Machine Learning: Chenhao Tan

University of Colorado Boulder

LECTURE 23

Slides adapted from Jordan Boyd-Graber, Chris Ketelsen

Logistics

- HW4 grading is done

Learning objectives

- Learn about general clustering
- Learn about the K-means algorithm

Quiz on PCA

Which of the following statements are true?

A. Feature scaling is not useful for PCA, since the eigenvector calculation takes care of this automatically.

B. Given an input $\boldsymbol{x} \in \mathbb{R}^n$, PCA compresses it to a lower-dimensional vector $\boldsymbol{z} \in \mathbb{R}^k$.

C. PCA can be used only to reduce the dimensionality of data by 1 (such as 3D to 2D, or 2D to 1D).

D. If the input features are on very different scales, it is a good idea to perform feature scaling before applying PCA.

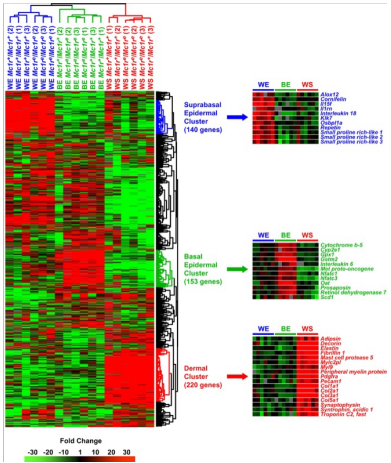
Outline

Clustering

- One important unsupervised method is clustering
- Goal: Organize data in classes

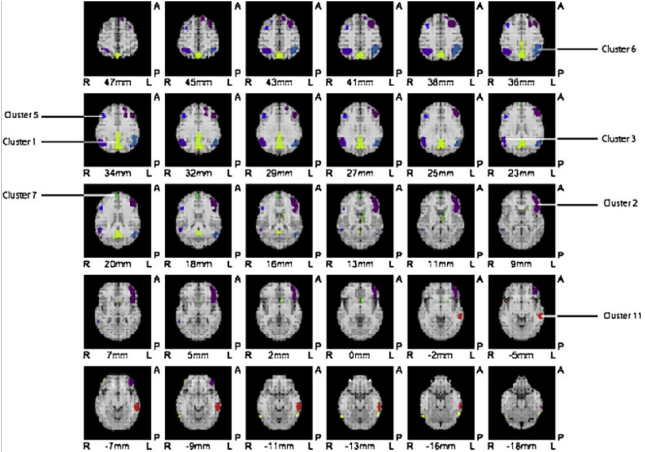
Clustering Applications

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)



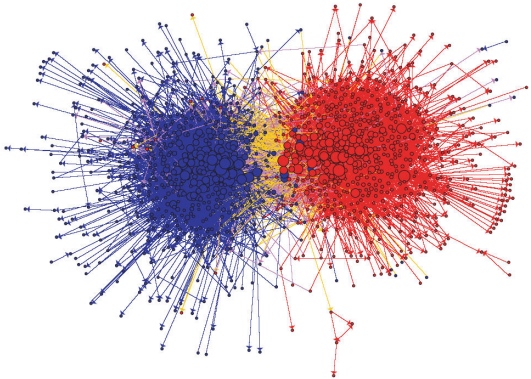
Clustering Applications

Medical imaging



Clustering Applications

Community detection



Adamic & Glance (2005)

Clustering Applications

News media

Clustering

- One important unsupervised method is clustering
- Goal: Organize data in classes
  - Classes are hard to define
  - Different data representation may lead to different clusterings

Clustering

- One important unsupervised method is clustering
- Goal: Organize data in classes
  - Data have high in-class similarity
  - Data have low out-of-class similarity

Clustering — Similarity

We'll call $d(\mathbf{x}, \mathbf{y})$ the similarity measure of $\mathbf{x}$ and $\mathbf{y}$
Examples:
Euclidean Distance: $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$
Edit Distance: $d(\mathbf{x}, \mathbf{y}) =$ `# replace, insert, deletes to turn` $\mathbf{x}$ `into` $\mathbf{y}$

$$d(\texttt{kitten}, \texttt{sitting}) = 3$$

$$\texttt{kitten} \rightarrow \texttt{sitten} \rightarrow \texttt{sittin} \rightarrow \texttt{sitting}$$

What properties make a good similarity measure?

