(One) bad case for k-means

- Clusters may overlap
- Some clusters may be “wider” than others
Gaussians in $m$ Dimensions

$$P(x) = \frac{1}{(2\pi)^{m/2}\|\Sigma\|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right]$$

Suppose You Have a Gaussian For Each Class

$$P(x \mid y = i) \propto \frac{1}{(2\pi)^{m/2}\|\Sigma_i\|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i)\right]$$
Gaussian Bayes Classifier

You have a Gaussian over $x$ for each class $y=i$:

$$P(x \mid y = i) = \frac{1}{(2\pi)^{m/2} \mid \Sigma \mid^{1/2}} \exp \left[ -\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right]$$

But you need probability of class $y=i$ given $x$:

$$P(y=i \mid x)$$

Thank you Bayes Rule!!

$$P(y = i \mid x) = \frac{P(x \mid y = i) P(y = i)}{p(x)}$$

Predicting wealth from age

<table>
<thead>
<tr>
<th>wealth = poor</th>
<th>wealth = rich</th>
</tr>
</thead>
<tbody>
<tr>
<td>feature = age</td>
<td>feature = age</td>
</tr>
<tr>
<td>mean</td>
<td>mean</td>
</tr>
<tr>
<td>37.374</td>
<td>44.772</td>
</tr>
<tr>
<td>cov</td>
<td>cov</td>
</tr>
<tr>
<td>190.005</td>
<td>111.00</td>
</tr>
</tbody>
</table>

$P(y = \text{poor} \mid x)$

$P(y = \text{rich} \mid x)$
Predicting wealth from age

Learning model \( \text{year}, \text{mpg} \rightarrow \text{maker} \)

\[
\Sigma = \begin{pmatrix}
\sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1m} \\
\sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{1m} & \sigma_{2m} & \cdots & \sigma_m^2
\end{pmatrix}
\]
General: \( O(m^2) \)

parameters

\[
\Sigma = \begin{pmatrix}
\sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1m} \\
\sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{1m} & \sigma_{2m} & \cdots & \sigma_m^2
\end{pmatrix}
\]

Aligned: \( O(m) \)

parameters

\[
\Sigma = \begin{pmatrix}
\sigma_1^2 & 0 & 0 & \cdots & 0 & 0 \\
0 & \sigma_2^2 & 0 & \cdots & 0 & 0 \\
0 & 0 & \sigma_3^2 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \sigma_{m-1}^2 & 0 \\
0 & 0 & 0 & \cdots & 0 & \sigma_m^2
\end{pmatrix}
\]
Aligned: $O(m)$
parameters

$$\Sigma = \begin{pmatrix}
\alpha_1^2 & 0 & 0 & \cdots & 0 & 0 \\
0 & \alpha_2^2 & 0 & \cdots & 0 & 0 \\
0 & 0 & \alpha_3^2 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \alpha_{m-2}^2 & 0 \\
0 & 0 & 0 & \cdots & 0 & \alpha_m^2
\end{pmatrix}$$

Spherical: $O(1)$
cov parameters

$$\Sigma = \begin{pmatrix}
\alpha^2 & 0 & 0 & \cdots & 0 & 0 \\
0 & \alpha^2 & 0 & \cdots & 0 & 0 \\
0 & 0 & \alpha^2 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \alpha^2 & 0 \\
0 & 0 & 0 & \cdots & 0 & \alpha^2
\end{pmatrix}$$
What if we want to do density estimation with multimodal or clumpy data?

Next... back to Density Estimation

Spherical: $O(1)$

$\Sigma = \begin{pmatrix}
\sigma^2 & 0 & 0 & \cdots & 0 & 0 \\
0 & \sigma^2 & 0 & \cdots & 0 & 0 \\
0 & 0 & \sigma^2 & \cdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \cdots & \sigma^2 & 0 \\
0 & 0 & 0 & \cdots & 0 & \sigma^2
\end{pmatrix}$
But we don’t see class labels!!!

- **MLE:**
  - \( \text{argmax} \prod_j P(y^j, x^j) \)

But we don’t know \( y^j \)!!!

- **Maximize marginal likelihood:**
  - \( \text{argmax} \prod_j P(x^j) = \text{argmax} \prod_j \sum_{i=1}^k P(y^j = i, x^j) \)

**Special case: spherical Gaussians and hard assignments**

\[
P(y = i \mid x^j) \propto \frac{1}{(2\pi)^{m/2} \|\Sigma_i\|^{1/2}} \exp \left[ -\frac{1}{2} (x^j - \mu_i)^T \Sigma_i^{-1} (x^j - \mu_i) \right] P(y = i)
\]

- If \( P(X\mid Y=i) \) is spherical, with same \( \sigma \) for all classes:
  - \( P(x^j \mid y = i) \propto \exp \left[ -\frac{1}{2\sigma^2} \| x^j - \mu_i \|^2 \right] \)

