# Unsupervised Continuous Clustering

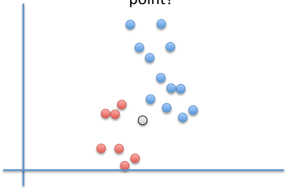
Machine Learning: Jordan Boyd-Graber
University of Colorado Boulder
LECTURE 16

## Lecture for Today

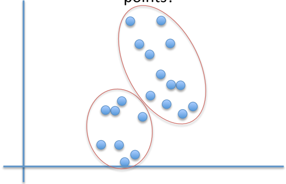
- What is clustering?
- K-Means
- Gaussian Mixture Models

# Clustering

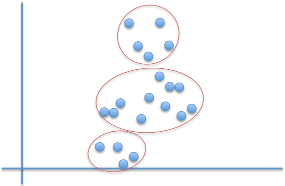
**Classification:** what is label of new point?



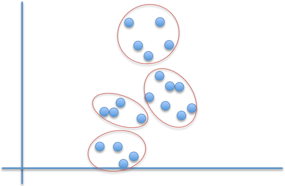
**Clustering:** how should we group these points?



**Clustering:** or is this the right grouping?



**Clustering:** what about this?

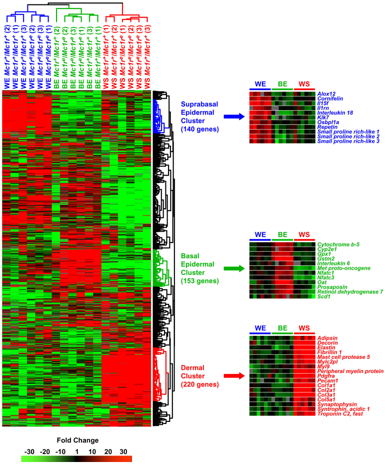


# Clustering

Uses:

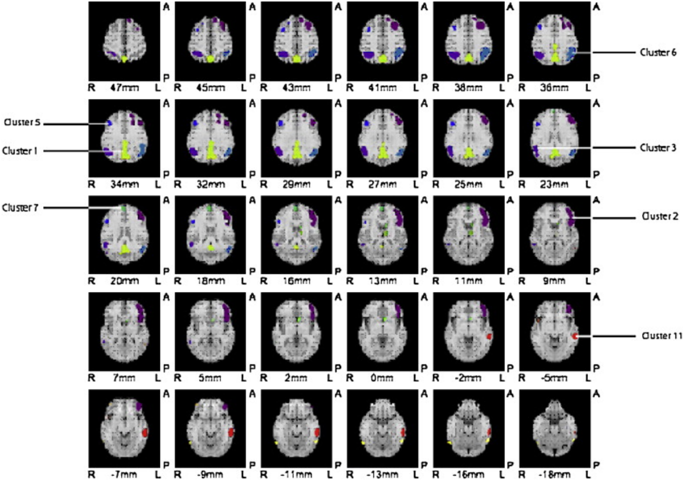
- genomics
- medical imaging
- social network analysis
- recommender systems
- market segmentation
- voter analysis

# Microarray Gene Expression Data



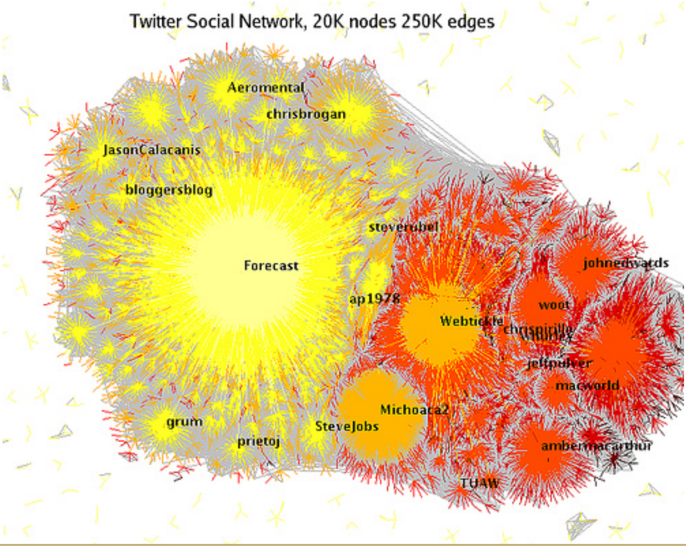

From: "Skin layer-specific transcriptional profiles in normal and recessive yellow (Mc1re/Mc1re) mice" by April and Barsh in *Pigment Cell Research* (2006)

