18-661 Introduction to Machine Learning

Clustering, Part I

Spring 2025

ECE – Carnegie Mellon University

Announcements

- HW 4 is due on Wednesday, April 16.
- The final exam is scheduled for 1:00pm-4:00 pm ET on Friday, May 2. Please let us know by April 15 if you cannot take the exam at this time (more than 2 exams starting within a 24 hour period or a direct time conflict).
- No recitation this Friday (enjoy Carnival!)

Outline

1. Clustering

2. K-means

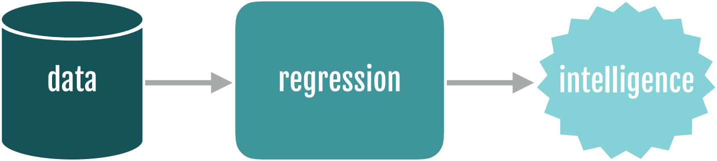
3. K-means$++$

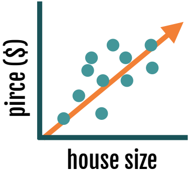
4. Gaussian Mixture Models

Clustering

Supervised Learning: Regression

How much should you sell your house for?

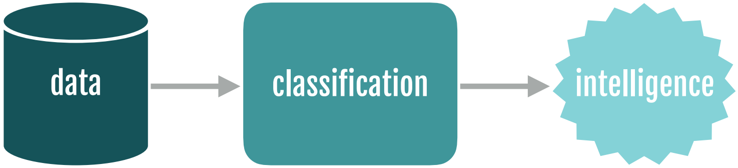


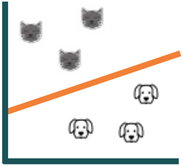


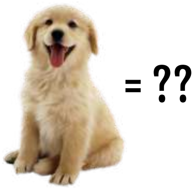
input: houses & features **learn**: $x \to y$ relationship **predict**: y (*continuous*)

Supervised Learning: Classification

Cat or dog?







input: cats and dogs

learn: $x \rightarrow y$ relationship

predict: y (*categorical*)

Supervised versus Unsupervised Learning

Supervised Learning: labeled observations $\{(\boldsymbol{x}_1, y_1), \ldots (\boldsymbol{x}_n, y_n)\}$

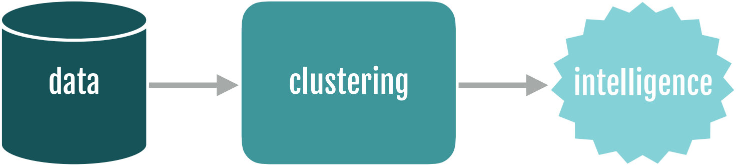
- Labels 'teach' algorithm to learn mapping from observations to labels
- Examples: Classification (Logistic Reg., SVMs, Neural Nets, Nearest Neighbors, Decision Trees), Regression (Linear Reg., Neural Nets)

Unsupervised Learning: unlabeled observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$

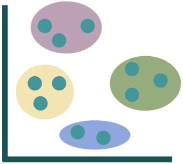
- Learning algorithm must find patterns from features alone
- Can be goal in itself (discover hidden patterns, exploratory analysis)
- Can be means to an end (pre-processing for supervised task)
- Examples:
    - K-means clustering (today), Gaussian Mixture Models (next lecture)
    - Dimensionality Reduction: Transform an initial feature representation into a more concise representation

Clustering

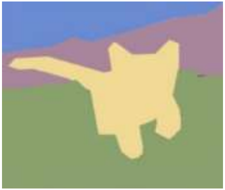
How to segment an image?



input: raw pixels $\{x\}$

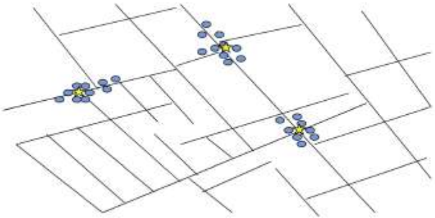


separate: $\{x\}$ into sets



output: cluster labels $\{z\}$

History of Clustering?

