COMP 330: Learning Mixture Models

Chris Jermaine and Kia Teymourian

Rice University

We've Talked A Lot About Some Exmaple Mixture Models

But how to learn them?

- MLE: standard is EM
- Bayesian: typically use Gibbs sampling

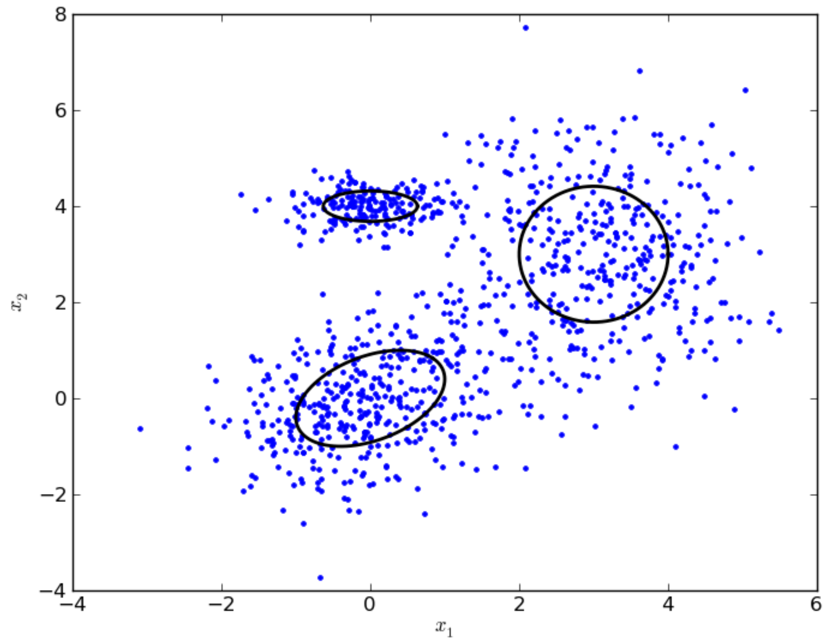
This lecture: will focuss on GMM learning

Concepts easily extended to other mixtures

Gaussian Mixture Modeling

Recall: in GMM, assume data from the following stochastic program:

For i = 1 to n **do:**
  $c_i \sim \text{Categorical}(\pi)$
  $x_i \sim \text{Normal}(\mu_{c_i}, \Sigma_{c_i})$



EM for GMM

We won't derive it, but will just describe algorithm

Goal is to learn k components over n data points

EM for GMM

We won't derive it, but will just describe algorithm

Goal is to learn k components over n data points

Begin by initializing

- Choose a set of k points randomly, $\langle s_1, s_2, \ldots, s_k \rangle$
- Use those to init each μ: $\mu_1 = s_1$, $\mu_2 = s_2$, etc.
- Then set each Σ to be the variance of the data
- That is, each $\Sigma_{j,j} = \sum_i \left(\left(\sum_{i'} x_{i',j} / n \right) - x_{i,j} \right)^2$
- All off-diagonals are zero

The E-Step

Compute a "soft" assignment of data to each cluster

Simply compute $Pr[c_i = j|x_i]$ for each i, j

From Bayes' rule:

$$
\begin{aligned}
Pr[c_i = j|x_i] &= \frac{f(c_i = j, x_i)}{f(x_i)} \\
&= \frac{\pi_j \text{Normal}(x_i|\mu_j, \Sigma_j)}{\sum_{j'} \pi_{j'} \text{Normal}(x_i|\mu_{j'}, \Sigma_{j'})}
\end{aligned}
$$

Easy to do in a MapReduce job!

The M-Step

First, simply do an MLE for each of the k Gaussians

Where each point fractionally contributes to MLE (based on E step)

Update for μ_j:

$$\mu_j = \frac{\sum_i Pr[c_i = j | x_i] \times x_i}{\sum_i Pr[c_i = j | x_i]}$$

Note that each x_i is a vector!

The M-Step

First, simply do an MLE for each of the k Gaussians

Where each point fractionally contributes to MLE (based on E step)

Update for μ_j:

$$\mu_j = \frac{\sum_i Pr[c_i = j | x_i] \times x_i}{\sum_i Pr[c_i = j | x_i]}$$

Note that each x_i is a vector!

Update for σ_j:

$$\sigma_j = \frac{\sum_i Pr[c_i = j | x_i] \text{outerProd}\left(x_i - \mu_j, x_i - \mu_j\right)}{\sum_i Pr[c_i = j | x_i]}$$

The M-Step

Lastly, update for π_j:

$$\pi_j = \sum_i Pr[c_i = j | x_i]/n$$

Also easy to do over MapReduce!

A Few Caveats

Fails if less than d points have non-zero probability of belonging to a cluster

- d is dimensionality of the data
- Why? Covariance matrix is singular

A Few Caveats

Fails if less than d points have non-zero probability of belonging to a cluster

- d is dimensionality of the data
- Why? Covariance matrix is singular

Very sensitive to poor initialization

- Typical: run many times, with different initializations
- Choose one with highest likelihood

Now, Let's Go Bayesian

Recall: Bayesians never estimate parameters

They only estimate variables

Everything has a prior!

So this:

For i = 1 to n **do:**
$c_i \sim \text{Categorical}(\pi)$
$x_i \sim \text{Normal}(\mu_{c_i}, \Sigma_{c_i})$

Now, Let's Go Bayesian

Recall: Bayesians never estimate parameters

They only estimate variables

Everything has a prior!

