EM Algorithm &
High Dimensional Data

Ken Kreutz-Delgado
(Nuno Vasconcelos)

ECE 175A – Winter 2012 – UCSD
Gaussian EM Algorithm

For the Gaussian mixture model, we have

- **Expectation Step (E-Step):**

  \[ h_{ij} = P_{Z|X}(j|x_i) = \frac{\mathcal{G}(x_i, \mu_j, \sigma_j) \pi_j}{\sum_{k=1}^{C} \mathcal{G}(x_i, \mu_k, \sigma_k) \pi_k} \]

- **Maximization Step (M-Step):**

  \[
  \mu_{j}^{\text{new}} = \frac{\sum_i h_{ij} x_i}{\sum_i h_{ij}} \quad \pi_j^{\text{new}} = \frac{1}{n} \sum_i h_{ij} \\
  \sigma_{j}^{2\text{new}} = \frac{\sum_i h_{ij} (x_i - \mu_j)^2}{\sum_i h_{ij}} \]

EM versus K-means

**EM**

**k-Means**

### Data Class Assignments

**Soft Decisions:**

\[ h_{ij} = P_{Z|X}(j | x_i) \]

**Hard Decisions:**

\[ i^*(x_i) = \arg \max_j P_{Z|X}(j | x_i) \]

\[ h_{ij} = \begin{cases} 1, & P_{Z|X}(j | x_i) > P_{Z|X}(k | x_i), k \neq j \\ 0, & \text{otherwise} \end{cases} \]

### Parameter Updates

**Soft Updates:**

\[ \mu_{j}^{\text{new}} = \frac{\sum_i h_{ij} x_i}{\sum_i h_{ij}} \]

\[ \pi_{j}^{\text{new}} = \frac{1}{n} \sum_i h_{ij} \]

\[ \sigma_{j}^{2\text{new}} = \frac{\sum_i h_{ij} (x_i - \mu_j)^2}{\sum_i h_{ij}} \]

**Hard Updates:**

\[ \mu_i^{\text{new}} = \frac{1}{n} \sum_j x_j^{(i)} \]
Important Application of EM

Recall, in **Bayesian decision theory** we have

- **World:** States $Y$ in \{1, ..., $M$\} and **observations** of $X$
- Class-conditional densities $P_{X|Y}(x|y)$
- Class (prior) probabilities $P_Y(i)$
- Bayes decision rule (BDR)

\[
i^* = \arg \max_i P_{X|Y}(x|i) P_Y(i)\]

We have seen that this is **only optimal insofar as all probabilities involved are correctly estimated**

**One of the important applications of EM is to more accurately learn the class-conditional densities**
Example

► **Image segmentation:**
  - Given this image, can we segment it into the cheetah and background classes?
  - Useful for **many applications**
    - **Recognition:** “this image has a cheetah”
    - **Compression:** code the cheetah with fewer bits
    - **Graphics:** plug in for photoshop would allow manipulating objects

► Since we have two classes (cheetah and grass), we should be able to do this with a Bayesian classifier
Example

Start by collecting **a lot of examples of cheetahs**

and **a lot of examples of grass**

One can get tons of such images via Google image search
Example

- Represent images as **bags of little image patches**
- We can fit a simple Gaussian to the transformed patches

![Diagram showing image patches being transformed through a discrete cosine transform, resulting in a bag of DCT vectors, and then fit with a Gaussian distribution.](image-url)
Example

- Do the same for grass and apply BDR to each patch to classify each patch into “cheetah” or “grass”

\[ i^* = \arg \max_i \{ \log G(x, \mu_i, \Sigma_i) + \log P_Y(i) \} \]

Bag of DCT vectors

Discrete cosine transform

\( P_{X|W}(x | \text{grass}) \)

\( P_{X|W}(x | \text{cheetah}) \)
Example

Better performance is achieved by modeling the cheetah class distribution as a mixture of Gaussians.
Example

Do the same for grass and apply BDR to each patch to classify

$$i^* = \arg \max_i \left\{ \log \sum_k G(x, \mu_{i,k}, \Sigma_{i,k}) \pi_{i,k} + \log P_Y(i) \right\}$$

Bag of DCT vectors

discrete cosine transform

$$P_{X|Y}(x|\text{grass})$$

$$P_{X|Y}(x|\text{cheetah})$$
The use of more sophisticated probability models, e.g. mixtures, *usually* improves performance.