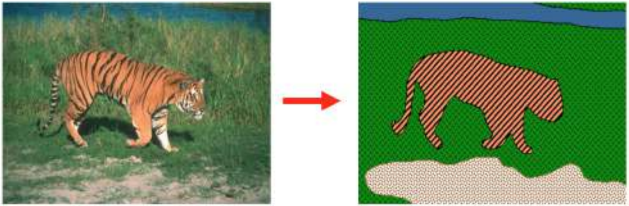


- John Snow, a London physician plotted the location of cholera deaths on a map during an outbreak in the 1850s.
- The locations indicated that cases were clustered around certain intersections where there were polluted wells – thus exposing both the problem and the solution.

Clustering Objective

- Consider a set of training data points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. How do we "cluster" these points into different groups based on their similarity?
- More formally, assign one of K labels $1, 2, \ldots, K$ to each point such that points with label k are "similar" to each other.

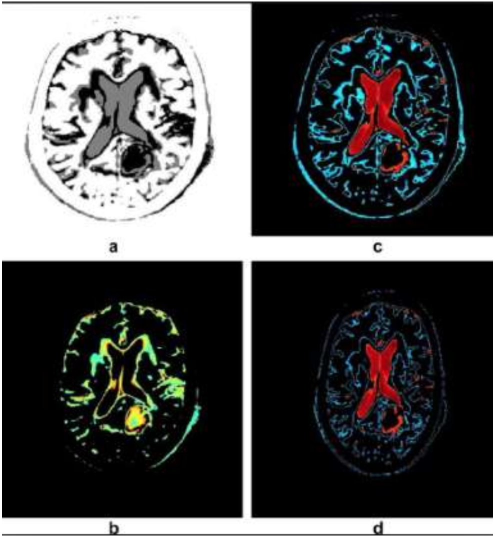
Image segmentation into foreground and background



- Cluster pixels (points) by color (orange, black, brown, green, blue).
- Naturally segments the image into foreground and background.

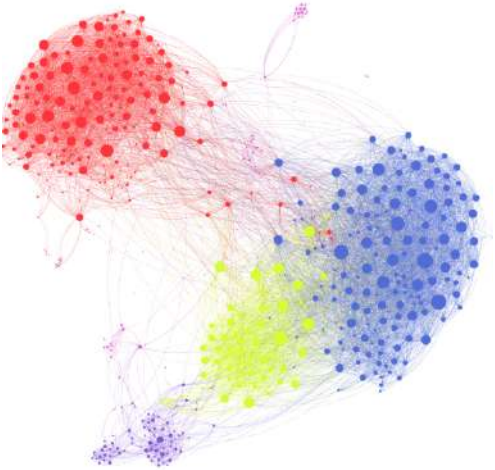
More Examples

Detecting brain lesions from MRI Scans



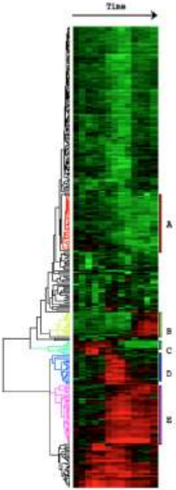
More Examples

Social network analysis



More Examples

Clustering gene expression data



Clustering

Today we will cover two methods for clustering

- K-means
- K-means$++$

K-means

K-means

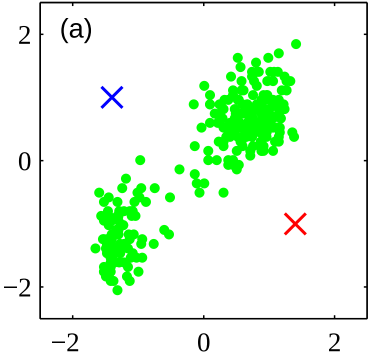
K-means: an iterative clustering method

High-level idea:

- Initialize: Pick k random points as cluster centers, $\{\mu_1, \ldots, \mu_k\}$
- Alternate:
    1. Assign data points to closest cluster center in $\{\mu_1, \ldots, \mu_k\}$
    2. Change each cluster center to the average of its assigned points
- Stop: When the clusters are stable

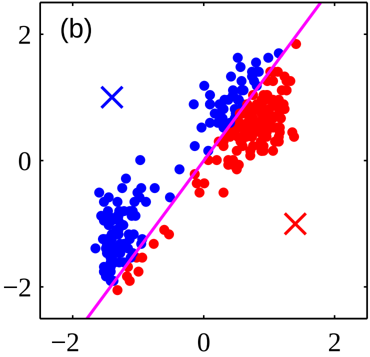
K-means Example

- Initialize: Pick k random points as cluster centers
- (Shown here for k=2)



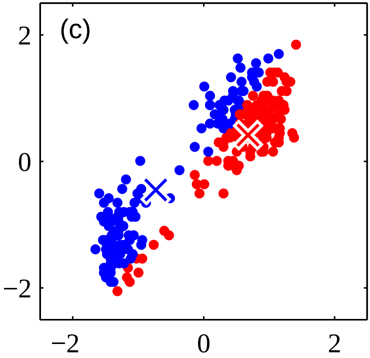
K-means Example

- Alternating Step 1: Assign data points to closest cluster center



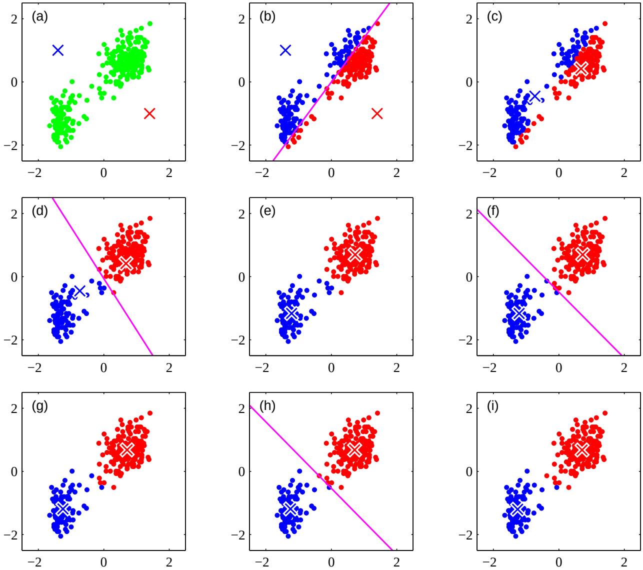
K-means Example

- Alternating Step 2: Change the cluster center to the average of the assigned points



Then: Repeat . . .

K-means Example (Several Iterations)



K-means Clustering: Details

Intuition: Data points assigned to cluster k should be near prototype $\boldsymbol{\mu}_k$

Distortion measure: (clustering objective function, cost function)

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 = \sum_{k=1}^{K} \underbrace{\sum_{n:A(\boldsymbol{x}_n)=k} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2}_{\text{spread within the } k\text{th cluster}}$$

where $r_{nk} \in \{0, 1\}$ is an indicator variable

$$r_{nk} = 1 \quad \text{if and only if} \quad A(\boldsymbol{x}_n) = k$$

How to measure distortion?

- Distance measure: $\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$ calculates how far $\boldsymbol{x}_n$ is from the cluster center $\boldsymbol{\mu}_k$
- Canonical example is the 2-norm, i.e., $\|\cdot\|_2^2$, but could be some other distance measure!