## Medical Imaging (MRIs and PET scans)

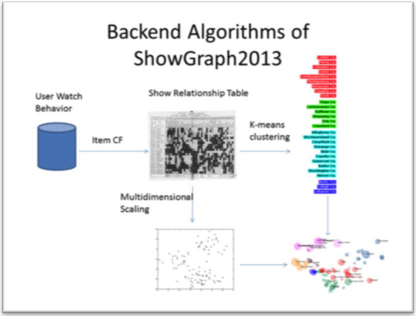


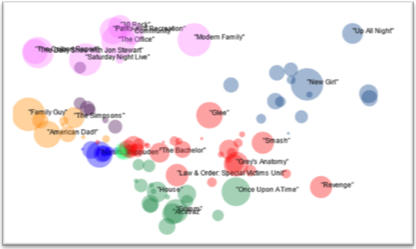

From: "Fluorodeoxyglucose positron emission tomography of mild cognitive impairment with clinical follow-up at 3 years" by Pardo et al. in *Alzheimer's and Dementia* (2010)

# Social Networks



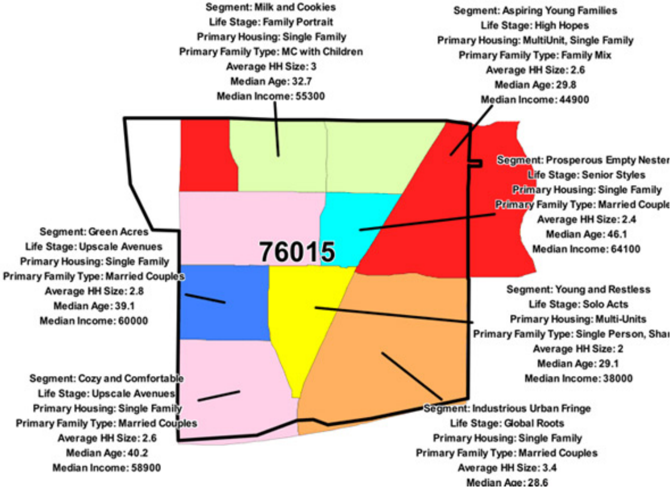
## Recommender Systems



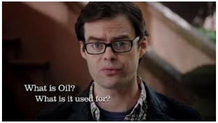

From: tech.hulu.com/blog/

# Market Segmentation




From: mappinganalytics.com/map/segmentation-maps/segmentation-map.html

## Voter Analysis




- soccer moms (female, middle aged, married, middle income, white, kids, suburban)
- Nascar dads (male, middle aged, married, middle income, white, kids, Southern, suburban or rural)
- security moms ( ... )
- low information voters
- Ivy League Elites

# Clustering

Questions:

- how do we fit clusters?
- how many clusters should we use?
- how should we evaluate model fit?

## K-Means

How do we fit the clusters?

- simplest method: K-means
- requires: real-valued data
- idea:
  - pick $K$ initial cluster means
  - associate all points closest to mean $k$ with cluster $k$
  - use points in cluster $k$ to update mean for that cluster
  - re-associate points closest to new mean for $k$ with cluster $k$
  - use new points in cluster $k$ to update mean for that cluster
  - ...
  - stop when no change between updates

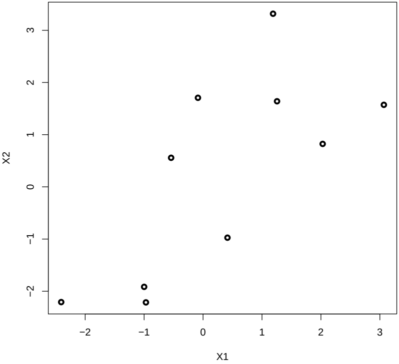
## K-Means

Animation at:
`http://shabal.in/visuals/kmeans/1.html`

## K-Means: Example

Data:

