}$) increases
  2. During the E-step, the "floor" increases to the actual log likelihood (evaluated at $\hat{\boldsymbol{\pi}}^{(m)}$ and $\boldsymbol{\theta}^{(m)}$)

Technical Digression: Local Maximum Property

- We have just proved the following:
  1. During the M-step, the "floor" for our current log likelihood (evaluated at $\hat{\boldsymbol{\pi}}^{(m)}$ and $\boldsymbol{\theta}^{(m)}$) increases
  2. During the E-step, the "floor" increases to the actual log likelihood (evaluated at $\hat{\boldsymbol{\pi}}^{(m)}$ and $\boldsymbol{\theta}^{(m)}$)
- Hence the "floor" is monotonically increasing; and since the current log likelihood at $\hat{\pi}, \hat{\boldsymbol{\theta}}$ is at that floor after each E step, the updates to $\hat{\boldsymbol{\pi}}$ and $\hat{\boldsymbol{\theta}}$ increase the log likelihood at each iteration.

Technical Digression: Local Maximum Property

- We have just proved the following:
  1. During the M-step, the "floor" for our current log likelihood (evaluated at $\hat{\boldsymbol{\pi}}^{(m)}$ and $\boldsymbol{\theta}^{(m)}$) increases
  2. During the E-step, the "floor" increases to the actual log likelihood (evaluated at $\hat{\boldsymbol{\pi}}^{(m)}$ and $\boldsymbol{\theta}^{(m)}$)
- Hence the "floor" is monotonically increasing; and since the current log likelihood at $\hat{\pi}, \hat{\boldsymbol{\theta}}$ is at that floor after each E step, the updates to $\hat{\boldsymbol{\pi}}$ and $\hat{\boldsymbol{\theta}}$ increase the log likelihood at each iteration.
- Therefore, EM is guaranteed to reach a local maximum.

Outline

EM for Missing Data

- The EM algorithm is highly general — not just for mixture models/clustering

EM for Missing Data

- The EM algorithm is highly general — not just for mixture models/clustering
- Can be used any time there is a latent variable whose absence from the data makes likelihood function intractable (e.g., supervised learning w/ missing features; semi-supervised learning with only a few labeled instances)

EM for Missing Data

- The EM algorithm is highly general — not just for mixture models/clustering
- Can be used any time there is a latent variable whose absence from the data makes likelihood function intractable (e.g., supervised learning w/ missing features; semi-supervised learning with only a few labeled instances)
- General structure: Want

$$\operatorname*{argmax}_{\theta} p(\mathbf{X} \mid \boldsymbol{\theta}) = \operatorname*{argmax}_{\theta} \int_{\mathbf{z}} p(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\theta}) \, d\mathbf{z}$$

  where $p(\mathbf{X} \mid \mathbf{z}, \theta)$ is easily maximized.

EM for Missing Data

- The EM algorithm is highly general — not just for mixture models/clustering
- Can be used any time there is a latent variable whose absence from the data makes likelihood function intractable (e.g., supervised learning w/ missing features; semi-supervised learning with only a few labeled instances)
- General structure: Want

$$\operatorname*{argmax}_{\theta} p(\mathbf{X} \mid \boldsymbol{\theta}) = \operatorname*{argmax}_{\theta} \int_{\mathbf{z}} p(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\theta}) \; d\mathbf{z}$$

  where $p(\mathbf{X} \mid \mathbf{z}, \theta)$ is easily maximized.
- EM applies by iteratively setting

$$\mathbf{Q}^{(m)}(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{X}, \hat{\boldsymbol{\theta}}^{(m-1)})$$

$$\hat{\boldsymbol{\theta}}^{(m)} = \operatorname*{argmax}_{\boldsymbol{\theta}} \mathbf{Q}^{(m)}(\mathbf{z}) \log \left(p(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\theta}) \right)$$

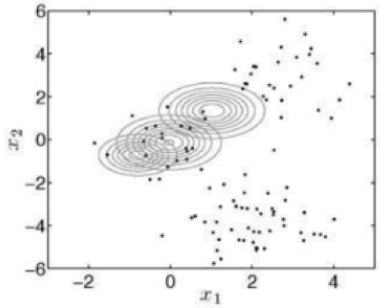
Outline

Outline

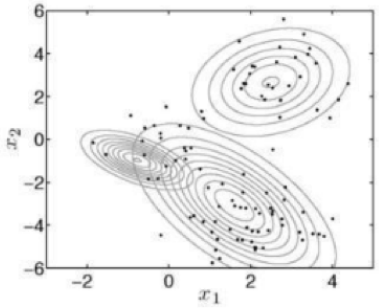
Debugging EM

- Since EM must result in updates to the parameters that monotonically increase the log likelihood, it is advisable to **calculate the log likelihood at each iteration**
- If it ever goes down, there is a bug
- In addition, it can be useful to keep track of the lower bound
- If at the end of the M-step the lower bound is above the likelihood, there is a bug

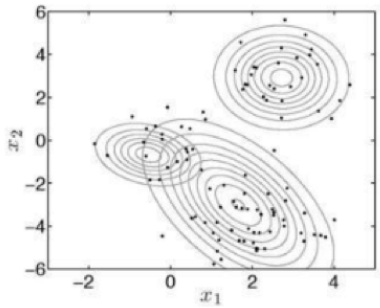
Tracking the Log Likelihood and Bound



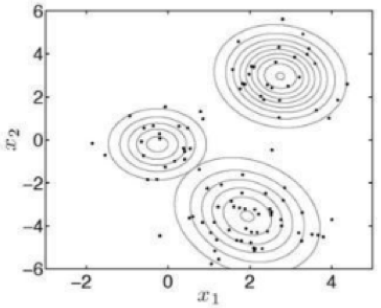
(a) The three randomly initialised Gaussian mixture components