Becomes this:

$\pi \sim \text{Dirichlet}(\alpha)$
For j = 1 to k **do**:
    $\mu_j \sim \text{Normal}(\mu_0, \Sigma_0)$
    $\Sigma_j \sim \text{InvWishart}(\nu, \Psi)$

For i = 1 to n **do**:
    $c_i \sim \text{Categorical}(\pi)$
    $x_i \sim \text{Normal}(\mu_{c_i}, \Sigma_{c_i})$

PDF

First step in deriving an MCMC algorithm to learn this model:

- Get a PDF!

$\pi \sim \text{Dirichlet}(\alpha)$

For j = 1 to k **do**:

$\quad \mu_j \sim \text{Normal}(\mu_0, \Sigma_0)$

$\quad \Sigma_j \sim \text{InvWishart}(\nu, \Psi)$

For i = 1 to n **do**:

$\quad c_i \sim \text{Categorical}(\pi)$

$\quad x_i \sim \text{Normal}(\mu_{c_i}, \Sigma_{c_i})$

PDF is simply:

PDF

- PDF is simply:

$$\text{Dirichlet}(\pi|\alpha)\times$$
$$\prod_j \text{Normal}(\mu_j|\mu_0, \Sigma_0) \times \text{InvWishart}(\Sigma_j|(\nu, \Psi)\times$$
$$\prod_i \text{Categorical}(c_i|\pi) \times \text{Normal}(x_i|\mu_{c_i}, \Sigma_{c_i})$$

Deriving a Gibbs Sampler

Initialize each unseen variable with an appropriate value

Then go through all of the variables cyclically

- Drop any part of the PDF that does not contain the variable in question
- And sample from the resulting function!

Update for Pi

- Start with the PDF:

$$\text{Dirichlet}(\pi|\alpha)\times$$
$$\prod_j \text{Normal}(\mu_j|\mu_0,\Sigma_0) \times \text{InvWishart}(\Sigma_j|(\nu,\Psi)\times$$
$$\prod_i \text{Categorical}(c_i|\pi) \times \text{Normal}(x_i|\mu_{c_i},\Sigma_{c_i})$$

- Drop all terms without π:

$$\text{Dirichlet}(\pi|\alpha) \times \prod_i \text{Categorical}(c_i|\pi)$$

- How to sample from this??

Update for Pi

- Start with the PDF:

$$\text{Dirichlet}(\pi|\alpha)\times$$
$$\prod_j \text{Normal}(\mu_j|\mu_0, \Sigma_0) \times \text{InvWishart}(\Sigma_j|(\nu, \Psi)\times$$
$$\prod_i \text{Categorical}(c_i|\pi) \times \text{Normal}(x_i|\mu_{c_i}, \Sigma_{c_i})$$

- Drop all terms without π:

$$\text{Dirichlet}(\pi|\alpha) \times \prod_i \text{Categorical}(c_i|\pi)$$

- How to sample from this??
- Easy! Wikipedia: Dirichlet is conjugate prior for Categorical!
- So use SciPy to generate the sample

Update for Component Means

- Start with the PDF:

$$\text{Dirichlet}(\pi|\alpha)\times$$
$$\prod_j \text{Normal}(\mu_j|\mu_0,\Sigma_0)\times\text{InvWishart}(\Sigma_j|(\nu,\Psi)\times$$
$$\prod_i \text{Categorical}(c_i|\pi)\times\text{Normal}(x_i|\mu_{c_i},\Sigma_{c_i})$$

- Drop all terms without μ_j:

$$\text{Normal}(\mu_j|\mu_0,\Sigma_0)\times\prod_{i\text{ s.t. }c_i=j}\text{Normal}(x_i|\mu_j,\Sigma_j)$$

- How to sample from this??
- Easy! Wikipedia: MD Normal is conjugate prior for mean of MD Normal!
- So use SciPy to generate the sample

Update for Component Covars

- Start with the PDF:

$$\text{Dirichlet}(\pi|\alpha)\times$$
$$\prod_j \text{Normal}(\mu_j|\mu_0, \Sigma_0) \times \text{InvWishart}(\Sigma_j|(\nu, \Psi)\times$$
$$\prod_i \text{Categorical}(c_i|\pi) \times \text{Normal}(x_i|\mu_{c_i}, \Sigma_{c_i})$$

- Drop all terms without Σ_j:

$$\text{InvWishart}(\Sigma_j|(\nu, \Psi) \times \prod_{i \text{ s.t. } c_i=j} \text{Normal}(x_i|\mu_j, \Sigma_j)$$

- How to sample from this??
- Easy! Wikipedia: InvWishart is conjugate prior for mean of MD Normal!
- So use SciPy to generate the sample

Final Algorithm

It is simply:

```
Initialize π, plus each c_i, μ_j, Σ_j
For count = 1 to big do:
  Update π
  For j = 1 to j do:
    Update μ_j
    Update Σ_j
  For i = 1 to n do:
    Update c_i
```

Final Remarks

We've gone thru learning for a GMM

- All other mixtures at least somewhat similar

We've actually derived a Gibbs sampler for a GMM **From Scratch**

- Super cool!

Compare and contrast: Gibbs sampler vs. EM

Questions?