However, it is not a magic solution.

Earlier on in the course we talked about features.

Typically, you have to start from a good feature set.

It turns out that even with a good feature set, you must be careful.

Consider the following example, from our image classification problem.
Example

- **Cheetah Gaussian classifier, DCT space**

  8 first DCT features
  all 64 features

  Prob. of error: 4%
  8%

  Interesting observation: **more features = higher error**!
Comments on the Example

- The first reason why this happens is that things are not always what we think they are in high dimensions.
- One could say that high dimensional spaces are STRANGE!!!
- In practice, we invariably have to do some form of dimensionality reduction.
- We will see that eigenvalues play a major role in this.
- One of the major dimensionality reduction techniques is principal component analysis (PCA).
- But let’s start by discussing the problems of high dimensions.
High Dimensional Spaces

Are strange!

First thing to know:

“Never fully trust your intuition in high dimensions!”

More often than not you will be wrong!

- There are many examples of this
- We will do a couple here, skipping most of the math
- These examples are both fun and instructive
The Hypersphere

Consider the **ball** of radius $r$ in a space of dimension $d$

$$S = \left\{ x \mid \sum_{i=1}^{d} x_i^2 \leq r^2 \right\}$$

The **surface** of this ball is a $(d-1)$-dimensional hypersphere.

The ball has volume

$$V_d(r) = \frac{r^d \pi^{d/2}}{\Gamma\left(\frac{d}{2} + 1\right)}$$

where $\Gamma(n)$ is the **gamma function**

$$\Gamma(n) = \int_{0}^{\infty} e^{-x} x^{n-1} \, dx$$

When we talk of the “**volume of a hypersphere**”, we will actually mean the volume of the ball it contains.

Similarly, for “the volume of a hypercube”, etc.
Hypercube versus Hypersphere

Consider the hypercube \([-a,a]^d\) and an inscribe hypersphere:

Q: what does your intuition tell you about the relative sizes of these two volumes?

1. volume of sphere \(\approx\) volume of cube?
2. volume of sphere \(>>\) volume of cube?
3. volume of sphere \(<<\) volume of cube?
Answer

- To find the answer, we can compute the relative volumes:

\[ f_d = \frac{Vol(sphere)}{Vol(cube)} = \frac{a^d \pi \frac{d}{2}}{\Gamma \left( \frac{d}{2} + 1 \right)} = \frac{\pi \frac{d}{2}}{2^d \Gamma \left( \frac{d}{2} + 1 \right)} \]

- This is a sequence that does not depend on the radius \( a \), just on the dimension \( d \)!

<table>
<thead>
<tr>
<th>( d )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_d )</td>
<td>1.00</td>
<td>.785</td>
<td>.524</td>
<td>.308</td>
<td>.164</td>
<td>.08</td>
<td>.037</td>
</tr>
</tbody>
</table>

- The relative volume goes to zero, and goes to zero fast!
This means that:

“As the dimension of the space increases, the volume of the sphere is much smaller (infinitesimally so) than that of the cube!”

Is this *really* going against intuition?

It is actually not very surprising, if we think about it. We can see it even in low dimensions:

1. $d = 1$  
   - Volume is the same

2. $d = 2$  
   - Volume of sphere is already smaller
**Hypercube vs Hypersphere**

- As the dimension increases, the volume of the shaded corners becomes larger.

- **In high dimensions** the picture you should imagine is:

  All the volume of the cube is in the “spikes” (corners)!
Believe it or Not …

… we can actually check this mathematically: Consider $d$ and $p$

\[ d = (a, a, \ldots, a) \in \mathcal{R}^d \]

\[ p = (a, 0, \ldots, 0) \in \mathcal{R}^d \]

**note that**

\[
\cos \theta = \frac{d^T p}{\sqrt{||d||^2 ||p||^2}}
\]

\[
= \frac{a^2}{\sqrt{da^2 a^2}} = \frac{1}{\sqrt{d}} \to 0
\]