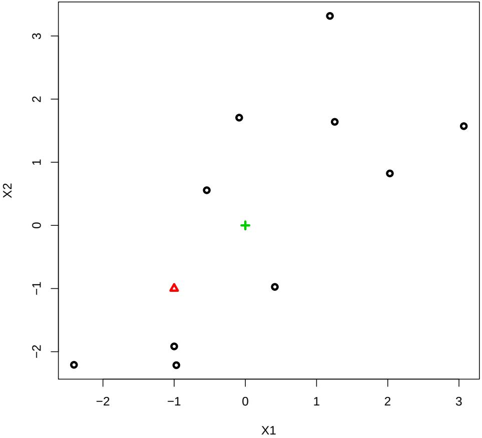
| $x_1$ | $x_2$ |
|---|---|
| 0.4 | -1.0 |
| -1.0 | -2.2 |
| -2.4 | -2.2 |
| -1.0 | -1.9 |
| -0.5 | 0.6 |
| -0.1 | 1.7 |
| 1.2 | 3.3 |
| 3.1 | 1.6 |
| 1.3 | 1.6 |
| 2.0 | 0.8 |



## K-Means: Example

Pick $K$ centers (randomly):

$$(-1, -1) \text{ and } (0, 0)$$



## K-Means: Example

Calculate distance between points and those centers:

| $x_1$ | $x_2$ | $(-1,-1)$ | $(0,0)$ |
|---|---|---|---|
| 0.4 | -1.0 | 1.4 | 1.1 |
| -1.0 | -2.2 | 1.2 | 2.4 |
| -2.4 | -2.2 | 1.9 | 3.3 |
| -1.0 | -1.9 | 0.9 | 2.2 |
| -0.5 | 0.6 | 1.6 | 0.8 |
| -0.1 | 1.7 | 2.9 | 1.7 |
| 1.2 | 3.3 | 4.8 | 3.5 |
| 3.1 | 1.6 | 4.8 | 3.4 |
| 1.3 | 1.6 | 3.5 | 2.1 |
| 2.0 | 0.8 | 3.5 | 2.2 |

```
> centers <- rbind(c(-1,-1),c(0,0))
> dist1 <- apply(x,1,function(x) sqrt(sum((x-centers[1,])^2)))
> dist2 <- apply(x,1,function(x) sqrt(sum((x-centers[2,])^2)))
```

## K-Means: Example

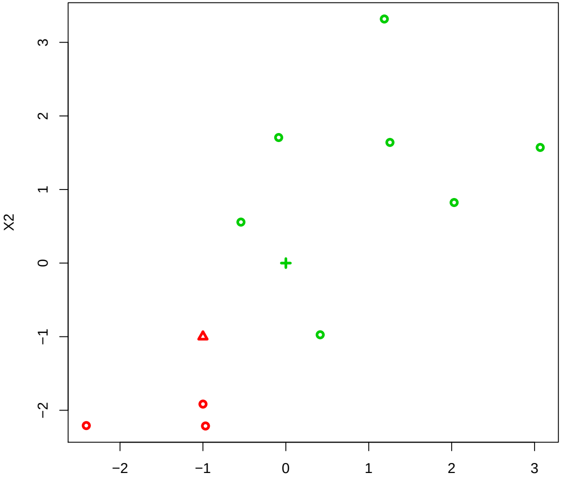
Choose mean with smaller distance:

| $x_1$ | $x_2$ | $(-1,-1)$ | $(0,0)$ |
|---|---|---|---|
| 0.4 | -1.0 | 1.4 | **1.1** |
| -1.0 | -2.2 | **1.2** | 2.4 |
| -2.4 | -2.2 | **1.9** | 3.3 |
| -1.0 | -1.9 | **0.9** | 2.2 |
| -0.5 | 0.6 | 1.6 | **0.8** |
| -0.1 | 1.7 | 2.9 | **1.7** |
| 1.2 | 3.3 | 4.8 | **3.5** |
| 3.1 | 1.6 | 4.8 | **3.4** |
| 1.3 | 1.6 | 3.5 | **2.1** |
| 2.0 | 0.8 | 3.5 | **2.2** |

```
> dists <- cbind(dist1,dist2)
> cluster.ind <- apply(dists,1,which.min)
```

## K-Means: Example

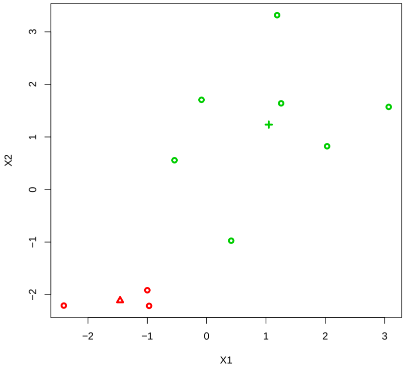
New clusters:



# K-Means: Example

Refit means for each cluster:

