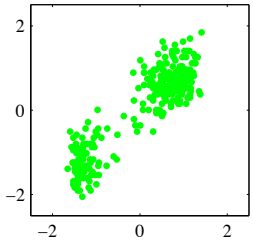
CS 335: Clustering and Mixture of Gaussians

Dan Sheldon

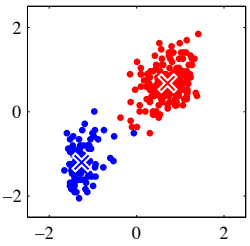
Clustering



[Bishop *Pattern Recognition and Machine Learning*]

- Suppose you are given feature vectors $\mathbf{x}^{(1)}, \dots \mathbf{x}^{(m)} \in \mathbb{R}^n$, and you believe they come from multiple different classes, but you don't have access to class labels

Clustering



[Bishop *Pattern Recognition and Machine Learning*]

- **Clustering**: automatically assign data points in n dimensions to *clusters* (classes)
- Optionally find *cluster representatives*

K-Means Clustering

Given

- Feature vectors $\mathbf{x}^{(1)}, \dots \mathbf{x}^{(m)} \in \mathbb{R}^n$
- Desired number of clusters k

Find

- Cluster labels $c^{(i)} \in \{1, 2, \dots, k\}$
- Cluster centers $\mu_1, \dots, \mu_k \in \mathbb{R}^n$

What approach? Cost function! **Minimize**

$$J(c, \mu) = \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

K-Means Algorithm

1. Initialize $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$ randomly
2. Repeat until convergence
   - For all points i, assign $\mathbf{x}^{(i)}$ to closest cluster center

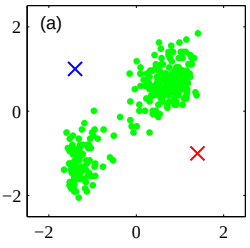
$$c^{(i)} \leftarrow \text{argmin}_j \, ||\mathbf{x}^{(i)} - \mu_j||^2$$

   - For all clusters j, set $\boldsymbol{\mu}_j$ = average of currently assigned points

$$\boldsymbol{\mu}_j \leftarrow \frac{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)} = j\}\mathbf{x}^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)} = j\}}$$

K-Means Example

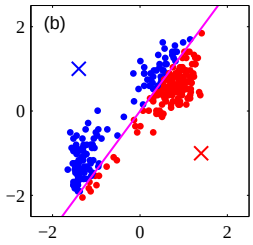
Initialize cluster centers arbitrarily:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

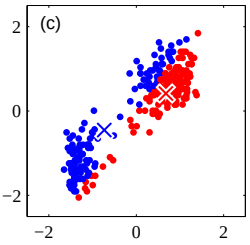
Assign points:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

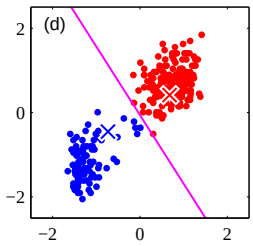
Update centers:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

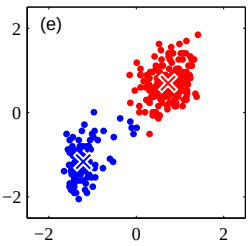
Assign points:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

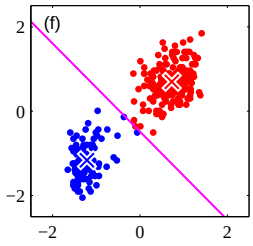
Update centers:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

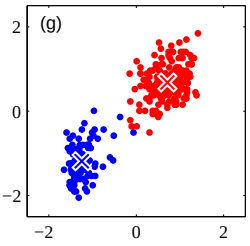
Assign points:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

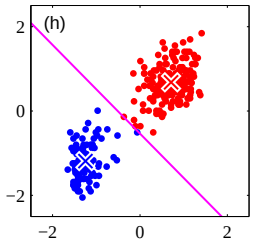
Update centers:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

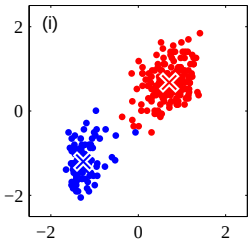
Assign points:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

Update centers:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Convergence

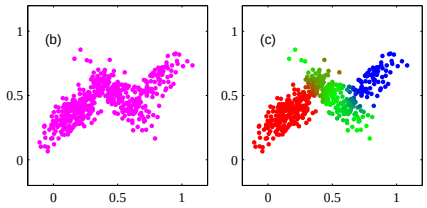
$$J(c, \mu) = \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

Not hard to show that

1. μ updates minimize J while holding c fixed
2. c updates minimize J while holding μ fixed
3. The algorithm converges

"Soft" clustering

Often desirable to fractionally assign points to clusters



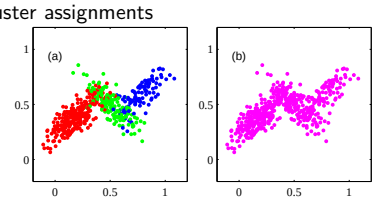
[Bishop *Pattern Recognition and Machine Learning*]

"Soft" clustering

There's something Bayesian happening here...