Optimization Algorithm

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

- What are the variables that we need to optimize? $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$
- Difficult to jointly optimize both
- Solution: Alternate optimization between $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$

- **Step 0** Initialize $\{\boldsymbol{\mu}_k\}$ to some values
- **Step 1** Fix $\{\boldsymbol{\mu}_k\}$ and minimize over $\{r_{nk}\}$, to get this assignment:

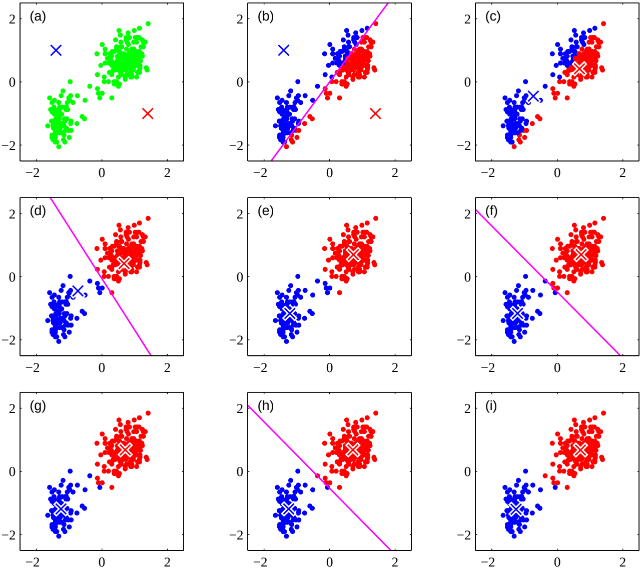
$$r_{nk} = \begin{cases} 1 & \text{if } k = \text{argmin}_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise} \end{cases}$$

- **Step 2** Fix $\{r_{nk}\}$ and minimize over $\{\boldsymbol{\mu}_k\}$ to get this update:

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \boldsymbol{x}_n}{\sum_n r_{nk}}$$

- **Step 3** Return to Step 1 unless stopping criterion is met

K-means Example (Alternate Updates to r_{nk} and μ_k)



Properties of the K-means Algorithm

Does it converge?

- **Guaranteed to converge in a finite number of iterations**
    - Key idea: K-means is an alternating optimization approach
    - Each step is guaranteed to decrease the objective/cost function—thus guaranteed to converge
    - *However*, may converge to a *local minimum* (objective is non-convex)

What's the runtime?

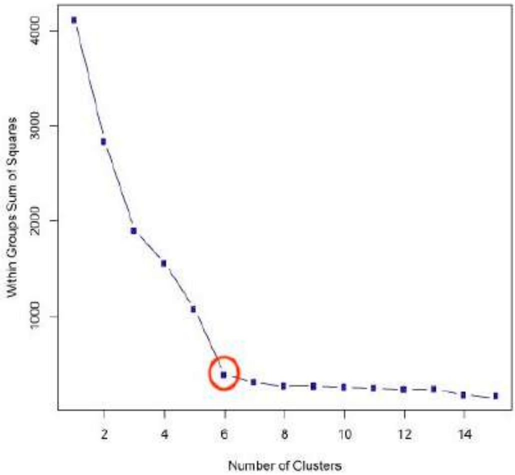
- **Running time per iteration**:
    - Assume: n data points, each with d features, and k clusters
    - Assign data points to closest cluster: $O(ndk)$
    - Re-compute cluster centers: $O(ndk)$
- **Thus, total runtime is**: $O(ndki)$, where i is the number of iterations

Practical Issues with K-means

- How to select k?
    - Prior knowledge
    - Heuristics (e.g., elbow method)

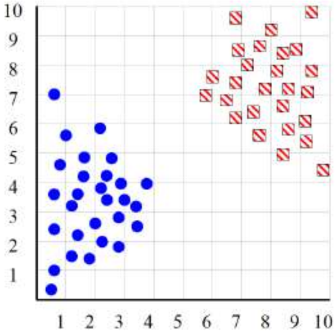
Elbow Method

Select a small value of *k* such that adding a new cluster doesn't reduce the within-cluster distances much

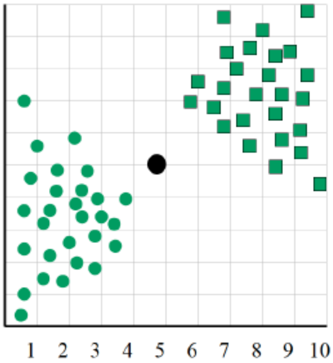


Elbow Method

How can we find the right number of clusters? Track the objective function as we increase k!