- cluster 1: $(-1.0, -2.2)$, $(-2.4, -2.2)$, $(-1.0, -1.9)$
- new mean: $(-1.5, -2.1)$
- cluster 2: $(0.4, -1.0)$, $(-0.5, 0.6)$, $(-0.1, 1.7)$, $(1.2, 3.3)$, $(3.1, 1.6)$, $(1.3, 1.6)$, $(2.0, 0.8)$
- new mean: $(1.0, 1.2)$



## K-Means: Example

Recalculate distances for each cluster:

| $x_1$ | $x_2$ | $(-1.5, -2.1)$ | $(1.0, 1.2)$ |
|---|---|---|---|
| 0.4 | -1.0 | 2.2 | 2.3 |
| -1.0 | -2.2 | 0.5 | 4.0 |
| -2.4 | -2.2 | 1.0 | 4.9 |
| -1.0 | -1.9 | 0.5 | 3.8 |
| -0.5 | 0.6 | 2.8 | 1.7 |
| -0.1 | 1.7 | 4.1 | 1.2 |
| 1.2 | 3.3 | 6.0 | 2.1 |
| 3.1 | 1.6 | 5.8 | 2.0 |
| 1.3 | 1.6 | 4.6 | 0.5 |
| 2.0 | 0.8 | 4.6 | 1.1 |

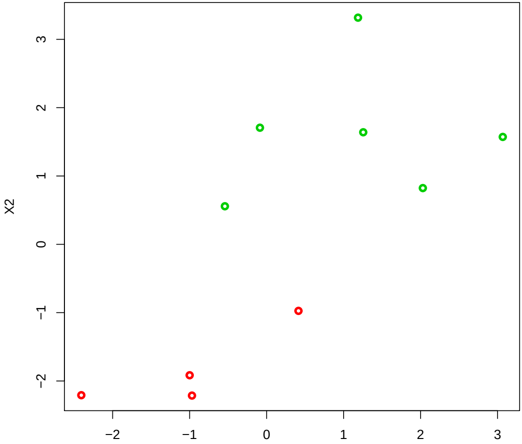
## K-Means: Example

Choose mean with smaller distance:

| $x_1$ | $x_2$ | $(-1.5, -2.1)$ | $(1.0, 1.2)$ |
|---|---|---|---|
| 0.4 | -1.0 | **2.2** | 2.3 |
| -1.0 | -2.2 | **0.5** | 4.0 |
| -2.4 | -2.2 | **1.0** | 4.9 |
| -1.0 | -1.9 | **0.5** | 3.8 |
| -0.5 | 0.6 | 2.8 | **1.7** |
| -0.1 | 1.7 | 4.1 | **1.2** |
| 1.2 | 3.3 | 6.0 | **2.1** |
| 3.1 | 1.6 | 5.8 | **2.0** |
| 1.3 | 1.6 | 4.6 | **0.5** |
| 2.0 | 0.8 | 4.6 | **1.1** |

## K-Means: Example

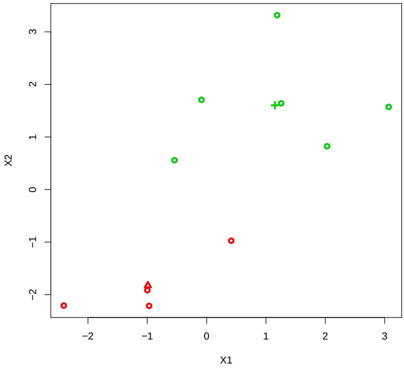
New clusters:



## K-Means: Example

Refit means for each cluster:

- cluster 1: $(0.4, -1.0)$, $(-1.0, -2.2)$, $(-2.4, -2.2)$, $(-1.0, -1.9)$
- new mean: $(-1.0, -1.8)$
- cluster 2: $(-0.5, 0.6)$, $(-0.1, 1.7)$, $(1.2, 3.3)$, $(3.1, 1.6)$, $(1.3, 1.6)$, $(2.0, 0.8)$
- new mean: $(1.2, 1.6)$



## K-Means: Example

Recalculate distances for each cluster:

| $x_1$ | $x_2$ | $(-1.0, -1.8)$ | $(1.2, 1.6)$ |
|---|---|---|---|
| 0.4 | -1.0 | 1.6 | 2.7 |
| -1.0 | -2.2 | 0.4 | 4.4 |
| -2.4 | -2.2 | 1.5 | 5.2 |
| -1.0 | -1.9 | 0.1 | 4.1 |
| -0.5 | 0.6 | 2.4 | 2.0 |
| -0.1 | 1.7 | 3.6 | 1.2 |
| 1.2 | 3.3 | 5.6 | 1.7 |
| 3.1 | 1.6 | 5.3 | 1.9 |
| 1.3 | 1.6 | 4.1 | 0.1 |
| 2.0 | 0.8 | 4.0 | 1.2 |

