CIS 490 Machine Learning

Lecture 14

Instructor: (Julia) Hua Fang

2

Last Time
Attn: We are entering
Unsupervised Learning

- Principal Components Analysis (PCA)

PCA: we are interested in **variance**

- Quick review of Exam I

Adapted from James, Witten, Hastie, Tibshirani, Friedman, Howbert, Sontag

Outline

- Unsupervised Learning: **Clustering**
  - K-means: you are expected to know
    - ✓ Distance measure
    - ✓ Objective function
    - ✓ Algorithm, how to choose K clusters and application areas
    - ✓ K-means issues and remedies
  - Run K-means in R Studio
- Final Project Proposal

K-means

K-means: overview

- K-means is a clustering method that aims to find the positions/centroids (center points, or means), μ_i, i=1…k of the clusters that minimize the ***distance*** from the data points to the cluster.
- The K-means clustering uses the square of the Euclidean distance

K-means: The Objective Function

The K-means objective function

- Let $\mu_1, \ldots, \mu_K$ be the K cluster centroids (means)
- Let $r_{nk} \in \{0, 1\}$ be indicator denoting whether point $\mathbf{x}_n$ belongs to cluster k
- K-means objective minimizes the total distortion (sum of distances of points from their cluster centers)

$$J(\mu, r) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \mu_k||^2$$

K-means algorithm (Lloyd, 1957)

- **Input:** N examples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ ($\mathbf{x}_n \in \mathbb{R}^D$); the number of partitions K
- **Initialize:** K cluster centers $\mu_1, \ldots, \mu_K$. Several initialization options:
  - Randomly initialized anywhere in $\mathbb{R}^D$ (called Random partition)
  - Choose any K examples as the cluster centers (called Forgy)
- **Iterate:**
  - Assign each of example $\mathbf{x}_n$ to its closest cluster center

$$\mathcal{C}_k = \{n : \quad k = \arg\min_k ||\mathbf{x}_n - \mu_k||^2\}$$

  ($\mathcal{C}_k$ is the set of examples closest to μ_k)
  - Recompute the new cluster centers μ_k (mean/centroid of the set $\mathcal{C}_k$)

$$\mu_k = \frac{1}{|\mathcal{C}_k|} \sum_{n \in \mathcal{C}_k} \mathbf{x}_n$$

  - Repeat while not converged
- A possible convergence criteria: cluster centers do not change anymore

Simplified Pseudo Code

```
Select K points as the initial centroids.
repeat
  Form K clusters by assigning all points to the closest centroid.
  Recompute the centroid of each cluster.
until The centroids don't change
```

K-means algorithm: Working with 2-cluster example

As a simple illustration of a k-means algorithm, consider the following data set consisting of the scores of two variables/attributes/features on each of seven individuals:

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Which pairs should be chosen as the initial centroids?

K-means algorithm: Working with a 2-cluster example

As a first step in finding a sensible initial partition, let the A & B values of the two individuals furthest apart (using the Euclidean distance measure), define the initial cluster "means", giving:

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector (initial centroid)
Cluster 1	1	(1.0, 1.0)
Cluster 2	4	(5.0, 7.0)

Answer: Sqrt $[(1-5)^2 + (1-7)^2]$ is larger than the Euclidean distance of any other pairs

K-means algorithm: Working with 2-cluster example (In class exercise)

Remaining individuals examined in sequence and allocated to the cluster to which they are closest, in terms of Euclidean distance to the cluster mean

Data

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector (initial centroid)
Cluster 1	1	*3Œ .#3Œ+#
Cluster 2	4	*7Œ .#9Œ+#

Step	Individual	Mean Vector (centroid)	Individual	Mean Vector (centroid)
1	1	(1.0, 1.0)	4	$\boxed{(5.0, 7.0)}$
2	1, 2	$\boxed{(1.2, 1.5)}$	4	$\boxed{(5.0, 7.0)}$
3	1, 2, 3	$\boxed{(1.8, 2.3)}$	4	$\boxed{(5.0, 7.0)}$
4	1, 2, 3	(1.8, 2.3)	4, 5	(4.2, 6.0)
5	1, 2, 3	(1.8, 2.3)	4, 5, 6	$\boxed{(4.3, 5.7)}$
6	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	$\boxed{(4.1, 5.4)}$