Clustering — Similarity

We'll call $d(\mathbf{x}, \mathbf{y})$ the similarity measure of $\mathbf{x}$ and $\mathbf{y}$
Properties:

$$\begin{aligned} \text{Symmetry} \quad & d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}) \\ \text{Self-Consistency} \quad & d(\mathbf{x}, \mathbf{x}) = 0 \\ \text{Positivity} \quad & d(\mathbf{x}, \mathbf{y}) = 0 \text{ iff } \mathbf{x} = \mathbf{y} \\ \text{Triangle Inequality} \quad & d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y}) \end{aligned}$$

OK, so say we have a good similarity measure.
How do we find clusters in the data?

Outline

K-means

- Simplest clustering method
- Iterative in nature
- Reasonably fast
- Very popular in practice (though with more bells and whistles)
- Requires real-valued data

K-means

General Idea:
pick K initial cluster means
do until convergence ...

- associate examples closest to mean k with cluster k
- update cluster means with current examples in cluster k

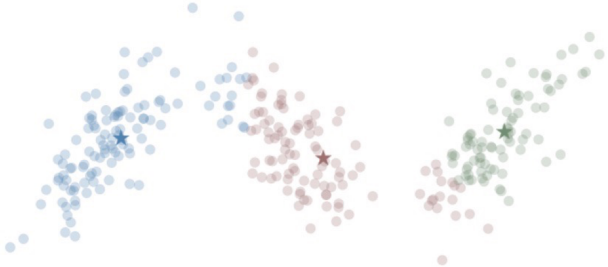
Stop when:

- cluster assignments don't change
- cluster means don't change (too much)

