Machine Learning for Data Science (CS4786)
Lecture 8

Mixture Models, Dimensionality Reduction

Course Webpage:
http://www.cs.cornell.edu/Courses/cs4786/2017fa/
Towards Hard Gaussian Mixture Model

For all $j \in [K]$, initialize cluster centroids $\hat{r}_j^0$, ellipsoids $\hat{\Sigma}_j^0$ and initial proportions $\pi_j^0$ randomly and set $m = 1$

Repeat until convergence (or until patience runs out)

4. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}_m(x_t) = \arg\min_{j \in [K]} (x_t - \hat{r}_{j}^{m-1})^\top (\hat{\Sigma}_j^{m-1})^{-1} (x_t - \hat{r}_{j}^{m-1}) - \log(\pi_j^{m-1})$$

2. For each $j \in [K]$, set new representative as

$$\hat{r}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{x_t \in \hat{C}_j^m} x_t \quad \hat{\Sigma}_j^m = \frac{1}{|C_j|} \sum_{t \in C_j} (x_t - \hat{r}_j^m)(x_t - \hat{r}_j^m)^\top \quad \pi_j^m = \frac{|C_j^m|}{n}$$

3. $m \leftarrow m + 1$
For all \( j \in [K] \), initialize cluster centroids \( \hat{r}_j^0 \), ellipsoids \( \hat{\Sigma}_j^0 \) and initial proportions \( \pi_j^0 \) randomly and set \( m = 1 \).

Repeat until convergence (or until patience runs out):

1. For each \( t \in \{1, \ldots, n\} \), set cluster identity of the point

\[
\hat{c}^m(x_t) = \arg\min_{j \in [K]} (x_t - \hat{r}_j^{m-1})^\top (\hat{\Sigma}_j^{m-1})^{-1} (x_t - \hat{r}_j^{m-1}) - \log(\pi_j^{m-1})
\]

2. For each \( j \in [K] \), set new representative as

\[
\hat{r}_j^m = \frac{1}{|C_j^m|} \sum_{x_t \in \hat{C}_j^m} x_t
\]
\[
\hat{\Sigma}_j^m = \frac{1}{|C_j|} \sum_{t \in C_j} (x_t - \hat{r}_j^m)(x_t - \hat{r}_j^m)^\top
\]
\[
\pi_j^m = \frac{|C_j^m|}{n}
\]

3. \( m \leftarrow m + 1 \)
Towards Hard Gaussian Mixture Model

- For all $j \in [K]$, initialize cluster centroids $\hat{r}_j^0$, ellipsoids $\hat{\Sigma}_j^0$ and initial proportions $\pi_j^0$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point
     
     $\hat{c}^m(x_t) = \arg\min_{j \in [K]} \left( x_t - \hat{r}_j^{m-1} \right)^\top \left( \hat{\Sigma}_j^{m-1} \right)^{-1} \left( x_t - \hat{r}_j^{m-1} \right) - \log(\pi_j^{m-1})$

     
     $d(x_t, C_j)$

  2. For each $j \in [K]$, set new representative as

     $\hat{r}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{x_t \in \hat{C}_j^m} x_t$

     $\hat{\Sigma}_j^m = \frac{1}{|C_j|} \sum_{t \in C_j} (x_t - \hat{r}_j^m)(x_t - \hat{r}_j^m)^\top$

     $\pi_j^m = \frac{|C_j^m|}{n}$

  3. $m \leftarrow m + 1$
For all \( j \in [K] \), initialize \( \pi^0 \) and parameters \( \theta_1, \ldots, \theta_K \) randomly and set \( m = 1 \)

Repeat until convergence (or until patience runs out)

1. For each \( t \in \{1, \ldots, n\} \), set cluster identity of the point

\[
\hat{c}^m(x_t) = \arg\min_{j \in [K]} d(x_t, \theta_j) - \log(\pi^m_j)
\]