1. How to get these updated centroids?
2. Why is the initial centroid constant?

K-means algorithm: Working with 2-cluster example (In class exercise)

Remaining individuals examined in sequence and allocated to the cluster to which they are closest, in terms of Euclidean distance to the cluster mean

Data

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector (initial centroid)
Cluster 1	1	(1.0, 1.0)
Cluster 2	4	(5.0, 7.0)

Step	Individual	Mean Vector (centroid)	Individual	Mean Vector (centroid)
1	1	(1.0, 1.0)	4	(5.0, 7.0)
2	1, 2	(1.2, 1.5)	4	(5.0, 7.0)
3	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
4	1, 2, 3	(1.8, 2.3)	4, 5	(4.2, 6.0)
5	1, 2, 3	(1.8, 2.3)	4, 5, 6	(4.3, 5.7)
6	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	(4.1, 5.4)

1. How to get these updated centroids?
   Answer: For each new member, calculate and compare its Euclidean distances to cluster 1 and cluster 2 centroids, respectively, and assign its membership to the cluster with a shorter distance. Then update the centroids based on the average of included members. E.g. Subject 2 is assigned to cluster 1, then, initial centroids are updated as (1.2, 1.5), where (1.5+1)/2 =1.25; (2+1)/2 = 1.5

2. Why is the initial centroid constant for cluster 2?
   Answer: The algorithm completes all pair-wise distance calculation for cluster 1; then, move on the

K-means algorithm: Working with 2-cluster example

- Now the initial partition has changed, and the two clusters at this stage having the following characteristics:

	Individual	Mean Vector (initial centroid)
Cluster 1	1	(1.0, 1.0)
Cluster 2	4	(5.0, 7.0)

	Individual	Mean Vector (updated centroid)
Cluster 1	1,2,3	(1.8, 2.3)
Cluster 2	4,5,6,7	(4.1, 5.4)

K-means algorithm: Working with 2-cluster example (In class exercise)

Unsure that each individual has been assigned to the right cluster.

- So, additional step: compare each individual's distance to its own updated cluster mean and to that of the opposite cluster.

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector (updated centroid)
Cluster 1	1,2,3	(1.8, 2.3)
Cluster 2	4,5,6,7	(4.1, 5.4)

Individual	Distance to mean (centroid) of Cluster 1	Distance to mean (centroid) of Cluster 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

Subject 3: (3.0, 4.0)

- Euclidean distance to Cluster 1 centroid (1.8, 2.3):

Sqrt ((3-1.8)^2 + (4-2.3)^2) = sqrt (1.44 + 2.89) = 2.1;

- Euclidean distance to Cluster 2 centroid (4.1, 5.4):

Sqrt ((3-4.1)^2 + (4-5.4)^2) = sqrt (1.21 + 1.96) = 1.8;

2.1 > 1.8

So, Subject 3 need to be reassigned to cluster 2.

K-means algorithm: Working with 2-cluster example

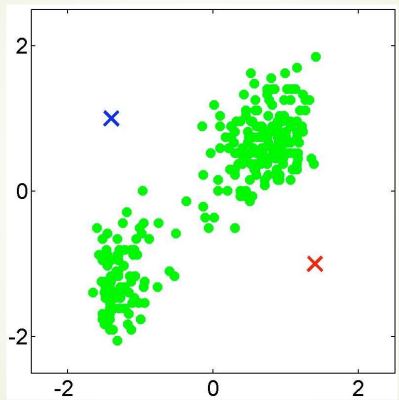
Individual 3 is relocated to Cluster 2 resulting in the new partition:

	Individual	Mean Vector (centroid)
Cluster 1	1, 2	(1.3, 1.5)
Cluster 2	3, 4, 5, 6, 7	(3.9, 5.1)

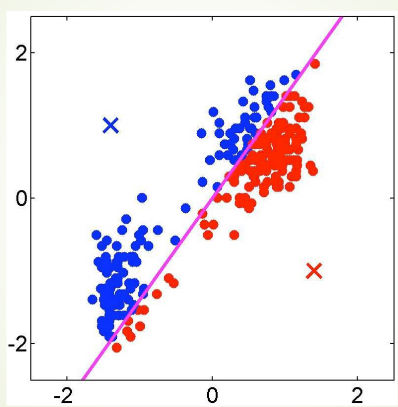
The iterative relocation would now continue from this new partition until no more relocations occur.