"Generative" Probabilistic Model: Mixture of Gaussians

- First choose cluster: $p(c^{(i)} = j) = \phi_j$
- Then generate $\mathbf{x}^{(i)}$ from conditional distribution $p(\mathbf{x}^{(i)} \mid c^{(i)})$
- Hide cluster assignments



$p(\mathbf{x}^{(i)} \mid c^{(i)} = k)$ follows a **Gaussian distribution** with parameters μ_k and Σ_k

Aside: Multivariate Gaussian Distribution

$$p(\mathbf{x};\boldsymbol{\mu},\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\Big)$$

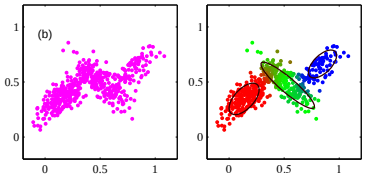
Describes random vector $\mathbf{x} \in \mathbb{R}^n$ with

- Mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$
- Covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$
- $P(\mathbf{x} \in A) = \int_A p(\mathbf{x};\boldsymbol{\mu},\Sigma)d\mathbf{x}$

Examples

Mixture of Gaussians Problem

Given feature vectors $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)}$, number of "clusters" k



Find

- Cluster priors $\phi_j = p(c = j)$
- Gaussian parameters $\boldsymbol{\mu}_j$ and Σ_j for each cluster
- Soft cluster assignments $p(c^{(i)} = j|\mathbf{x}^{(i)})$

Mixture of Gaussians Algorithm

Initialize all ϕ_j, μ_j, Σ_j

Repeat until convergence

1. Compute posterior probability that $\mathbf{x}^{(i)}$ comes from cluster j

$$\begin{aligned} w_j^{(i)} &= p(c^{(i)} = j \,|\, \mathbf{x}^{(i)}) \\ &= \frac{\phi_j \cdot p(\mathbf{x}^{(i)} \,|\, c^{(i)} = j)}{\sum_{l=1}^k \phi_l \cdot p(\mathbf{x}^{(i)} \,|\, c^{(i)} = l)} \quad \text{(Bayes rule)} \end{aligned}$$

2. Update parameters ϕ_j, μ_j, Σ_j using $w_j^{(i)}$ values as weights

Update Parameters

ϕ_j = average weight assigned to class j
$\boldsymbol{\mu}_j$ = weighted mean for class j
Σ_j = weighted covariance for class j

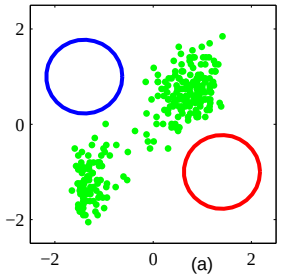
$$\phi_j = \frac{1}{m}\sum_{i=1}^m w_j^{(i)}$$

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^m w_j^{(i)}\mathbf{x}^{(i)}}{\sum_{i=1}^m w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^m w_j^{(i)}\sum_{i=1}^m(\mathbf{x}^{(i)}-\boldsymbol{\mu}_j)(\mathbf{x}^{(i)}-\boldsymbol{\mu}_j)^T}{\sum_{i=1}^m w_j^{(i)}}$$

Mixture of Gaussians

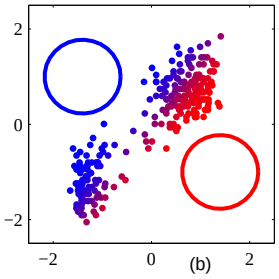
Initialize cluster parameters:



[Bishop *Pattern Recognition and Machine Learning*]

Mixture of Gaussians

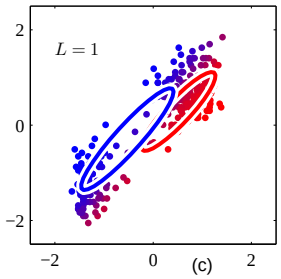
Update soft assignments:



[Bishop *Pattern Recognition and Machine Learning*]

Mixture of Gaussians

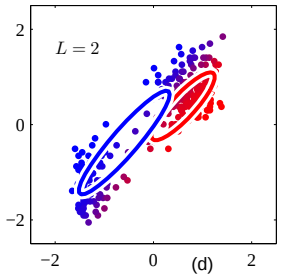
Update cluster parameters:



[Bishop *Pattern Recognition and Machine Learning*]

Mixture of Gaussians

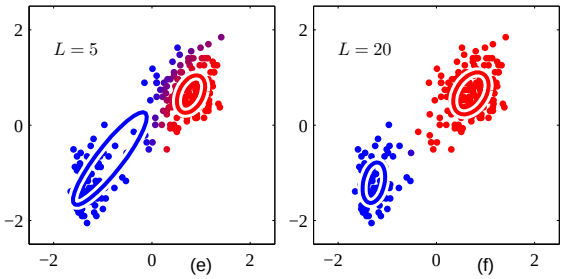
Next iteration:



[Bishop *Pattern Recognition and Machine Learning*]

Mixture of Gaussians

And so on:



[Bishop *Pattern Recognition and Machine Learning*]

Mixture of Gaussians Convergence

- This algorithm converges
- Can be formally justified as an instance of the Expectation Maximization (EM) algorithm
- For your next ML class!