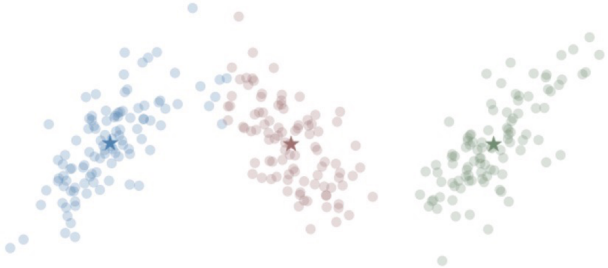
K-means example

K-means example

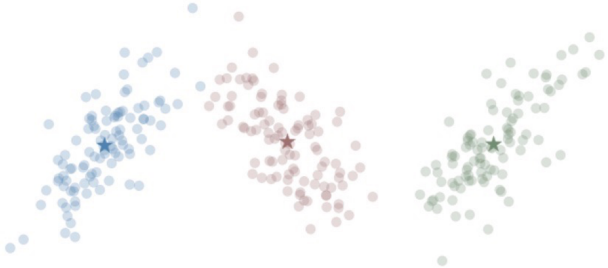
K-means example



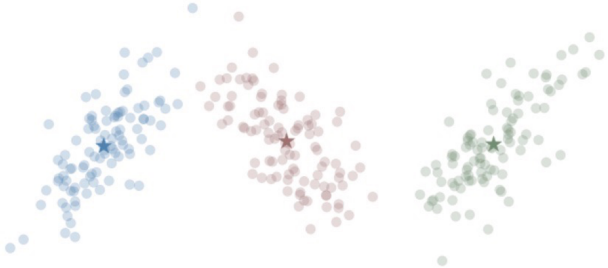
K-means example



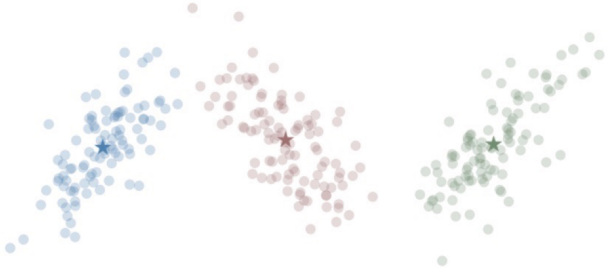
K-means example



K-means example



K-means example



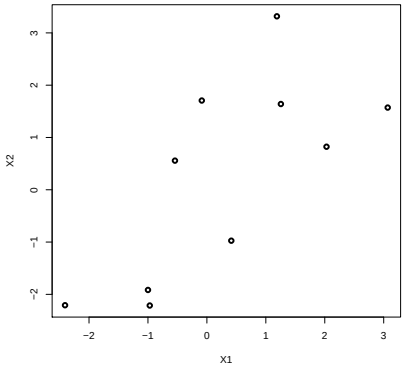
K-means example

Animations: `http://shabal.in/visuals/kmeans/4.html`

K-means example in numbers

Data:

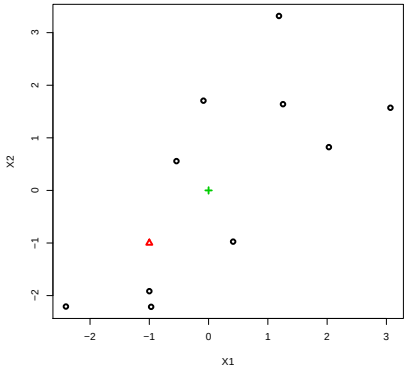
x_1	x_2
0.4	-1.0
-1.0	-2.2
-2.4	-2.2
-1.0	-1.9
-0.5	0.6
-0.1	1.7
1.2	3.3
3.1	1.6
1.3	1.6
2.0	0.8



K-means example in numbers

Pick K centers (randomly):

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



K-means example in numbers

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

K-means example in numbers

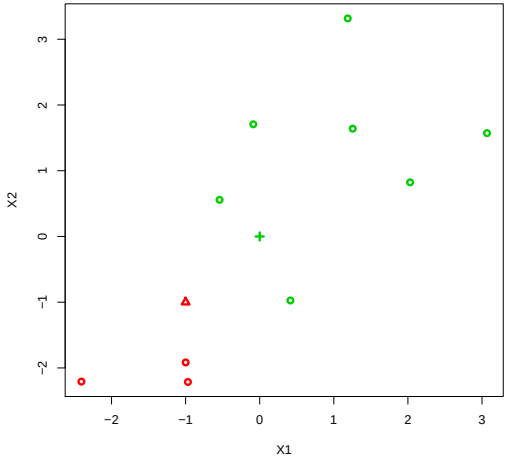
Choose mean with smaller distance:

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)
```

K-means example in numbers

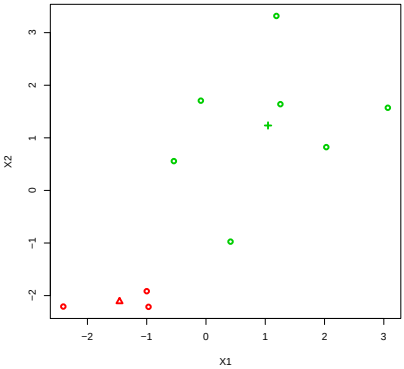
New clusters:



K-means example in numbers

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 in numbers

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

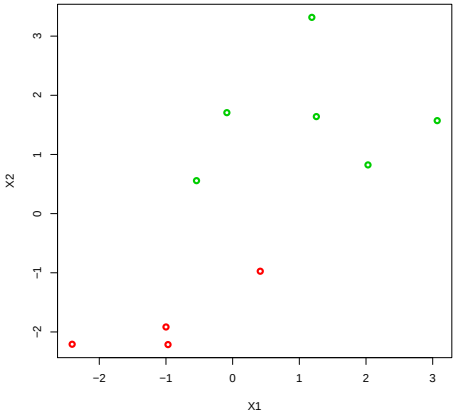
K-means example in numbers

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

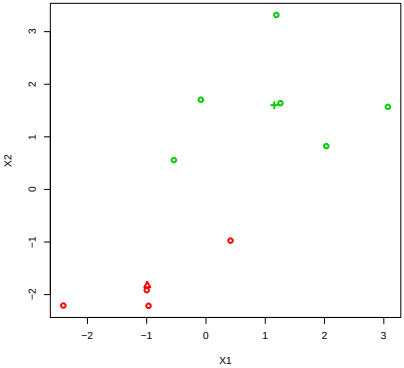
New clusters:



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 in numbers

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 in numbers

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

Strengths:

- Simple to understand
- Efficient - time complexity $\mathcal{O}(dNKT)$ for $\mathbf{x} \in \mathbb{R}^d$
- Simple to implement

K-means