K -means: Initialization (assume K = 2)

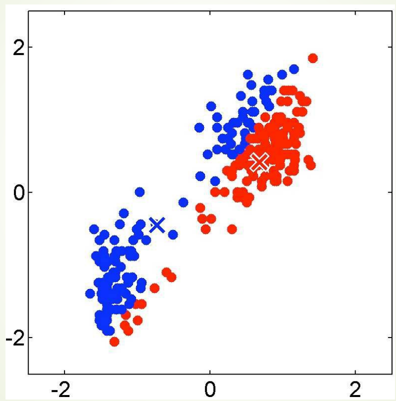
Initialization:
Random partition



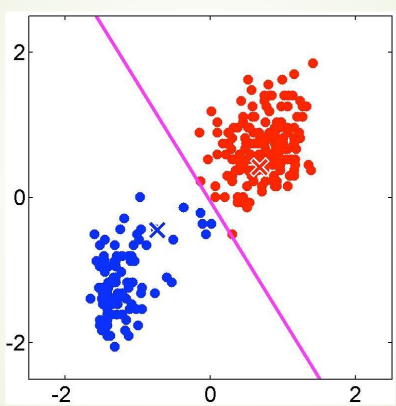
K -means iteration 1: Assigning points



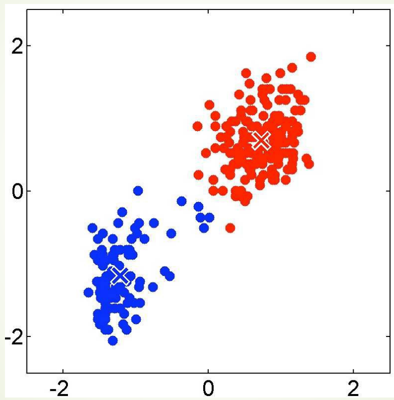
K -means iteration 1: Recomputing the cluster centers



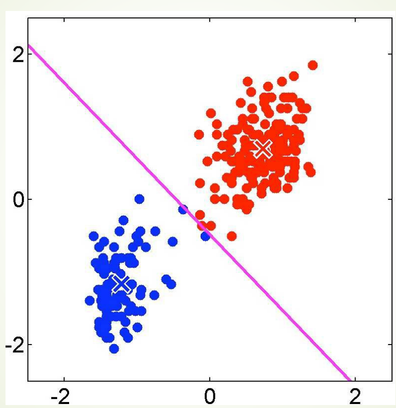
K -means iteration 2: Assigning points



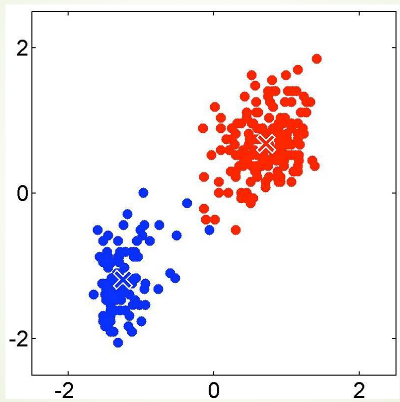
K -means iteration 2: Recomputing the cluster centers