2. For each \( j \in [K] \), set new representative as

\[
\text{compute } \theta_j \text{ for cluster } C_j \quad \& \quad \pi^m_j = \frac{|C^m_j|}{n}
\]

3. \( m \leftarrow m + 1 \)
Multivariate Gaussian

- Two parameters:
  - Mean $\mu \in \mathbb{R}^d$
  - Covariance matrix $\Sigma$ of size $d \times d$

$$p(x; \mu, \Sigma) = (2\pi)^{d/2} \det(\Sigma)^{-1/2} \exp \left( -\frac{1}{2} (x - \mu)^\top \Sigma^{-1} (x - \mu) \right)$$
Multivariate Gaussian

- Two parameters:
  - Mean $\mu \in \mathbb{R}^d$
  - Covariance matrix $\Sigma$ of size $d \times d$

\[
p(x; \mu, \Sigma) = (2\pi)^{d/2} \det(\Sigma)^{-1/2} \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right)
\]
HARD GAUSSIAN MIXTURE MODEL

- For all $j \in [K]$, initialize cluster centroids $\hat{r}_j^0$, ellipsoids $\hat{\Sigma}_j^0$ and initial proportions $\pi^0$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}^m(x_t) = \arg\max_{j \in [K]} p(x_t, \hat{r}_{m-1}^j, \hat{\Sigma}_{m-1}^j) \times \pi^m(j)$$

  2. For each $j \in [K]$, set new representative as

$$\hat{r}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{x_t \in \hat{C}_j^m} x_t \quad \hat{\Sigma}_j^m = \frac{1}{|C_j|} \sum_{t \in C_j} (x_t - \hat{r}_j^m)(x_t - \hat{r}_j^m)^\top \quad \pi_j^m = \frac{|C_j^m|}{n}$$

  3. $m \leftarrow m + 1$
For all $j \in [K]$, initialize $\pi^0$ and parameters $\theta_1, \ldots, \theta_K$ randomly and set $m = 1$

Repeat until convergence (or until patience runs out)

1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

   $$\hat{c}^m(x_t) = \arg\max_{j \in [K]} p(x_t, \theta_j) \times \pi_j^{m-1}$$

2. For each $j \in [K]$, set new representative as

   $$\text{compute } \theta_j \text{ for cluster } C_j \quad \& \quad \pi_j^m = \frac{|C_j^m|}{n}$$

3. $m \leftarrow m + 1$
Demo
Pitfall of Hard Assignment
(Soft) Gaussian Mixture Model

- For all $j \in [K]$, initialize cluster centroids $\hat{r}_j^0$ and ellipsoids $\hat{\Sigma}_j^0$ randomly and set $m = 1$

- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

     $$Q_t^m(j) \propto p(x_t, \hat{r}_j^{m-1}, \hat{\Sigma}_j^{m-1}) \times \pi^{m-1}(j)$$

  2. For each $j \in [K]$, set new representative as

     $$\hat{r}_j^m = \frac{\sum_{t=1}^{n} Q_t(j)x_t}{\sum_{t=1}^{n} Q_t(j)}$$

     $$\hat{\Sigma}_j^m = \frac{\sum_{t=1}^{n} Q_t(j)(x_t - \hat{r}_j^m)(x_t - \hat{r}_j^m)^\top}{\sum_{t=1}^{n} Q_t(j)}$$

     $$\pi_j^m = \frac{\sum_{t=1}^{n} Q_t(j)}{n}$$

  3. $m \leftarrow m + 1$
How to choose K

- Elbow method:
  - plot Objective versus K, typically it monotonically decreases.
  - Pick point where there is a kink (explanation in variance is not as much)
  - Intuition: look at rate of change
- Add to objective penalty + \( p(K) \) and minimize, where \( p \) increases with \( K \)
  - intuition we prefer smaller clusters
  - Use prior knowledge to pick \( p \)
  - (AIC, BIC etc can been seen to be specific cases)
Given feature vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, compress the data points into low dimensional representation $y_1, \ldots, y_n \in \mathbb{R}^K$ where $K \ll d$. 
Given feature vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, compress the data points into a low-dimensional representation $y_1, \ldots, y_n \in \mathbb{R}^K$ where $K \ll d$. 