- If each \( x_j \) belongs to one class \( C(j) \) (hard assignment), marginal likelihood:
  - \[
  \prod_j \sum_i \prod_{j=1}^m P(x^j, y = i) \propto \prod_j \exp \left[ -\frac{1}{2\sigma^2} \| x^j - \mu_{C(j)} \|^2 \right]
  \]

- Same as K-means!!!
The GMM assumption

• There are $k$ components
• Component $i$ has an associated mean vector $\mu_i$

Each data point is generated according to the following recipe:

Each component generates data from a Gaussian with mean $m_i$ and covariance matrix $\sigma^2 I$
The GMM assumption

- There are $k$ components
- Component $i$ has an associated mean vector $\mu_i$
- Each component generates data from a Gaussian with mean $m_i$ and covariance matrix $\sigma^2 I$

Each data point is generated according to the following recipe:

1. Pick a component at random: Choose component $i$ with probability $P(y=i)$
2. Datapoint $\sim N(\mu_i, \sigma^2 I)$
The General GMM assumption

- There are k components
- Component $i$ has an associated mean vector $\mu_i$
- Each component generates data from a Gaussian with mean $\mu_i$ and covariance matrix $\Sigma_i$

Each data point is generated according to the following recipe:

1. Pick a component at random:
   Choose component $i$ with probability $P(y=i)$
2. Datapoint $\sim N(\mu_i, \Sigma_i)$
Supervised Learning of Mixtures of Gaussians

- Mixtures of Gaussians:
  - Prior class probabilities: \( P(y) \)
  - Likelihood function per class: \( P(x|y=i) \)

- Suppose, for each data point, we know location \( x \) and class \( y \)
  - Learning is easy... 😊
  - For prior \( P(y) \)
  - For likelihood function:

Unsupervised Learning: not as hard as it looks

- Sometimes easy
- Sometimes impossible
- and sometimes in between

*IN CASE YOU'RE WONDERING WHAT THESE DIAGRAMS ARE, THEY SHOW 2-d UNLABELED DATA (X VECTORS) DISTRIBUTED IN 2-d SPACE. THE TOP ONE HAS THREE VERY CLEAR GAUSSIAN CENTERS*
EM: “Reducing” Unsupervised Learning to Supervised Learning

- If we knew assignment of points to classes → Supervised Learning!

- Expectation-Maximization (EM)
  - Guess assignment of points to classes
  - Recompute model parameters
  - Iterate

The E.M. Algorithm

- We’ll get back to unsupervised learning soon
- But now we’ll look at an even simpler case with hidden information
- The EM algorithm
  - Can do trivial things, such as the contents of the next few slides
  - An excellent way of doing our unsupervised learning problem, as we’ll see
  - Many, many other uses…
Silly Example

Let events be “grades in a class”

\[ w_1 = \text{Gets an A} \quad P(A) = \frac{1}{2} \]
\[ w_2 = \text{Gets a B} \quad P(B) = \mu \]
\[ w_3 = \text{gets a C} \quad P(C) = 2\mu \]
\[ w_4 = \text{Gets a D} \quad P(D) = \frac{1}{2} - 3\mu \]

(Note \[ 0 \leq \mu \leq \frac{1}{6} \])

Assume we want to estimate \( \mu \) from data. In a given class there were:

- \( a \) A’s
- \( b \) B’s
- \( c \) C’s
- \( d \) D’s

What’s the maximum likelihood estimate of \( \mu \) given \( a, b, c, d \)?

Trivial Statistics

\[ P(A) = \frac{1}{2} \quad P(B) = \mu \quad P(C) = 2\mu \quad P(D) = \frac{1}{2} - 3\mu \]

\[ P(a, b, c, d \mid \mu) = K(\mu)^a(2\mu)^b((\frac{1}{2} - 3\mu))^c \]

\[ \log P(a, b, c, d \mid \mu) = \log K + a\log \frac{1}{2} + b\log \mu + c\log 2\mu + d\log (\frac{1}{2} - 3\mu) \]

**FOR MAX LIKE \( \mu \), SET**

\[ \frac{\partial \log P}{\partial \mu} = 0 \]

\[ \frac{\partial \log P}{\partial \mu} = \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{1/2 - 3\mu} = 0 \]

Gives max like \( \mu = \frac{b + c}{6(b + c + d)} \)

So if class got

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>6</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Max like \( \mu = \frac{1}{10} \)

©Carlos Guestrin 2005-2013

Boring, but true!
Same Problem with Hidden Information

Someone tells us that

Number of High grades (A's + B's) = \( h \)
Number of C's = \( c \)
Number of D's = \( d \)

What is the max. like estimate of \( \mu \) now?

We can answer this question circularly:

**EXPECTATION**

If we know the value of \( \mu \) we could compute the expected value of \( a \) and \( b \)

\[
a = \frac{1}{2 + \mu} h, \quad b = \frac{\mu}{1 + \mu} h
\]

Since the ratio \( a:b \) should be the same as the ratio \( \frac{1}{2} : \mu \)

**MAXIMIZATION**

If we know the expected values of \( a \) and \( b \) we could compute the maximum likelihood value of \( \mu \)

\[
\mu = \frac{b + c}{6(b + c + d)}
\]

E.M. for our Trivial Problem

We begin with a guess for \( \mu \)

We iterate between **EXPECTATION** and **MAXIMALIZATION** to improve our estimates of \( \mu \) and \( a \) and \( b \).