## K-Means: Example

Select smallest distance and compare these clusters with previous:

Table : New Clusters

| $x_1$ | $x_2$ | $(-1.0, -1.8)$ | $(1.2, 1.6)$ |
|---|---|---|---|
| 0.4 | -1.0 | **1.6** | 2.7 |
| -1.0 | -2.2 | **0.4** | 4.4 |
| -2.4 | -2.2 | **1.5** | 5.2 |
| -1.0 | -1.9 | **0.1** | 4.1 |
| -0.5 | 0.6 | 2.4 | **2.0** |
| -0.1 | 1.7 | 3.6 | **1.2** |
| 1.2 | 3.3 | 5.6 | **1.7** |
| 3.1 | 1.6 | 5.3 | **1.9** |
| 1.3 | 1.6 | 4.1 | **0.1** |
| 2.0 | 0.8 | 4.0 | **1.2** |

Table : Old Clusters

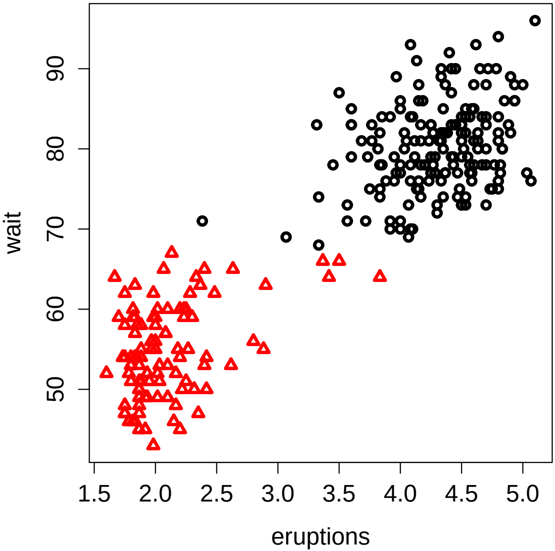
| $(-1.5, -2.1)$ | $(1.0, 1.2)$ |
|---|---|
| **2.2** | 2.3 |
| **0.5** | 4.0 |
| **1.0** | 4.9 |
| **0.5** | 3.8 |
| 2.8 | **1.7** |
| 4.1 | **1.2** |
| 6.0 | **2.1** |
| 5.8 | **2.0** |
| 4.6 | **0.5** |
| 4.6 | **1.1** |

## K-Means in Practice

`R` has a function for K-means in the `stats` package; this is probably already loaded

- let's use this for the Old Faithful data

```
> library(datasets)
> faith.2 <- kmeans(faithful,2)
> names(faith.2)
> plot(faithful[,1],faithful[,2],col=faith.2$cluster,
 +   pch=faith.2$cluster,lwd=3)
```



wait
90
80
70
60
50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions

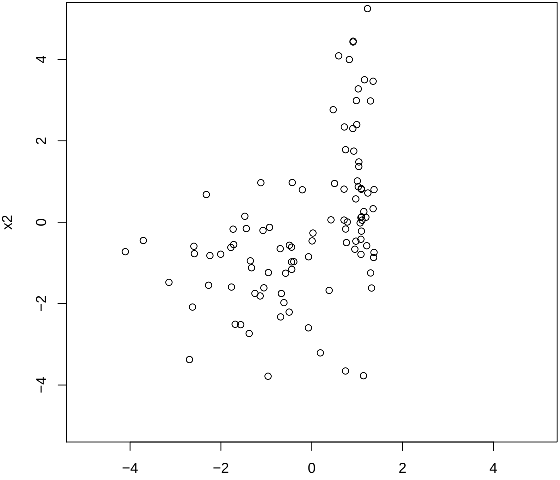
## K-Means in R

K-means can be used for *image segmentation*

- partition image into multiple segments
- find boundaries of objects
- make art

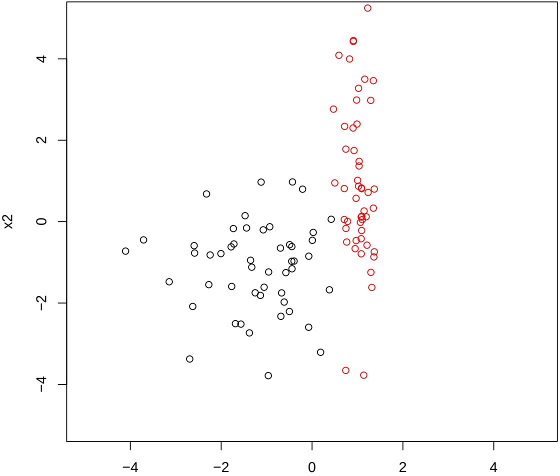
# K-Means Clustering

What is our data look like this?