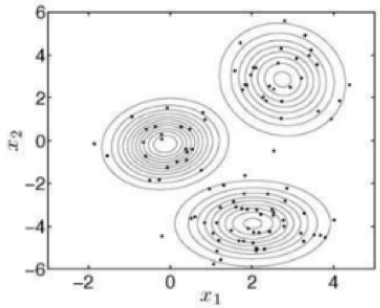
(b) The three components after one iteration of the EM algorithm



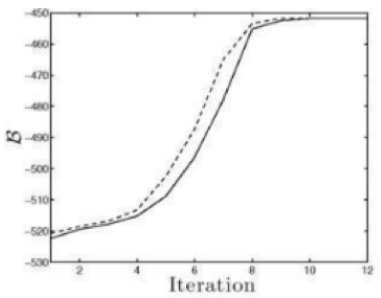
(c) The three components after five iterations of the EM algorithm



(d) The three components after seven iterations of the EM algorithm



(e) The three components at convergence of the EM algorithm



(f) The evolution of the bound $\mathcal{B}$ (solid line, Equation 6.8) and log likelihood L (dashed line, Equation 6.5)

Outline

In Practice: Dealing With Local Maxima

- Just as with K-means, the result depends on the initialization.

In Practice: Dealing With Local Maxima

- Just as with K-means, the result depends on the initialization.
- Just as with K-means, we can try many intializations and pick the one that yields the highest likelihood.

Outline

What are Good Initialization Methods?

- For a mixture of Normals, we can use K-means to initialize the parameters

What are Good Initialization Methods?

- For a mixture of Normals, we can use K-means to initialize the parameters
  - K-means sets $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

What are Good Initialization Methods?

- For a mixture of Normals, we can use K-means to initialize the parameters
  - K-means sets $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
  - We can use the "hard assignments" to set an initial $\mathbf{Q}$, and use that to find initial $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K$ and $\boldsymbol{\pi}$

What are Good Initialization Methods?

- For a mixture of Normals, we can use K-means to initialize the parameters
  - K-means sets $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
  - We can use the "hard assignments" to set an initial $\mathbf{Q}$, and use that to find initial $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K$ and $\boldsymbol{\pi}$
- To initialize K-means itself, it can be helpful to place cluster centers on individual points which are not too close together

What are Good Initialization Methods?

- For a mixture of Normals, we can use K-means to initialize the parameters
  - K-means sets $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
  - We can use the "hard assignments" to set an initial $\mathbf{Q}$, and use that to find initial $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K$ and $\boldsymbol{\pi}$
- To initialize K-means itself, it can be helpful to place cluster centers on individual points which are not too close together
  - For example, choose the first point at random, then choose each successive point in proportion to its distance from the first point

Outline

Degenerate Solutions

- With a Gaussian mixture model fit with MLE, in which the $\boldsymbol{\Sigma}_k$ are learned, what will happen if we end up with just one point in a particular cluster?

Degenerate Solutions

- With a Gaussian mixture model fit with MLE, in which the $\boldsymbol{\Sigma}_k$ are learned, what will happen if we end up with just one point in a particular cluster?
- The multivariate Normal density

$$p(\mathbf{x}_n \mid \boldsymbol{\mu}_{\mathbf{z}_n}, \boldsymbol{\Sigma}_{\mathbf{z}_n}) = (2\pi)^{D/2} |\boldsymbol{\Sigma}_{\mathbf{z}_n}|^{-D/2} \exp\{-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathbf{z}_n})^\top \boldsymbol{\Sigma}_{\mathbf{z}_n}^{-1}(\mathbf{x}_n$$

Degenerate Solutions

- With a Gaussian mixture model fit with MLE, in which the $\boldsymbol{\Sigma}_k$ are learned, what will happen if we end up with just one point in a particular cluster?
- The multivariate Normal density

$$p(\mathbf{x}_n \mid \boldsymbol{\mu}_{\mathbf{z}_n}, \boldsymbol{\Sigma}_{\mathbf{z}_n}) = (2\pi)^{D/2} |\boldsymbol{\Sigma}_{\mathbf{z}_n}|^{-D/2} \exp\{-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathbf{z}_n})^\top \boldsymbol{\Sigma}_{\mathbf{z}_n}^{-1}(\mathbf{x}_n$$

- With just one point in a cluster, $\hat{\boldsymbol{\mu}}_{\mathbf{z}_n}^{(MLE)} = \mathbf{x}_n$, and therefore $\hat{\boldsymbol{\Sigma}}_{\mathbf{z}_n}^{(MLE)}$ will be the zero matrix.

Degenerate Solutions

- With a Gaussian mixture model fit with MLE, in which the $\boldsymbol{\Sigma}_k$ are learned, what will happen if we end up with just one point in a particular cluster?
- The multivariate Normal density

$$p(\mathbf{x}_n \mid \boldsymbol{\mu}_{\mathbf{z}_n}, \boldsymbol{\Sigma}_{\mathbf{z}_n}) = (2\pi)^{D/2} |\boldsymbol{\Sigma}_{\mathbf{z}_n}|^{-D/2} \exp\{-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathbf{z}_n})^\top \boldsymbol{\Sigma}_{\mathbf{z}_n}^{-1}(\mathbf{x}_n$$

- With just one point in a cluster, $\hat{\boldsymbol{\mu}}_{\mathbf{z}_n}^{(MLE)} = \mathbf{x}_n$, and therefore $\hat{\boldsymbol{\Sigma}}_{\mathbf{z}_n}^{(MLE)}$ will be the zero matrix.
- This will make the overall likelihood infinite.