**$d$ becomes **orthogonal** to $p$ as $d$ increases,**

and **infinitely larger**!!!
But there is even more …

- Consider the crust of unit sphere of thickness $\varepsilon$
- We can compute the volume of the crust:

$$Vol(crust) = \left[ 1 - \frac{Vol(S_1)}{Vol(S_2)} \right] Vol(S_2)$$

$$\frac{Vol(S_1)}{Vol(S_2)} = \frac{(a-\varepsilon)^d \pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)} \Rightarrow \frac{a^d \left(1 - \frac{\varepsilon}{a}\right)^d}{a^d \pi^{d/2} \Gamma\left(\frac{d}{2}+1\right)} = \left(1 - \frac{\varepsilon}{a}\right)^d$$

- No matter how small $\varepsilon$ is, ratio goes to zero as $d$ increases
- I.e. “all the volume is in the crust!”
High Dimensional Gaussian

For a Gaussian, it can be shown that if

\[ \mathbf{X} \sim \mathcal{N}(0, \mathbf{I}), \quad \mathbf{x} \in \mathbb{R}^n \]

and one considers the region outside of the hypersphere where the probability density drops to 1\% of peak value

\[ S_{0.01}(\mathbf{x}) = \left\{ \mathbf{x} \mid \frac{G(\mathbf{x}, 0, \mathbf{I})}{G(0, 0, \mathbf{I})} \leq 0.01 \right\} \]

then the probability mass in this region is

\[ P_n = P[\chi^2(n) \geq 9.21] \]

where \( \chi^2(n) \) is a chi-squared random variable with \( n \) degrees of freedom.
High-Dimensional Gaussian

- If you evaluate this, you’ll find out that

<table>
<thead>
<tr>
<th>$n$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1-P_n$</td>
<td>.998</td>
<td>.99</td>
<td>.97</td>
<td>.94</td>
<td>.89</td>
<td>.83</td>
<td>.48</td>
<td>.134</td>
<td>.02</td>
</tr>
</tbody>
</table>

- As the dimension increases, virtually all the probability mass is in the tails

- Yet, the point of maximum density is still the mean

- This is really strange: in high-dimensions the Gaussian is a very heavy-tailed distribution

- Take-home message:
  - “In high dimensions never trust your low-dimensional intuition!”
The Curse of Dimensionality

Typical observation in Bayes decision theory:

- Error increases when number of features is large

This is unintuitive since theoretically:

- If I have a problem in $n$ dimensions I can always generate a problem in $n+1$ dimensions without increasing the probability of error, and even often decreasing the probability of error.

E.g. two uniform classes in 1-D

![Diagram showing two uniform classes in 1-D that can be transformed into a 2-D problem with the same error]

- Just add a non-informative variable (extra dimension) $y$.

The Curse of Dimensionality
Sometimes it is possible to reduce the error by adding a second variable which is informative

- On the left there is no decision boundary that will achieve zero error
- On the right, the decision boundary shown has zero error
Curse of Dimensionality

In fact, it is \textit{theoretically impossible} to do worse in 2-D than 1-D:

If we move the classes along the lines shown in green the error can only go down, since there will be less overlap
Curse of Dimensionality

- So **why** do we observe this **“curse of dimensionality”**?
- The problem is the **quality of the density estimates**
- All we have seen so far, **assumes perfect estimation** of the BDR
- We discussed various reasons why **this is not easy**
  - Most densities are **not** simply a Gaussian, exponential, etc
  - Typically densities are, **at best**, a **mixture** of several components.
  - There are **many unknowns** (# of components, what type), the likelihood has **local minima**, etc.
  - Even with algorithms like EM, it is difficult to get this right
Curse of Dimensionality

But the problem goes much deeper than this.

Even for simple models (e.g. Gaussian) we need a large number of examples \( n \) to have good estimates.

Q: What does “large” mean?
This depends on the dimension of the space.

The best way to see this is to think of an histogram:

- Suppose you have 100 points and you need at least 10 bins per axis in order to get a reasonable quantization.
- For uniform data you get, on average:

<table>
<thead>
<tr>
<th>dimension</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>points/bin</td>
<td>10</td>
<td>1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

- This is decent in 1-D; bad in 2-D; terrible in 3-D (9 out of each 10 bins empty).
Dimensionality Reduction

- We see that it can quickly become impossible to fill up a high dimensional space with a sufficient number of data points.
  - What do we do about this? We avoid unnecessary dimensions!

- Unnecessary can be measured in two ways:
  1. Features are non-discriminant (insufficiently discriminating)
  2. Features are not independent

- Non-discriminant means that they don’t separate classes well
END