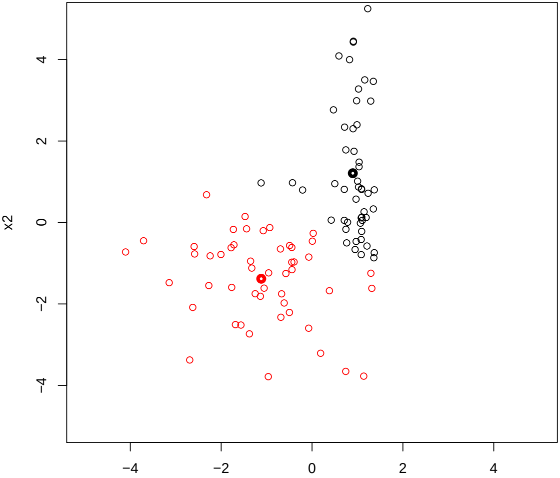
## K-Means Clustering

True clustering:



## K-Means Clustering

K-means clustering ($K = 2$):



## Mixture Models

K-means associates data with cluster centers.

What if we actually modeled the data?

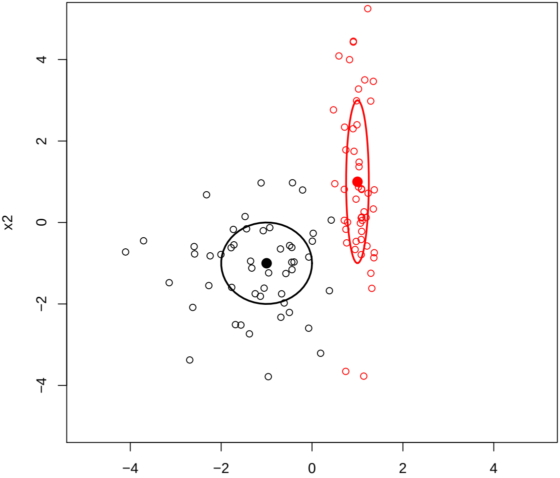
- real-valued data
- observation $\mathbf{x}_i$ in cluster $c_i$
- have $K$ clusters
- model each cluster with a Gaussian distribution

$$\mathbf{x}_i \mid c_i = k \sim N(\mu_k, \Sigma_k)$$

- $\mu_k$ is mean vector, $\Sigma_k$ is covariance matrix

## Mixture Models

Gaussian mixture model ($K = 2$):



Why mixture models?

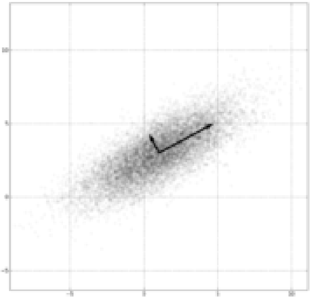
- more flexible: can account for clusters with different shapes
- have data model (will be useful for choosing $K$)
- less sensitive to data scaling

## Multivariate Gaussian

Multivariate Gaussian distribution for $\mathbf{x} \in \mathbb{R}^d$:

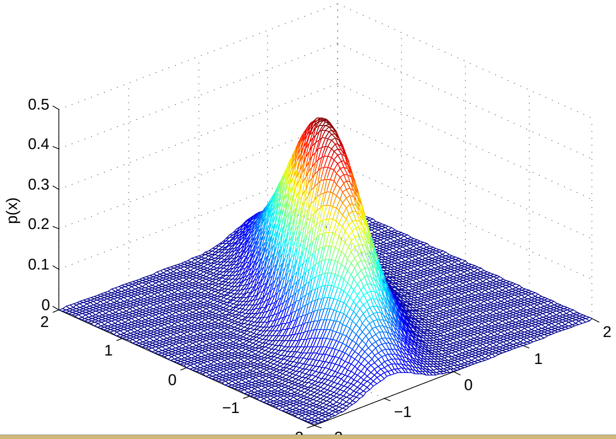
$$p(\mathbf{x} \mid \mu, \Sigma) = (2\pi)^{-\frac{d}{2}} \left|\Sigma\right|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)}$$