Degenerate Solutions

- With a Gaussian mixture model fit with MLE, in which the $\boldsymbol{\Sigma}_k$ are learned, what will happen if we end up with just one point in a particular cluster?
- The multivariate Normal density

$$p(\mathbf{x}_n \mid \boldsymbol{\mu}_{\mathbf{z}_n}, \boldsymbol{\Sigma}_{\mathbf{z}_n}) = (2\pi)^{D/2} |\boldsymbol{\Sigma}_{\mathbf{z}_n}|^{-D/2} \exp\{-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathbf{z}_n})^\top \boldsymbol{\Sigma}_{\mathbf{z}_n}^{-1}(\mathbf{x}_n$$

- With just one point in a cluster, $\hat{\boldsymbol{\mu}}_{\mathbf{z}_n}^{(MLE)} = \mathbf{x}_n$, and therefore $\hat{\boldsymbol{\Sigma}}_{\mathbf{z}_n}^{(MLE)}$ will be the zero matrix.
- This will make the overall likelihood infinite.
- In fact, this can happen even with multiple points in a cluster, since $\hat{\boldsymbol{\Sigma}}^{(MLE)}$ will have zero determinant if any dimensions are perfectly correlated

Degenerate Solutions

- With a Gaussian mixture model fit with MLE, in which the Σ_k are learned, what will happen if we end up with just one point in a particular cluster?
- The multivariate Normal density

$$p(\mathbf{x}_n \mid \boldsymbol{\mu}_{\mathbf{z}_n}, \boldsymbol{\Sigma}_{\mathbf{z}_n}) = (2\pi)^{D/2} |\boldsymbol{\Sigma}_{\mathbf{z}_n}|^{-D/2} \exp\{-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathbf{z}_n})^\top \boldsymbol{\Sigma}_{\mathbf{z}_n}^{-1}(\mathbf{x}_n$$

- With just one point in a cluster, $\hat{\boldsymbol{\mu}}_{\mathbf{z}_n}^{(MLE)} = \mathbf{x}_n$, and therefore $\hat{\boldsymbol{\Sigma}}_{\mathbf{z}_n}^{(MLE)}$ will be the zero matrix.
- This will make the overall likelihood infinite.
- In fact, this can happen even with multiple points in a cluster, since $\hat{\boldsymbol{\Sigma}}^{(MLE)}$ will have zero determinant if any dimensions are perfectly correlated
- Since EM increases the likelihood, it will push us toward these "degenerate" solutions if it can

Outline

Regularization / Bayesian Mixture Model

- We can avoid degenerate solutions by adding a penalty term in $\boldsymbol{\theta}$ to the objective function:

$$\mathcal{L}^*(\boldsymbol{\pi}, \boldsymbol{\theta}) = \log p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) - g(\boldsymbol{\pi}, \boldsymbol{\theta})$$

  for some g that penalizes covariances with determinants that are too small

Regularization / Bayesian Mixture Model

- We can avoid degenerate solutions by adding a penalty term in $\boldsymbol{\theta}$ to the objective function:

$$\mathcal{L}^*(\boldsymbol{\pi}, \boldsymbol{\theta}) = \log p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) - g(\boldsymbol{\pi}, \boldsymbol{\theta})$$

  for some g that penalizes covariances with determinants that are too small
- If $g(\boldsymbol{\pi}, \boldsymbol{\theta}) = -\log p(\boldsymbol{\pi}, \boldsymbol{\theta})$ for some density function p, this can be interpreted as maximizing the log posterior where $p(\boldsymbol{\pi}, \boldsymbol{\theta}) = \exp(-g(\boldsymbol{\pi}, \boldsymbol{\theta}))$ is the prior

Regularization / Bayesian Mixture Model

- We can avoid degenerate solutions by adding a penalty term in $\boldsymbol{\theta}$ to the objective function:

$$\mathcal{L}^*(\boldsymbol{\pi}, \boldsymbol{\theta}) = \log p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) - g(\boldsymbol{\pi}, \boldsymbol{\theta})$$

  for some g that penalizes covariances with determinants that are too small
- If $g(\boldsymbol{\pi}, \boldsymbol{\theta}) = -\log p(\boldsymbol{\pi}, \boldsymbol{\theta})$ for some density function p, this can be interpreted as maximizing the log posterior where $p(\boldsymbol{\pi}, \boldsymbol{\theta}) = \exp(-g(\boldsymbol{\pi}, \boldsymbol{\theta}))$ is the prior
- A natural choice for priors are conditionally conjugate:

$$\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1} \overset{i.i.d.}{\sim} \text{Wishart}(\boldsymbol{\Lambda}_0, n_0)$$

$$\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \overset{i.i.d.}{\sim} \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

$$\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha})$$

  The Wishart generalizes the Gamma to matrices

Regularization / Bayesian Mixture Model

- We can avoid degenerate solutions by adding a penalty term in $\boldsymbol{\theta}$ to the objective function:

$$\mathcal{L}^*(\boldsymbol{\pi}, \boldsymbol{\theta}) = \log p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) - g(\boldsymbol{\pi}, \boldsymbol{\theta})$$