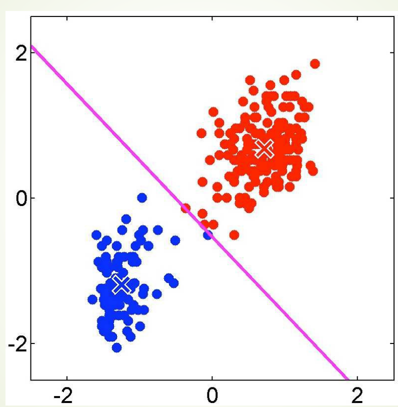
K -means iteration 3: Assigning points



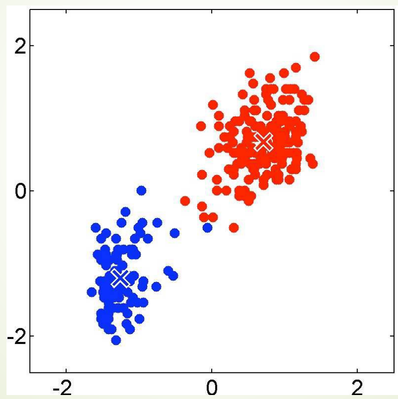
K -means iteration 3: Recomputing the cluster centers



K -means iteration 4: Assigning points



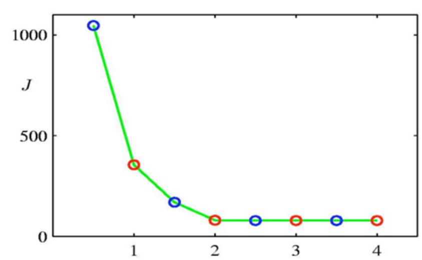
K -means iteration 4: Recomputing the cluster centers



K-means: **How to choose K**

- One way to select K for the K-means algorithm is to try different values of K, plot the K-means objective versus K, and look at the "elbow-point" in the plot

$$J(\mu, r) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \mu_k||^2$$

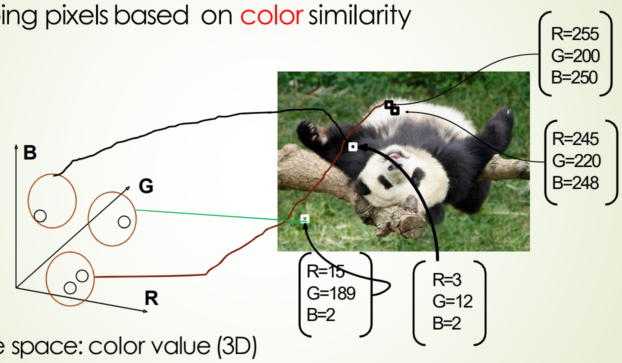


- For the above plot, $K = 2$ is the elbow point

See "Silhouette Analysis" for choosing K in the instruction file called "R_Kmeans_S22.docx"

K-means **Application**: Image Segmentation (1)

- Depending on what we choose as the *feature space*, we can group pixels in different ways.
- Grouping pixels based on color similarity



Feature space: color value (3D)

Fei-Fei Li   Lecture 13 -27

Slide credit: Kristen Grauman

Kmeans Application: Image Segmentation (2)

- K-means clustering based on intensity or color is essentially vector quantization of the image attributes
  - Clusters don't have to be spatially coherent

Image   Intensity-based clusters   Color-based clusters

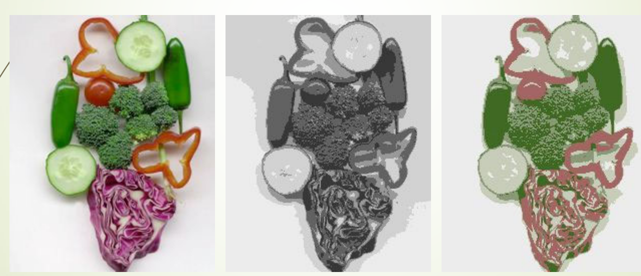


Image source: Forsyth & Ponce

Fei-Fei Li   29

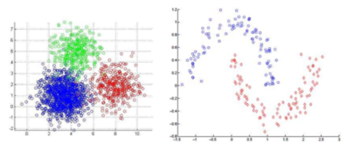
Suppl.: Similarity measures

- Different similarity criteria can lead to different clustering results

- Choice of the **similarity measure** is very important for clustering
- Similarity is inversely related to distance