- $\mu$ is vector of means
- $\Sigma$ is covariance matrix

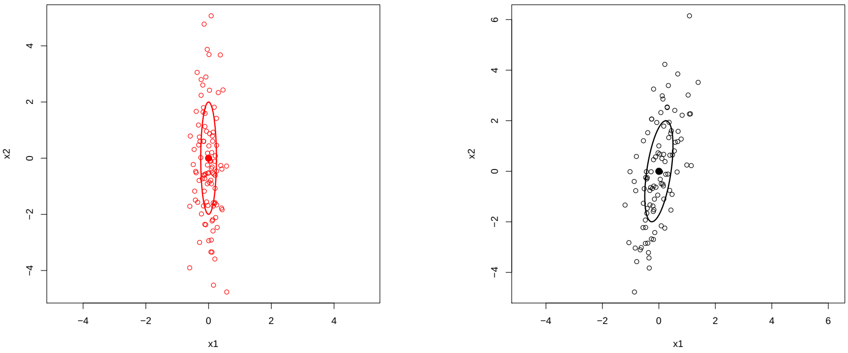


## Multivariate Gaussian

pdf when $\mu = [0, 0]$ and $\Sigma = \begin{bmatrix} 0.9 & 0.4 \\ 0.4 & 0.3 \end{bmatrix}$:



## Multivariate Gaussian



## Fitting a Mixture Model

Mixture model:

- observation $\mathbf{x}_i$ in cluster $c_i$ with $K$ clusters
- model each cluster with a Gaussian distribution

$$\mathbf{x}_i \mid c_i = k \sim N(\mu_k, \Sigma_k)$$

How do we find $c_1, \dots, c_n$ (clusters) and $(\mu_1, \Sigma_1), \dots, (\mu_K, \Sigma_K)$ (cluster centers)?

## Fitting a Mixture Model

First, let's simplify the model:

- covariance matrices have only diagonal elements,

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & \sigma_K^2 \end{bmatrix}$$

- set $\sigma_1^2 = \dots = \sigma_K^2$, suppose known

## Fitting a Mixture Model

Next, use a method similar to K-means:

- start with random cluster centers
- associate observations to clusters by (log-)likelihood,

$$\ell(\mathbf{x}_i \mid c_i = k) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log\left(\prod_{j=1}^{d}\sigma_{k,j}^2\right) - \frac{1}{2}\sum_{j=1}^{d}(x_{i,j} - \mu_{k,j})^2/\sigma_{k,j}^2$$

$$\propto -d\log(\sigma_k) - \frac{1}{2\sigma_k^2}\sum_{j=1}^{d}(x_{i,j} - \mu_{k,j})^2$$

$$\propto -\sum_{j=1}^{d}(x_{i,j} - \mu_{k,j})^2$$

- refit centers $\mu_1, \dots, \mu_K$ given clusters by

$$\mu_{k,j} = \frac{1}{n_k}\sum_{c_i = k} x_{i,j}$$

## Fitting a Mixture Model

**clustering with K-means**

minimize distance

$$d(\mathbf{x}_i, \mu_k) = \sqrt{\sum_{j=1}^{d} (x_{i,j} - \mu_{k,j})^2}$$

**clustering with GMM**

maximize likelihood

$$\ell(\mathbf{x}_i \mid c_i = k) \propto -\sum_{j=1}^{d} (x_{i,j} - \mu_{k,j})^2$$

**update means with K-means**

use average

$$\mu_{k,j} = \frac{1}{n_k} \sum_{c_i = k} x_{i,j}$$

**update means with GMM**

use average

$$\mu_{k,j} = \frac{1}{n_k} \sum_{c_i = k} x_{i,j}$$

## Fitting a Mixture Model

OK, now what if

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & \sigma_K^2 \end{bmatrix}$$

and $\sigma_1^2, \dots, \sigma_K^2$ can take different values?