  for some g that penalizes covariances with determinants that are too small
- If $g(\boldsymbol{\pi}, \boldsymbol{\theta}) = -\log p(\boldsymbol{\pi}, \boldsymbol{\theta})$ for some density function p, this can be interpreted as maximizing the log posterior where $p(\boldsymbol{\pi}, \boldsymbol{\theta}) = \exp(-g(\boldsymbol{\pi}, \boldsymbol{\theta}))$ is the prior
- A natural choice for priors are conditionally conjugate:

$$\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1} \overset{i.i.d.}{\sim} \text{Wishart}(\boldsymbol{\Lambda}_0, n_0)$$

$$\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \overset{i.i.d.}{\sim} \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

$$\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha})$$

  The Wishart generalizes the Gamma to matrices
- **Note:** If we force the $\boldsymbol{\Sigma}_k^{-1}$ to be diagonal, we can instead put independent Gamma priors on each coordinate

EM With Regularized Objective

- The E-step in EM is unchanged by modifying the objective for $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$
- Provided $g(\boldsymbol{\pi}, \boldsymbol{\theta})$ separates as $g_1(\boldsymbol{\pi}) + \sum_{k=1}^{K} g_2(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, the M-step is still straightforward
- Instead of maximizing the log likelihood, set:

$$\hat{\boldsymbol{\pi}}^{(m)} \mid \mathbf{Q} = \underset{\boldsymbol{\pi}}{\operatorname{argmax}} \left(-g_1(\boldsymbol{\pi}) + \sum_{k=1}^{K} \sum_{n=1}^{N} q_{nk} \log(\pi_k) \right)$$

$$\text{subject to } \sum_{k=1}^{K} \pi_k = 1$$

$$\hat{\boldsymbol{\mu}}_k^{(m)}, \hat{\boldsymbol{\Sigma}}_k^{(m)} \mid \mathbf{Q} = \underset{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}{\operatorname{argmax}} \Big(- g_2(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \sum_{n=1}^{N} q_{nk} \log \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Big)$$

- If g is chosen to be "nice" (e.g., based on a conjugate prior), this will be analytically tractable

Outline

Evaluating a Clustering Result

- Recall that K-means optimized

$$L(\mathbf{z}) = \sum_{n=1}^{N} \left\| \mathbf{x}_n - \boldsymbol{\mu}_{z_n} \right\|^2$$

Evaluating a Clustering Result

- Recall that K-means optimized

$$L(\mathbf{z}) = \sum_{n=1}^{N} \left\| \mathbf{x}_n - \boldsymbol{\mu}_{z_n} \right\|^2$$

- This is not a useful metric for choosing K, since it decreases with more clusters, **even on held-out data**.

Evaluating a Clustering Result

- Recall that K-means optimized

$$L(\mathbf{z}) = \sum_{n=1}^{N} \left\| \mathbf{x}_n - \boldsymbol{\mu}_{z_n} \right\|^2$$

- This is not a useful metric for choosing K, since it decreases with more clusters, **even on held-out data**.
- A mixture model on the other hand, yields a natural metric: the **log likelihood** (perhaps modified with a regularization term):

$$\log p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k p(\mathbf{x}_n \mid \boldsymbol{\theta}_k) \right)$$

Evaluating a Clustering Result

- Recall that K-means optimized

$$L(\mathbf{z}) = \sum_{n=1}^{N} \left\| \mathbf{x}_n - \boldsymbol{\mu}_{z_n} \right\|^2$$

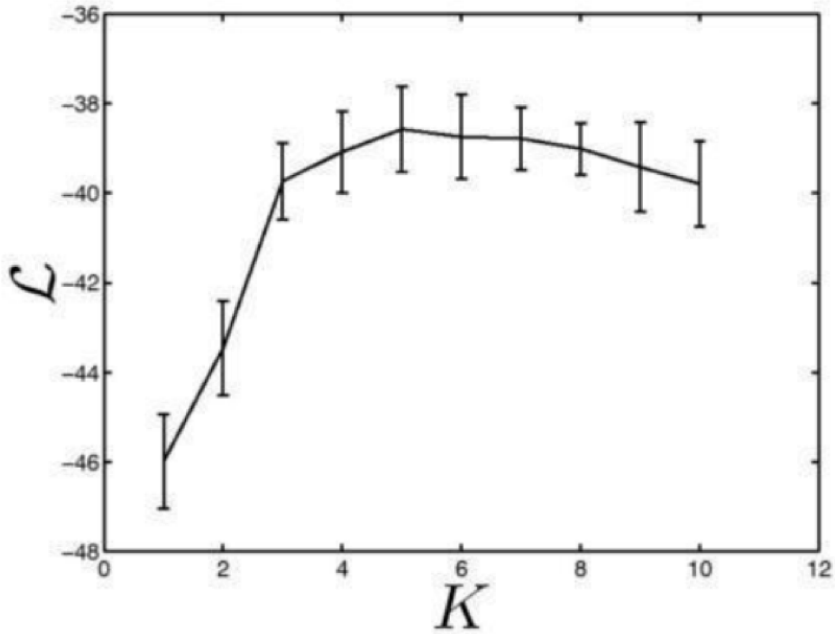
- This is not a useful metric for choosing K, since it decreases with more clusters, **even on held-out data**.
- A mixture model on the other hand, yields a natural metric: the **log likelihood** (perhaps modified with a regularization term):

$$\log p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k p(\mathbf{x}_n \mid \boldsymbol{\theta}_k) \right)$$

- And since the result is a probability model for new data, this can be evaluated on a validation/test set:

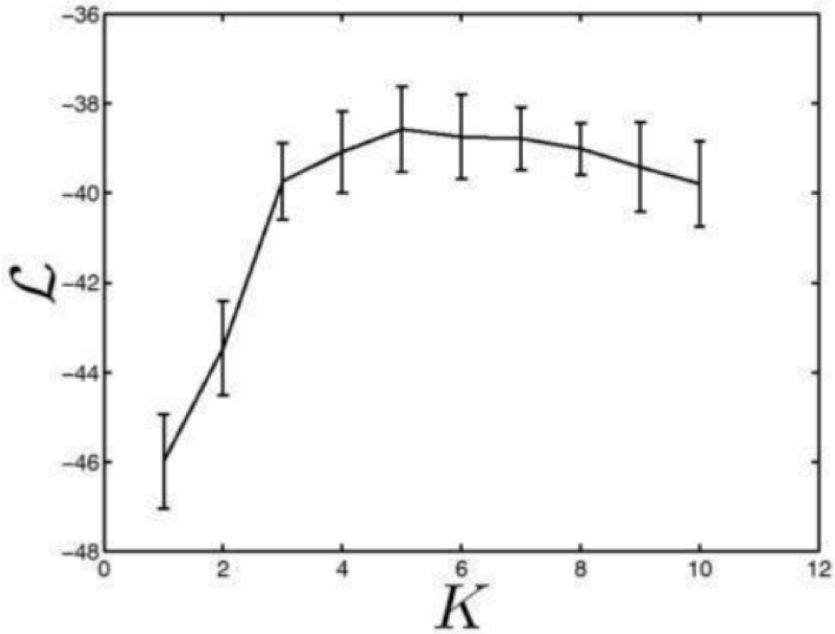
$$\log p(\mathbf{X}_{new} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{n=1}^{N_{new}} \log \left(\sum_{k=1}^{K} \pi_k p(\mathbf{x}_{new,n} \mid \boldsymbol{\theta}_k) \right)$$

Choosing K with Cross-Validation with the Log Likelihood



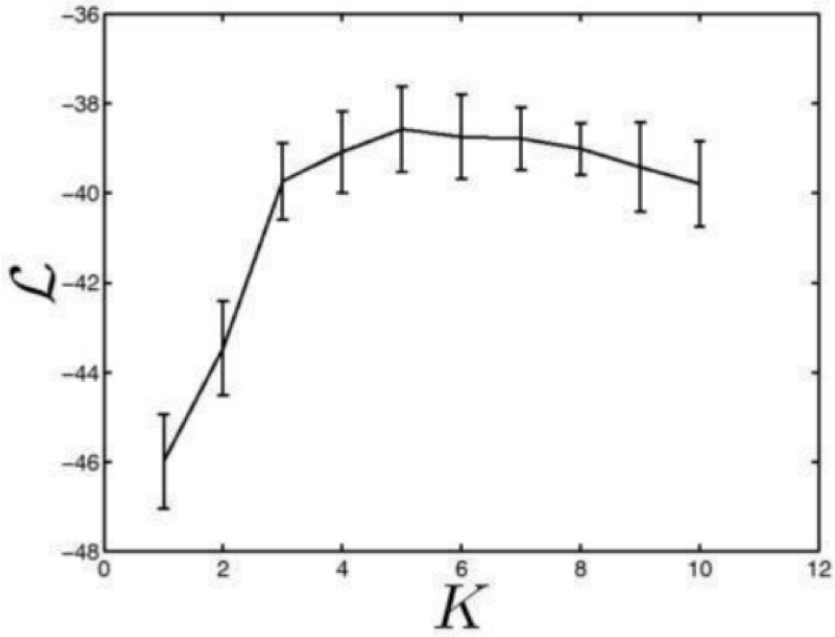
- How does this protect against too many clusters?

Choosing K with Cross-Validation with the Log Likelihood



- How does this protect against too many clusters?
- **Intuition:** By marginalizing over $\mathbf{z}$, we are evaluating the fit of the *entire mixture* to the data, not just the single best cluster

Choosing K with Cross-Validation with the Log Likelihood



- How does this protect against too many clusters?
- **Intuition:** By marginalizing over $\mathbf{z}$, we are evaluating the fit of the *entire mixture* to the data, not just the single best cluster
- If a "true" cluster is subdivided and the subcluster variances are too tight, the density may be very low in

Outline

Regularized Estimation vs. Bayesian Inference

General form of a mixture model:

$$p(\mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{y} \mid \boldsymbol{\theta}_k), \qquad \sum_{k=1}^{K} \pi_k = 1$$

Regularized Estimation vs. Bayesian Inference

General form of a mixture model:

$$p(\mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{y} \mid \boldsymbol{\theta}_k), \qquad \sum_{k=1}^{K} \pi_k = 1$$

Three approaches to inference in mixture models:

1. **Max Likelihood:** Find

$$\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\pi}, \boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

2. **Regularized/MAP:** Find

$$\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\pi}, \boldsymbol{\theta}}{\operatorname{argmax}}\, \mathcal{L}^*(\boldsymbol{\pi}, \boldsymbol{\theta}) = \underset{\boldsymbol{\pi}, \boldsymbol{\theta}}{\operatorname{argmax}} \{p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) - g(\boldsymbol{\pi}, \boldsymbol{\theta})\}$$

3. **Fully Bayesian:** Put a prior, $p(\boldsymbol{\pi}, \boldsymbol{\theta})$ on $\boldsymbol{\pi}, \boldsymbol{\theta}$ and find