$n \times d$
Given feature vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, compress the data points into low dimensional representation $y_1, \ldots, y_n \in \mathbb{R}^K$ where $K \ll d$. 
Given feature vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, compress the data points into low dimensional representation $y_1, \ldots, y_n \in \mathbb{R}^K$ where $K \ll d$. 

$$X \xrightarrow{\text{Dimensionality Reduction}} Y$$
You are provided with \( n \) data points each in \( \mathbb{R}^d \)

Goal: Compress data into \( n \), points in \( \mathbb{R}^K \) where \( K << d \)

- Retain as much information about the original data set
- Retain desired properties of the original data set
WHY DIMENSIONALITY REDUCTION?

- For computational ease
  - As input to supervised learning algorithm
  - Before clustering to remove redundant information and noise
- Data compression & Noise reduction
- Data visualization
Given feature vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, compress the data points into low dimensional representation $y_1, \ldots, y_n \in \mathbb{R}^K$ where $K \ll d$
Desired properties:

1. Original data can be (approximately) reconstructed

2. Preserve distances between data points

3. “Relevant” information is preserved

4. Noise is reduced
**Dim Reduction: Linear Transformation**

- Pick a low dimensional subspace
- Project linearly to this subspace
- Subspace retains as much information
Dim Reduction: Linear Transformation

Pick a low dimensional subspace
Project linearly to this subspace
Subspace retains as much information
Dim Reduction: Linear Transformation

Pick a low dimensional subspace
Project linearly to this subspace
Subspace retains as much information

\[ \mathbf{X}^{d \times n} \rightarrow \mathbf{X}^{d \times \mathbf{n}} \]
Dim Reduction: Linear Transformation

- Pick a low dimensional subspace
- Project linearly to this subspace
- Subspace retains as much information

\[ X \] 

\[ \mathbf{x}_1 \]

\[ \mathbf{x}_{\mathbf{n}}^\top \]

\[ d \]

\[ W \]

\[ K \]
Dim Reduction: Linear Transformation

Pick a low dimensional subspace

Project linearly to this subspace

Subspace retains as much information

\[ X \times d W = n Y \]
Dim Reduction: Linear Transformation

Pick a low dimensional subspace
Project linearly to this subspace
Subspace retains as much information

\[ \begin{align*}
  \mathbf{X} & \quad \mathbf{d} \quad \mathbf{W} \quad \mathbf{Y} \\
  n & \quad d & \quad K \\
  x_1^T & \quad x_n^T & \quad W \\
  y_1^T & \quad y_n^T & \quad K \\
  y_i^T & = x_i^T W
\end{align*} \]
Example:
Students in classroom
Example:
Students in classroom
PCA: Variance Maximization

Pick directions along which data varies the most

First principal component: $$w_1 = \arg \max_{w} \| w \|^2_2 = \frac{1}{n} \sum_{t=1}^{n} (x_t - \mu) \cdot (x_t - \mu)^\top$$

Writing down Lagrangian and optimizing, $$W \cdot \Sigma \cdot W = \lambda I$$
PCA: Variance Maximization

Pick directions along which data varies the most

First principal component:

\[ w_1 = \arg \max_{w : \|w\|_2 = 1} \sum_{t=1}^{n} w \cdot (x_t - \mu)^2 \]

Writing down Lagrangian and optimizing,

\[ w = \frac{w_1}{\|w_1\|} \]
PCA: Variance Maximization

Pick directions along which data varies the most

First principal component:

\[ w_1 = \arg\max_{w: \|w\|^2 = 1} \langle w, \sum_{t=1}^n (x_t - \mu) \rangle \]