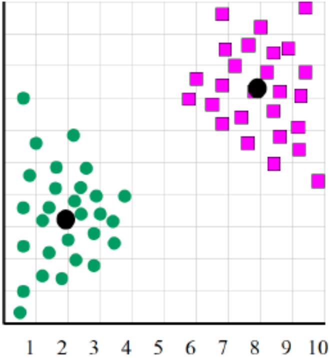
When $k = 1$, objective value is 873.



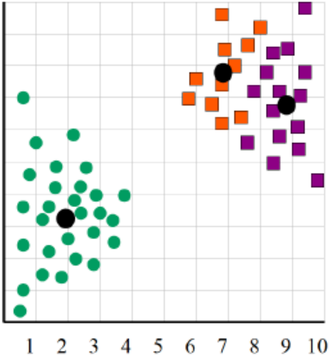
Elbow Method

How can we find the right number of clusters? Track the objective function as we increase k!

When $k = 2$, objective value is 173.1.

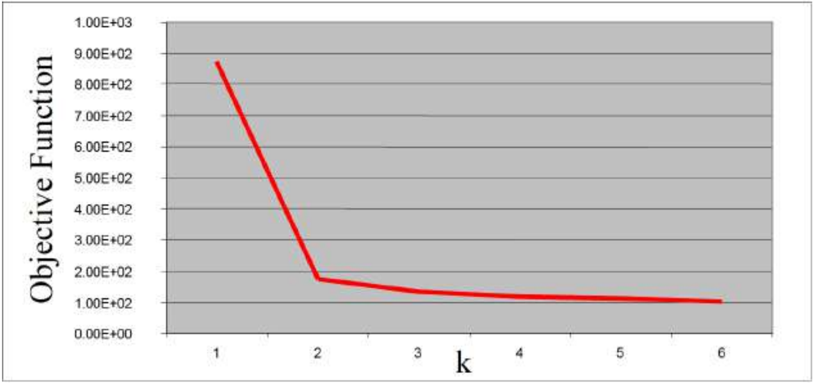


When $k = 3$, objective value is 133.6.



Elbow Method

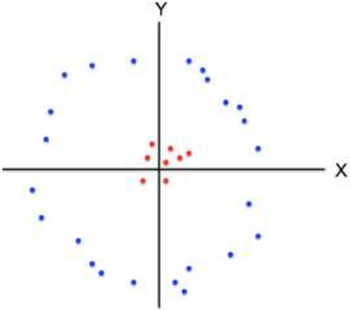
In this case, there is an abrupt change at $k = 2$ that suggests there are two "natural" clusters in the data.



Practical Issues with K-means

- How to select k?
  - Prior knowledge
  - Heuristics (e.g., elbow method)
- How to select distance measure?
  - Often requires some knowledge of problem
  - Some examples: Euclidean distance (for images), Hamming distance (distance between two strings), shared key words (for websites)

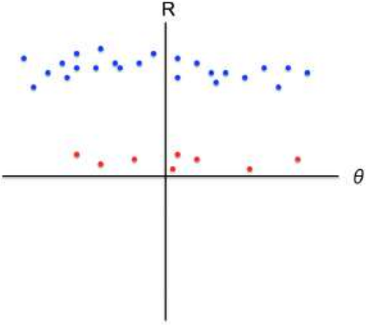
How to Get *K*-means to Work on This Data?