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) = \frac{p(\boldsymbol{\pi}, \boldsymbol{\theta}) p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})}{\int p(\boldsymbol{\pi}', \boldsymbol{\theta}') p(\mathbf{Y} \mid \boldsymbol{\pi}', \boldsymbol{\theta}')\, d\boldsymbol{\theta}' d\boldsymbol{\pi}'}$$

Regularized Estimation vs. Bayesian Inference

- Since the posterior is

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) = k_{\mathbf{X}} p(\boldsymbol{\pi}, \boldsymbol{\theta}) p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

  the log posterior is

$$\log(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) = \log(k_{\mathbf{X}}) + \log(p(\boldsymbol{\pi}, \boldsymbol{\theta})) + \log(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

Regularized Estimation vs. Bayesian Inference

- Since the posterior is

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) = k_{\mathbf{X}} p(\boldsymbol{\pi}, \boldsymbol{\theta}) p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

  the log posterior is

$$\log(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) = \log(k_{\mathbf{X}}) + \log(p(\boldsymbol{\pi}, \boldsymbol{\theta})) + \log(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

- Hence choosing $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ to maximize the posterior is equivalent to maximizing a regularized likelihood function

$$\mathcal{L}^*(\boldsymbol{\pi}, \boldsymbol{\theta}) = \log L(\boldsymbol{\pi}, \boldsymbol{\theta}) - g(\boldsymbol{\pi}, \boldsymbol{\theta})$$

  where the "penalty" $g(\boldsymbol{\pi}, \boldsymbol{\theta})$ is

$$g(\boldsymbol{\pi}, \boldsymbol{\theta}) = -\log(p(\boldsymbol{\pi}, \boldsymbol{\theta}))$$

Regularized Estimation vs. Bayesian Inference

- Since the posterior is

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) = k_{\mathbf{X}} p(\boldsymbol{\pi}, \boldsymbol{\theta}) p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

  the log posterior is

$$\log(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) = \log(k_{\mathbf{X}}) + \log(p(\boldsymbol{\pi}, \boldsymbol{\theta})) + \log(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

- Hence choosing $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ to maximize the posterior is equivalent to maximizing a regularized likelihood function

$$\mathcal{L}^*(\boldsymbol{\pi}, \boldsymbol{\theta}) = \log L(\boldsymbol{\pi}, \boldsymbol{\theta}) - g(\boldsymbol{\pi}, \boldsymbol{\theta})$$

  where the "penalty" $g(\boldsymbol{\pi}, \boldsymbol{\theta})$ is

$$g(\boldsymbol{\pi}, \boldsymbol{\theta}) = -\log(p(\boldsymbol{\pi}, \boldsymbol{\theta}))$$

- So is there any practical difference between Bayesian inference and regularized estimation?

Regularized Estimation vs. Bayesian Inference

- Since the posterior is

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) = k_{\mathbf{X}} p(\boldsymbol{\pi}, \boldsymbol{\theta}) p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

  the log posterior is

$$\log(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) = \log(k_{\mathbf{X}}) + \log(p(\boldsymbol{\pi}, \boldsymbol{\theta})) + \log(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

- Hence choosing $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ to maximize the posterior is equivalent to maximizing a regularized likelihood function

$$\mathcal{L}^*(\boldsymbol{\pi}, \boldsymbol{\theta}) = \log L(\boldsymbol{\pi}, \boldsymbol{\theta}) - g(\boldsymbol{\pi}, \boldsymbol{\theta})$$

  where the "penalty" $g(\boldsymbol{\pi}, \boldsymbol{\theta})$ is

$$g(\boldsymbol{\pi}, \boldsymbol{\theta}) = -\log(p(\boldsymbol{\pi}, \boldsymbol{\theta}))$$

- So is there any practical difference between Bayesian inference and regularized estimation?
- **Yes**: Bayesian inference yields a **distribution** over parameters, not a point estimate

Regularized Estimation vs. Bayesian Inference

- Why does point estimate vs. distribution matter?

Regularized Estimation vs. Bayesian Inference

- Why does point estimate vs. distribution matter?
- **Supervised setting:** avoid **overconfident conclusions** by taking weighted averages of predictions over all possible parameters (posterior predictive distribution)

Regularized Estimation vs. Bayesian Inference

- Why does point estimate vs. distribution matter?
- **Supervised setting:** avoid **overconfident conclusions** by taking weighted averages of predictions over all possible parameters (posterior predictive distribution)
- **More generally:** Make better decisions by taking more information into account

Regularized Estimation vs. Bayesian Inference

- Why does point estimate vs. distribution matter?
- **Supervised setting:** avoid **overconfident conclusions** by taking weighted averages of predictions over all possible parameters (posterior predictive distribution)
- **More generally:** Make better decisions by taking more information into account
- **Clustering:** Can answer questions like, "What is the marginal probability that two points are in the same cluster"?