Z is μ

- Different ways exist to measure distances. Some examples:
  - Euclidean distance: $d(\mathbf{x},\mathbf{z}) = ||\mathbf{x}-\mathbf{z}|| = \sqrt{\sum_{d=1}^{D}(x_d - z_d)^2}$
  - Manhattan distance: $d(\mathbf{x},\mathbf{z}) = \sum_{d=1}^{D}|x_d - z_d|$
  - Kernelized (non-linear) distance: $d(\mathbf{x},\mathbf{z}) = ||\phi(\mathbf{x}) - \phi(\mathbf{z})||$



  - For the left figure above, Euclidean distance may be reasonable
  - For the right figure above, kernelized distance seems more reasonable

K-means issues and remedies

K-means: Initialization issues

K-means is extremely sensitive to cluster center initialization

Recall: What initialization methods do K-means apply?

Bad initialization can lead to
- Poor convergence speed
- Bad overall clustering

See remedies in next slide

K-means: Initialization Remedies

Safeguarding measures:

- Choose first center as one of the examples, second which is the farthest from the first, third which is the farthest from both, and so on.
- Try multiple initializations and choose the best result
- Other smarter initialization schemes (e.g., Bisecting K-means; look at the K-means++ algorithm by Arthur and Vassilvitskii)

K-means: Sensitive to outliers

Sensitive to outliers (such examples can affect the mean by a lot)

K-means: Sensitive to outliers and remedies

K-medians algorithm is a more robust alternative for data with outliers Reason.

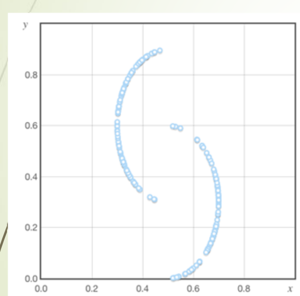
K-means: Assumptions

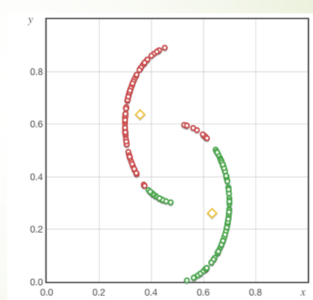
The k-means algorithm works reasonably well when the data fits the cluster model:

- **The clusters are spherical:** the data points in a cluster are centered around that cluster
- **The spread/variance/density/size of the clusters is similar:** Each data point belongs to the closest cluster

K-Means Example Assumption 1:

Issue*:* when the clusters are not spherical

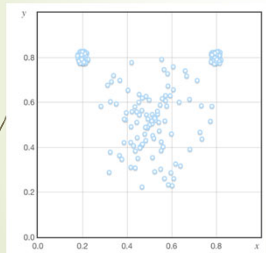


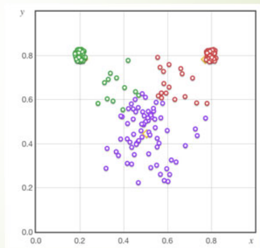


Suppl. Remedies: try other unsupervised learning methods, eg. Spectral clustering; and a different similarity measure, e.g. kernelized distance.

K-Means Example Assumption 2:

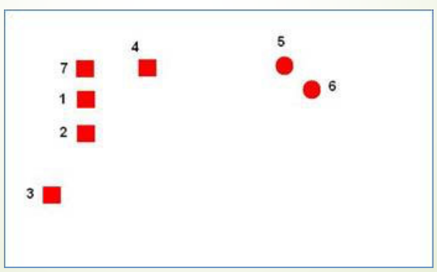
Issue: the clusters have different density/size/variance





Suppl. Remedies: try other unsupervised learning methods, eg. Gaussian Mixture Models

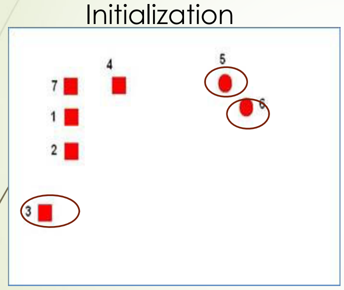
K-Means: Issue --Empty Cluster Example



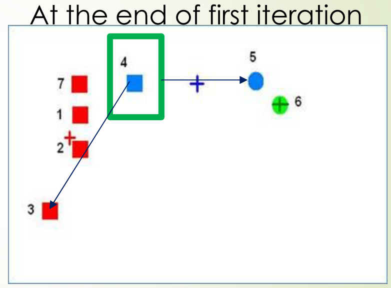
K-Means: Empty Cluster Example

- Consider the following example for which we want the number of clusters to be 3. The shapes of the points have no meaning for now.
- Using Forgy initialization, let us assume that we have chosen points 3, 5, and 6 as our initial cluster centers.