Define \( \mu^{(t)} \) the estimate of \( \mu \) on the \( t \)th iteration

\( b^{(t)} \) the estimate of \( b \) on \( t \)th iteration

\[
\mu^{(0)} = \text{initial guess}
\]

\[
b^{(t)} = \frac{\mu^{(t)} h}{1/2 + \mu^{(t)}} = \mathbb{E}[b | \mu^{(t)}]
\]

\[
\mu^{(t+1)} = \frac{b^{(t)} + c}{6(b^{(t)} + c + d)} = \text{max like est. of } \mu \text{ given } b^{(t)}
\]

Continue iterating until converged.

Good news: Converging to local optimum is assured.

Bad news: I said “local” optimum.
E.M. Convergence

- Convergence proof based on fact that Prob(data | µ) must increase or remain same between each iteration \[\text{[not obvious]}\]
- But it can never exceed 1 \[\text{[obvious]}\]
  So it must therefore converge \[\text{[obvious]}\]

In our example, suppose we had

\begin{align*}
h &= 20 \\
c &= 10 \\
d &= 10 \\
\mu^{(0)} &= 0
\end{align*}

<table>
<thead>
<tr>
<th>t</th>
<th>µ^{(t)}</th>
<th>b^{(t)}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.0833</td>
<td>2.857</td>
</tr>
<tr>
<td>2</td>
<td>0.0937</td>
<td>3.158</td>
</tr>
<tr>
<td>3</td>
<td>0.0947</td>
<td>3.185</td>
</tr>
<tr>
<td>4</td>
<td>0.0948</td>
<td>3.187</td>
</tr>
<tr>
<td>5</td>
<td>0.0948</td>
<td>3.187</td>
</tr>
<tr>
<td>6</td>
<td>0.0948</td>
<td>3.187</td>
</tr>
</tbody>
</table>

Convergence is generally linear: error decreases by a constant factor each time step.

Back to Unsupervised Learning of Mixtures of Gaussians – a simple version

A simple case:

- We have unlabeled data \(x_1, x_2, \ldots, x_m\)
- We know there are \(k\) classes
- We know \(P(y_1) P(y_2) P(y_3) \ldots P(y_k)\)
- We don’t know \(\mu_1, \mu_2, \ldots, \mu_k\)

We can write \(P(\text{data} | \mu_1, \ldots, \mu_k)\)

\[
\begin{align*}
  &= p(x_1, \ldots, x_m | \mu_1, \ldots, \mu_k) \\
  &= \prod_{j=1}^m p(x_j | \mu_1, \ldots, \mu_k) \\
  &= \prod_{j=1}^m \sum_{i=1}^k p(x_j | \mu_i) P(y = i) \\
  &\propto \prod_{j=1}^m \sum_{i=1}^k \exp \left( -\frac{1}{2\sigma^2} \|x_j - \mu_i\|^2 \right) P(y = i)
\end{align*}
\]
EM for simple version of Mixtures of Gaussians: The E-step

- If we know \( \mu_1, \ldots, \mu_k \) → easily compute prob. point \( x_j \) belongs to class \( y = i \)

\[
p(y = i | x_j, \mu_1, \ldots, \mu_k) \propto \exp \left( -\frac{1}{2\sigma^2} \left\| x_j - \mu_i \right\|^2 \right) p(y = i)
\]

EM for simple version of Mixtures of Gaussians: The M-step

- If we know prob. point \( x_j \) belongs to class \( y = i \) → MLE for \( \mu_i \) is weighted average
  - imagine k copies of each \( x_j \), each with weight \( P(y = i | x_j) \):

\[
\mu_i = \frac{\sum_{j=1}^{n} P(y = i | x_j) x_j}{\sum_{j=1}^{n} P(y = i | x_j)}
\]
E.M. for Simple version of Mixtures of Gaussians

E-step
Compute “expected” classes of all datapoints for each class

\[ p(y = i | x_j, \mu_1, \ldots, \mu_k) \propto \exp\left(-\frac{1}{2\sigma^2} \| x_j - \mu_i \|^2 \right) p(y = i) \]

M-step
Compute Max. like \( \mu \) given our data’s class membership distributions

\[ \mu_i = \frac{\sum_{j=1}^{m} P(y = i | x_j) x_j}{\sum_{j=1}^{m} P(y = i | x_j)} \]

E.M. Convergence

- EM is coordinate ascent on an interesting potential function
- Coord. ascent for bounded pot. func. ! convergence to a local optimum guaranteed

- This algorithm is REALLY USED. And in high dimensional state spaces, too. E.G. Vector Quantization for Speech Data