```
def KMeans(X, K, max_it=500):

    # Initialize cluster means to K samples from data
    rstart = choice(range(X.shape[0]), size=(K), replace=False)
    mu, muold = X[rstart,:], 1000*np.ones(X[rstart,:].shape)

    its = 0
    while its < max_it and np.linalg.norm(mu-muold) > 1e-4 :

        # compute distance b/w each point and centroid
        dist = np.array([[np.linalg.norm(x-m) for m in mu] for x in X]

        # compute new cluster assignments
        z = np.array([np.argmin(d) for d in dist])

        # move centroids
        muold = mu
        mu = np.array([np.mean(X[z==k, :], axis=0) for k in range(K)])
        its += 1

    return mu, z
```

K-means

Weaknesses:

- Doesn't really work with categorical data
- Usually only converges to local minimum
- Have to determine number of clusters
- Can be sensitive to outliers
- Only generates convex clusters

K-means weaknesses

Doesn’t really work with categorical data

K-means weaknesses

Doesn't really work with categorical data
Fix: Do K-modes instead

K-means weaknesses

Usually only converges to local minimum

K-means weaknesses

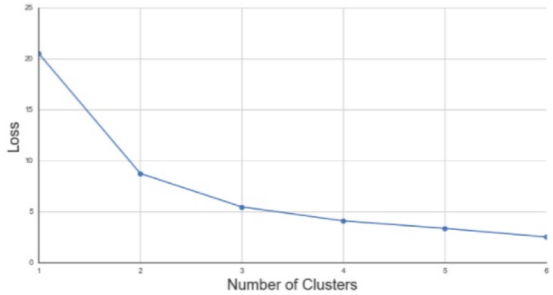
Usually only converges to local minimum
Fix: Do several runs with random initializations and choose best

K-means weaknesses

Have to determine number of clusters

K-means weaknesses

Have to determine number of clusters
Fix: Use the elbow method.
Run K-Means for different values of k and look at loss function



K-means weaknesses

Sensitive to outliers

K-means weaknesses

Sensitive to outliers

K-means weaknesses

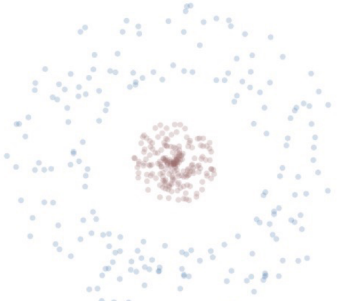
Sensitive to outliers

K-means weaknesses

Sensitive to outliers

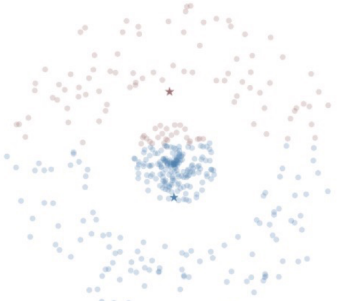
K-means weaknesses

Convex clusters



K-means weaknesses

Convex clusters



Wrap up

- Clustering requires additional assumptions to identify classes
- K-means represent a simple yet strong clustering method