Initialization



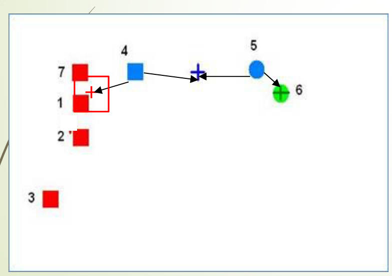
At the end of first iteration



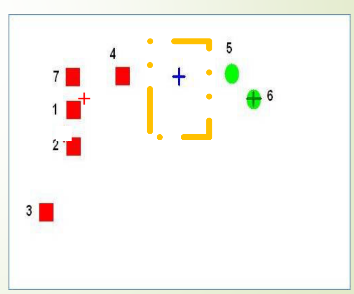
- Note that the distance between 4 and 3 is larger than the distance between 4 and 5, and so 4 is assigned to the cluster represented by 5. Remember 5 is one of initial centroids
- Before the 2nd iteration, we update the cluster centers and the picture shows the centroids and the clusters at the end of first step: Points 3, 1, 2, and 7 are in red cluster; 4 and 5 in blue cluster; 6 in green cluster.

K-Means: Empty Cluster Example

- In the next iteration point 4 will decide that it is closer to the red cluster and point 5 will decide that it is closer to the green cluster.



Next iteration



This will cause blue cluster to be empty.

K Means: How to handle empty clusters (remedies)

- Choose the point that contributes most to SSE as centroid
- Choose a point from the cluster with the highest SSE as centroid
- If there are several empty clusters, the above can be repeated several times.

Suppl.: K-means++

- prevent arbitrarily bad local minima?
  1. Randomly choose first center.
  2. Pick new center with prob. proportional to $(x_i - \mu)^2$
     – (Contribution of x to total error)
  3. Repeat until K centers.
- Expected error O(logK) (optimal)

Refer to Arthur, D. and Vassilvitskii, S. (2007) K-Means++: The Advantages of Careful Seeding Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, Philadelphia, January 2007, 1027-1035.

41

R: Running Kmeans in R studio.

R: Run Kmeans in Rstudio using Iris data

Let's go through the instruction file "R_Kmeans_S22.docx" posted with LS14 slides at myCourses.

42

Final Project Proposal: Slide submission due April 6; Presentation in class April 7

- Project goal/motivation/application area: First, determine if you are working on supervised or unsupervised learning
- Pick **one** dataset:

Note: Don't use datasets you picked for sectional projects or any data examples mentioned or used in class. Depending on what methods you are going to compare, pick your appropriate dataset.

- What specific supervised/unsupervised learning methods you will use and why you think they are appropriate for your application.

Suggestions:

- If picking supervised, first decide regression or classification, and then choose either specific regression (e.g., ridge, lasso, regression tree, etc.) or classification methods (e.g., logistic, classification tree, naïve Bayes, etc).

(Advanced methods from posted reading materials or supplementary materials are optional but not required, e.g., bagging, boosting, random forests, neural net, etc.)

43

Final Project Proposal: Continued

- If picking unsupervised, decide dimension reduction or clustering:

Note: for final projects you are encouraged to focus on clustering methods.

- Pick **two** methods: Run on the same dataset and compare them by checking all possible evaluation metrics (and graphics).
  - Refer to instruction files and slides for each method introducted
  - What evaluation methods/metrics/graphics you are considering (as detailed as possible): Generic indices and specific ones for chosen methods.
- Cite references
- Submission: no need to attach any code; tables, graphs and flow chart can be used for your proposed work); only need ~ 10 slides.

Warning: Don't compare apples with oranges