- use same algorithm
- update $\mu_k$ and $\sigma_k^2$ with maximum likelihood estimator,

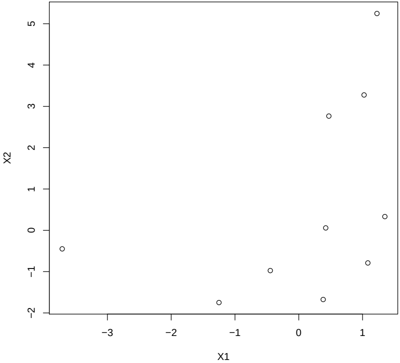
$$\mu_{k,j} = \frac{1}{n_k} \sum_{c_i = k} x_{i,j}$$

$$\sigma_{k,j}^2 = \frac{1}{n_k} \sum_{c_i = k} (x_{i,j} - \mu_{k,j})^2$$

## Fitting a Mixture Model

Data:

| $x_1$ | $x_2$ |
|---|---|
| -3.7 | -0.4 |
| 0.4 | 0.1 |
| 0.4 | -1.7 |
| -0.4 | -1.0 |
| -1.3 | -1.7 |
| 1.0 | 3.3 |
| 1.2 | 5.2 |
| 1.3 | 0.3 |
| 1.1 | -0.8 |
| 0.5 | 2.8 |



## Fitting a Mixture Model

- pick centers and variances, $\mu_1 = [-1, -1]$, $\sigma_1^2 = [1, 1]$, $\mu_1 = [1, 1]$, $\sigma_1^2 = [1, 1]$
- compute (proportional) log likelihoods,

$$\ell(\mathbf{x}_i \mid c_i = k) = -\sum_{j=1}^{d} \log(\sigma_j) - \frac{1}{2} \sum_{j=1}^{d} (x_{i,j} - \mu_{k,j})^2 / \sigma_{k,j}^2$$

| $x_1$ | $x_2$ | $k = 1$ | $k = 2$ |
|---|---|---|---|
| -3.7 | -0.4 | -3.8 | -12.1 |
| 0.4 | 0.1 | -1.6 | -0.6 |
| 0.4 | -1.7 | -1.2 | -3.8 |
| -0.4 | -1.0 | -0.2 | -3.0 |
| -1.3 | -1.7 | -0.3 | -6.3 |
| 1.0 | 3.3 | -11.2 | -2.6 |
| 1.2 | 5.2 | -22.0 | -9.0 |
| 1.3 | 0.3 | -3.6 | -0.3 |
| 1.1 | -0.8 | -2.2 | -1.6 |
| 0.5 | 2.8 | - 8.2 | -1.7 |

## Fitting a Mixture Model

- fit new means and variances:

$$\mu_1 = [-1.3, -1.2]$$
$$\sigma_1^2 = [3.1, 0.4]$$
$$\mu_2 = [0.9, 1.8]$$
$$\sigma_2^2 = [0.2, 5.4]$$

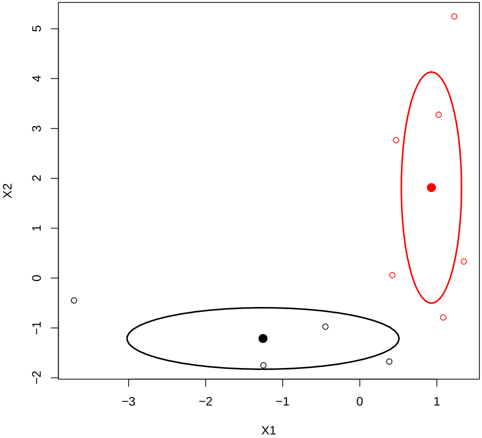
- compute new distances...

## Fitting a Mixture Model

| $x_1$ | $x_2$ | $k=1$ | $k=2$ |
|---|---|---|---|
| -3.7 | -0.4 | -1.8 | -70.8 |
| 0.4 | 0.1 | -2.7 | -1.0 |
| 0.4 | -1.7 | -0.8 | -2.0 |
| -0.4 | -1.0 | -0.3 | -6.8 |
| -1.3 | -1.7 | -0.5 | -16.6 |
| 1.0 | 3.3 | -27.4 | -0.1 |
| 1.2 | 5.2 | -55.9 | -1.3 |
| 1.3 | 0.3 | -4.3 | -0.7 |
| 1.1 | -0.8 | -1.2 | -0.6 |
| 0.5 | 2.8 | -21.3 | -0.7 |

No change, so clusters are final

# Fitting a Mixture Model



## Limitations of $k$-means / mixture models

$k$-means is fast and simple, but ...

- What if your data are discrete?
- What if each data point has more than one cluster? (digits vs. documents)
- What if you don't know the number of clusters?