Should look at the distance of the data points from the origin $\sqrt{x_n^2 + y_n^2}$

Distance Measure

Changing features (distance measure) can help



If the cluster i mean is $(\mu_{i,x}, \mu_{i,y})$, the distance of (x_n, y_n) from it can be defined as $|\sqrt{\mu_{i,x}^2 + \mu_{i,y}^2} - \sqrt{x_n^2 + y_n^2}|$

Scaling Features

Suppose the $\mathbf{x}_n$ represent homes, with features (# of bedrooms, square footage).

For data point $(2, 1000)$ and cluster center $(3, 2000)$:

$$\|\mathbf{x}_n - \mu_k\|_2^2 = (2-3)^2 + (1000-2000)^2.$$

- If one feature of $\mathbf{x}_n$ is much larger than the others, this feature will dominate our distance measure and thus the clustering.
- Scale features to ensure all features are considered.
- As in linear regression, many scaling methods are possible:

$$x_{nd} \rightarrow \frac{x_{nd}}{\max_m x_{md} - \min_m x_{md}}, \quad x_{nd} \rightarrow \frac{x_{nd}}{\text{stdev}\left\{x_{md}\right\}}, \ldots$$

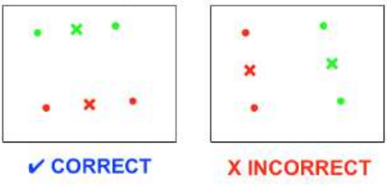
- Requires domain knowledge. E.g., if data points represent pixels with features (red value, green value), then red values between 1 and 10 and green values between 100 and 200 mean we *should* cluster mostly on green values, as these differentiate the colors more.

Practical Issues with K-means

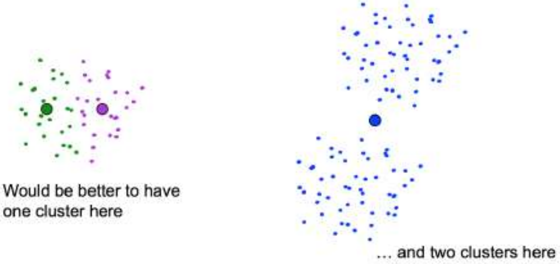
- How to select k?
  - Prior knowledge
  - Heuristics (e.g., elbow method)
- How to select distance measure?
  - Often requires some knowledge of problem
  - Some examples: Euclidean distance (for images), Hamming distance (distance between two strings), shared key words (for websites)
- How to initialize cluster centers?
  - The final clustering can depend significantly on the initial points you pick!

How to Initialize Cluster Centers?

Random initialization can lead to *different results*



Choosing k is also non-trivial



K-means++

K-means++

Key idea: Run K-means, but with a better initialization

- Choose center μ_1 at random
- For $j = 2, \dots, k$
    - Choose μ_j among $x_1, \dots, x_n$ with probability:

      $$P(\mu_j = x_i) \propto min_{j' < j} \|x_i - \mu_{j'}\|^2$$

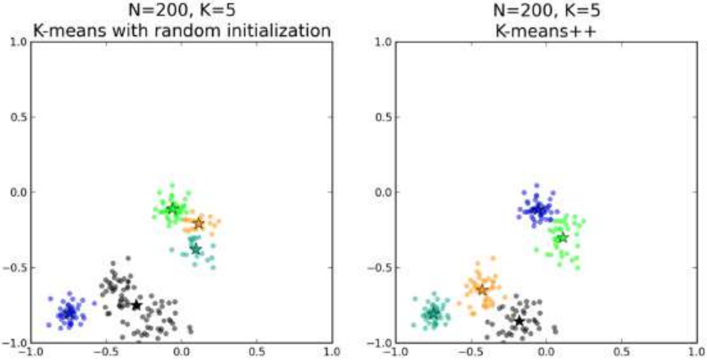
      This means that if x_i is close to one of the already chosen cluster means μ_1, $\dots \mu_{j-1}$, then we assign a lower probability of selecting it as the next cluster mean.

Initialization helps to get good coverage of the space

Theorem: K-means++ always obtains a $O(\log k)$ approximation to the optimal solution in expectation.