Writing down Lagrangian and optimizing,

\[ w_{max} = \frac{1}{n} \sum_{t=1}^n (x_t - \mu) \]

is the covariance matrix
Prelude: reducing to 1 dimension
Prelude: reducing to 1 dimension
Dim Reduction: Linear Transformation

Prelude: reducing to 1 dimension

\[ \mathbf{w} \]

\[ \mathbf{x}_1 \]
\[ \mathbf{x}_2 \]
\[ \mathbf{x}_3 \]
\[ \mathbf{x}_4 \]
Prelude: reducing to 1 dimension
**Dim Reduction: Linear Transformation**

Prelude: reducing to 1 dimension

\[ y_1 = w^\top x_1 = \|x_1\| \cos(\angle wx_1) \]
PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

\[ w_1 = \arg \max_{w: \|w\|_2^2 = 1} \frac{1}{n} \sum_{t=1}^{n} \left( w^\top x_t - \frac{1}{n} \sum_{t=1}^{n} w^\top x_t \right)^2 \]
PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

\[
\mathbf{w}_1 = \arg \max_{\mathbf{w}: \|\mathbf{w}\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top \mathbf{x}_t - \frac{1}{n} \sum_{t=1}^{n} \mathbf{w}^\top \mathbf{x}_t \right)^2
\]

\[
= \arg \max_{\mathbf{w}: \|\mathbf{w}\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top (\mathbf{x}_t - \mu) \right)^2
\]
PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

\[
\mathbf{w}_1 = \arg \max_{\mathbf{w} : \|\mathbf{w}\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top \mathbf{x}_t - \frac{1}{n} \sum_{t=1}^{n} \mathbf{w}^\top \mathbf{x}_t \right)^2
\]

\[
= \arg \max_{\mathbf{w} : \|\mathbf{w}\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} (\mathbf{w}^\top (\mathbf{x}_t - \mu))^2
\]

\[
= \arg \max_{\mathbf{w} : \|\mathbf{w}\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} \mathbf{w}^\top (\mathbf{x}_t - \mu)(\mathbf{x}_t - \mu)^\top \mathbf{w}
\]
PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

\[
\mathbf{w}_1 = \arg \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top \mathbf{x}_t - \frac{1}{n} \sum_{t=1}^{n} \mathbf{w}^\top \mathbf{x}_t \right)^2
\]

\[
= \arg \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n} \sum_{t=1}^{n} (\mathbf{w}^\top (\mathbf{x}_t - \mu))^2
\]

\[
= \arg \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n} \sum_{t=1}^{n} \mathbf{w}^\top (\mathbf{x}_t - \mu)(\mathbf{x}_t - \mu)^\top \mathbf{w}
\]

\[
= \arg \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top \Sigma \mathbf{w}
\]

\(\Sigma\) is the covariance matrix
Review

• Review covariance

• Review Eigen vectors
Its a $d \times d$ matrix, $\Sigma[i, j]$ measures “covariance” of features $i$ and $j$

$$\Sigma[i, j] = \frac{1}{n} \sum_{t=1}^{n} (x_t[i] - \mu[i])(x_t[j] - \mu[j])$$
Covariance matrix:

\[
\Sigma = \frac{1}{n} \sum_{t=1}^{n} (x_t - \mu)(x_t - \mu)^\top
\]

- It's a \( d \times d \) matrix, \( \Sigma[i, j] \) measures "covariance" of features \( i \) and \( j \)

\[
\Sigma[i, j] = \frac{1}{n} \sum_{t=1}^{n} (x_t[i] - \mu[i])(x_t[j] - \mu[j])
\]
What are Eigen Vectors?
What are Eigen Vectors?

$x \mapsto Ax$
What are Eigen Vectors?

\[ x \mapsto Ax \]
What are Eigen Vectors?

$x \mapsto Ax$

$Ax = \lambda x$
What are Eigen Vectors?

$x \rightarrow Ax$

$Ax = \lambda x$