Outline

Intractable Posterior

- It is easy to write down the unnormalized posterior density:

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) \propto p(\boldsymbol{\pi}, \boldsymbol{\theta}) p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

Intractable Posterior

- It is easy to write down the unnormalized posterior density:

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) \propto p(\boldsymbol{\pi}, \boldsymbol{\theta}) p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

- Mixture of Normals case:

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \,|\, \mathbf{X}) \propto p(\boldsymbol{\pi}) \prod_{k=1}^{K} p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}) \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

  where, e.g., $p(\boldsymbol{\pi})$ is Dirichlet, $p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1})$ is Normal-Wishart

Intractable Posterior

- It is easy to write down the unnormalized posterior density:

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) \propto p(\boldsymbol{\pi}, \boldsymbol{\theta}) p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

- Mixture of Normals case:

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \,|\, \mathbf{X}) \propto p(\boldsymbol{\pi}) \prod_{k=1}^{K} p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}) \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

  where, e.g., $p(\boldsymbol{\pi})$ is Dirichlet, $p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1})$ is Normal-Wishart

- But hard to do much with it (except maybe find a local maximum)

Intractable Posterior

- It is easy to write down the unnormalized posterior density:

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{X}) \propto p(\boldsymbol{\pi}, \boldsymbol{\theta}) p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$$

- Mixture of Normals case:

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} \,|\, \mathbf{X}) \propto p(\boldsymbol{\pi}) \prod_{k=1}^{K} p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}) \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

  where, e.g., $p(\boldsymbol{\pi})$ is Dirichlet, $p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1})$ is Normal-Wishart

- But hard to do much with it (except maybe find a local maximum)
- Solution: Try to draw **samples** from it.

Outline

Gibbs Sampling as "Stochastic EM"

- The EM algorithm involves, iteratively

Gibbs Sampling as "Stochastic EM"

- The EM algorithm involves, iteratively
  1. Computing an expectation over cluster assignments, $\mathbf{z}$ (using the conditional posterior, $p(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\theta})$)

Gibbs Sampling as "Stochastic EM"

- The EM algorithm involves, iteratively
  1. Computing an expectation over cluster assignments, $\mathbf{z}$ (using the conditional posterior, $p(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\theta})$)
  2. Maximizing the "quantum" posterior $p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{Q}, \mathbf{X})$

Gibbs Sampling as "Stochastic EM"

- The EM algorithm involves, iteratively
  1. Computing an expectation over cluster assignments, $\mathbf{z}$ (using the conditional posterior, $p(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\theta})$)
  2. Maximizing the "quantum" posterior $p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{Q}, \mathbf{X})$
- Gibbs sampling (in this context) involves, iteratively

Gibbs Sampling as "Stochastic EM"

- The EM algorithm involves, iteratively
  1. Computing an expectation over cluster assignments, $\mathbf{z}$ (using the conditional posterior, $p(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\theta})$)
  2. Maximizing the "quantum" posterior $p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{Q}, \mathbf{X})$
- Gibbs sampling (in this context) involves, iteratively
  1. Sampling cluster assignments, $\mathbf{z}$ from the conditional posterior $p(\mathbf{z} \mid \boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{X})$

Gibbs Sampling as "Stochastic EM"

- The EM algorithm involves, iteratively
  1. Computing an expectation over cluster assignments, $\mathbf{z}$ (using the conditional posterior, $p(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\theta})$)
  2. Maximizing the "quantum" posterior $p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{Q}, \mathbf{X})$
- Gibbs sampling (in this context) involves, iteratively
  1. Sampling cluster assignments, $\mathbf{z}$ from the conditional posterior $p(\mathbf{z} \mid \boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{X})$
  2. Sampling parameter values from the **conditional** posterior $p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{z}, \mathbf{X})$

What does Gibbs Yield?

- Over many iterations, Gibbs sampling yields a collection of many cluster assignments and many parameter values, distributed according to the joint posterior

$$\left\{\mathbf{z}^{(m)}, \boldsymbol{\pi}^{(m)}, \left\{\boldsymbol{\mu}_k^{(m)}, \boldsymbol{\Sigma}_k^{(m)}\right\}_{k=1}^{K}\right\}_{m=1}^{M}$$

What does Gibbs Yield?

- Over many iterations, Gibbs sampling yields a collection of many cluster assignments and many parameter values, distributed according to the joint posterior

$$\left\{\mathbf{z}^{(m)}, \boldsymbol{\pi}^{(m)}, \left\{\boldsymbol{\mu}_k^{(m)}, \boldsymbol{\Sigma}_k^{(m)}\right\}_{k=1}^{K}\right\}_{m=1}^{M}$$

- These can be used to approximate more or less any function of the assignments and parameters we want

What does Gibbs Yield?

- Over many iterations, Gibbs sampling yields a collection of many cluster assignments and many parameter values, distributed according to the joint posterior

$$\{\mathbf{z}^{(m)}, \boldsymbol{\pi}^{(m)}, \{\boldsymbol{\mu}_k^{(m)}, \boldsymbol{\Sigma}_k^{(m)}\}_{k=1}^K\}_{m=1}^M$$

- These can be used to approximate more or less any function of the assignments and parameters we want
- Example:

$$p(z_n = z_{n'} \mid \mathbf{X}) \approx \frac{1}{M} \sum_{m=1}^{M} I(z_n^{(s)} = z_{n'}^{(s)})$$

What does Gibbs Yield?

- Over many iterations, Gibbs sampling yields a collection of many cluster assignments and many parameter values, distributed according to the joint posterior

$$\{\mathbf{z}^{(m)}, \boldsymbol{\pi}^{(m)}, \{\boldsymbol{\mu}_k^{(m)}, \boldsymbol{\Sigma}_k^{(m)}\}_{k=1}^K\}_{m=1}^M$$

- These can be used to approximate more or less any function of the assignments and parameters we want
- Example:

$$p(z_n = z_{n'} \mid \mathbf{X}) \approx \frac{1}{M} \sum_{m=1}^{M} I(z_n^{(s)} = z_{n'}^{(s)})$$

- Or, for future data:

$$p(z_{new} = z_n \mid \mathbf{x}_{new}, \mathbf{X}) \approx \frac{1}{M} \sum_{m=1}^{M} \frac{\pi_{z_n^{(m)}}^{(m)} \mathcal{N}(\mathbf{x}_{new} \mid \boldsymbol{\mu}_{z_n^{(m)}}, \boldsymbol{\Sigma}_{z_n^{(m)}})}{\sum_{k=1}^K \pi_k^{(m)} \mathcal{N}(\mathbf{x}_{new} \mid \boldsymbol{\mu}_k^{(m)}, \boldsymbol{\Sigma}_k^{(m)})}$$

Outline

Concretely: The Algorithm

- In general, Gibbs sampling applies when we have multiple sets of unknowns, but where it is difficult to sample all sets simultaneously.