Running K-means after this initialization can only improve on the result

K-means++


N=200, K=5
K-means with random initialization
N=200, K=5
K-means++
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0

Connection to k-Nearest Neighbors

- Nearest Neighbors is a supervised learning method
  - Each training point $\mathbf{x}_n$ has a corresponding given label y_n
  - Objective: Assign label to a new $\mathbf{x}$ by looking at the labels of its k nearest points
- Clustering is an unsupervised learning method
  - We are given training points $\mathbf{x}_n$ without labels
  - Objective: Divide them into k groups to understand patterns in the data

The meaning of the parameter k is also different in these two methods

Clustering Can Make Nearest Neighbors More Efficient

- A drawback of nearest neighbors is that we have to remember the training data
- Clustering can help compress the training data into a small number of representative points

Algorithm to Improve Nearest Neighbors

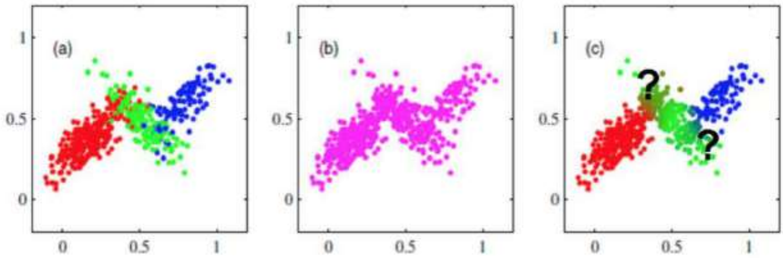
- For all training data points $\mathbf{x}_n$ with label $y_n = c$, for C classes $c = 1, \ldots C$, cluster the $\mathbf{x}_n$ into R groups.
- Store these R cluster means for each of the C classes
- For a test data point $\mathbf{x}$, find the k nearest neighbors among the RC cluster means and assign their majority label to $\mathbf{x}$

Gaussian Mixture Models

Potential Issues with k-means . . .

Data points are assigned *deterministically* to one (and only one) cluster

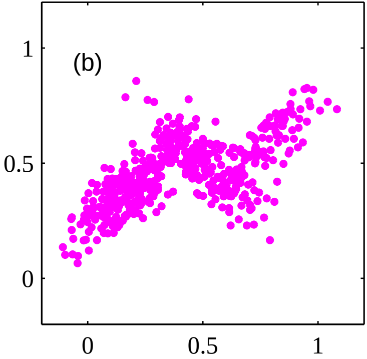
In reality, clusters may overlap, and it may be better to identify the *probability* that a point belongs to each cluster



Also, distances are measured in a homogeneous manner. In reality, some clusters may be more spread out than others

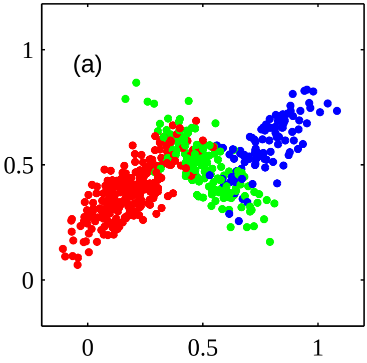
Probabilistic Interpretation of Clustering?

How can we model $p(\boldsymbol{x})$ to reflect our intuition that points stay close to their cluster centers?



- Points seem to form 3 clusters
- We cannot model $p(\boldsymbol{x})$ with simple and known distributions
- E.g., the data is not a Gaussian b/c we have 3 distinct concentrated regions

Gaussian Mixture Models: Intuition



- Key idea: Model *each* region with a distinct distribution
- Can use Gaussians — Gaussian mixture models (GMMs)
- *However*, we don't know *cluster assignments* (label), *parameters* of Gaussians, or *mixture components*!
- Must learn from *unlabeled* data $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$

You Should Know

- What unsupervised learning is
- What clustering is
- How to cluster using K-means
- Practical issues with K-means
- How K-means++ improves on K-means