Concretely: The Algorithm

- In general, Gibbs sampling applies when we have multiple sets of unknowns, but where it is difficult to sample all sets simultaneously.
- Here, unknowns are $\mathbf{z}$, $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$

Concretely: The Algorithm

- In general, Gibbs sampling applies when we have multiple sets of unknowns, but where it is difficult to sample all sets simultaneously.
- Here, unknowns are $\mathbf{z}$, $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$
- Idea: **partition the unknowns into "blocks"**, and iteratively **sample each block from its posterior conditioned on the current values of the others**

Concretely: The Algorithm

- In general, Gibbs sampling applies when we have multiple sets of unknowns, but where it is difficult to sample all sets simultaneously.
- Here, unknowns are $\mathbf{z}$, $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$
- Idea: **partition the unknowns into "blocks"**, and iteratively **sample each block from its posterior conditioned on the current values of the others**
- Here, we need

$$1\text{: } p(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\pi})$$
$$2\text{: } p(\boldsymbol{\theta}, \boldsymbol{\pi} \mid \mathbf{z}, \mathbf{X})$$

Finding the conditional posterior for $\mathbf{z}$

- Recall, in EM, we computed

$$q_{nk} = p(z_n = k \mid \boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{Y}) = \frac{\pi_k \mathcal{N}(\mathbf{y}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \pi_k \mathcal{N}(\mathbf{y}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

Finding the conditional posterior for $\mathbf{z}$

- Recall, in EM, we computed

$$q_{nk} = p(z_n = k \mid \boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{Y}) = \frac{\pi_k \mathcal{N}(\mathbf{y}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \pi_k \mathcal{N}(\mathbf{y}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

- For Gibbs sampling, compute the q_{nk}s conditioned on the current $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ and sample

$$z_n \mid \boldsymbol{\pi}, \boldsymbol{\theta} \sim \text{Categorical}(q_{n1}, \ldots, q_{nK})$$

Finding the conditional posterior for $\boldsymbol{\theta}$

- Having fixed the z_ns, the conditional posterior for $\boldsymbol{\pi}, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ separates into a factor for $\boldsymbol{\pi}$ and one factor for each $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ pair

Finding the conditional posterior for $\boldsymbol{\theta}$

- Having fixed the z_ns, the conditional posterior for $\boldsymbol{\pi}, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ separates into a factor for $\boldsymbol{\pi}$ and one factor for each $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ pair
- We get, for each k

$$p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \mid \mathbf{z}, \mathbf{X}) \propto p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \prod_{n:z_n=k} p(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Finding the conditional posterior for $\boldsymbol{\theta}$

- Having fixed the z_ns, the conditional posterior for $\boldsymbol{\pi}, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ separates into a factor for $\boldsymbol{\pi}$ and one factor for each $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ pair
- We get, for each k

$$p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \mid \mathbf{z}, \mathbf{X}) \propto p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \prod_{n:z_n=k} p(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- This is often just a conjugate update of parameters from prior to posterior)

Finding the conditional posterior for $\boldsymbol{\theta}$

- Having fixed the z_ns, the conditional posterior for $\boldsymbol{\pi}, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ separates into a factor for $\boldsymbol{\pi}$ and one factor for each $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ pair
- We get, for each k

$$p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \mid \mathbf{z}, \mathbf{X}) \propto p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \prod_{n:z_n=k} p(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- This is often just a conjugate update of parameters from prior to posterior)
- Same idea for $\boldsymbol{\pi}$ (which is conditionally independent of $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ given $\mathbf{z}$)

$$p(\boldsymbol{\pi} \mid \mathbf{z}, \mathbf{X}) \propto p(\boldsymbol{\pi}) \prod_k \prod_{n:z_n=k} \pi_k$$

  This is often just a Dirichlet distribution where the $\mathbf{z}$s update the prior parameters

Summary: Gibbs Algorithm for GMM

1. Initialize cluster assignments, $\mathbf{z}^{(0)}$ (e.g., using K-means)
2. For $m = 1, \ldots, M$ (for M chosen), sample

   (a) $\boldsymbol{\mu}_k^{(m)}, \boldsymbol{\Sigma}_k^{(m)} \sim p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \mid \mathbf{z}^{(m-1)}, \mathbf{X}), k = 1, \ldots, K$

   (b) $\boldsymbol{\pi}^{(m)} \sim p(\boldsymbol{\pi} \mid \mathbf{z}^{(m-1)})$

   (c) $z_n^{(m)} \sim p(z_n \mid \boldsymbol{\pi}^{(m)}, \boldsymbol{\theta}^{(m)}, \mathbf{x}_n), n = 1, \ldots, N$

3. Use samples from all $m > M_{burnin}$ to approximate expected values of interest (where $M_{burnin} > 1$ is chosen to reduce the influence of initialization (analogy: letting an engine "burn in" before use)