CPSC 340:
Machine Learning and Data Mining

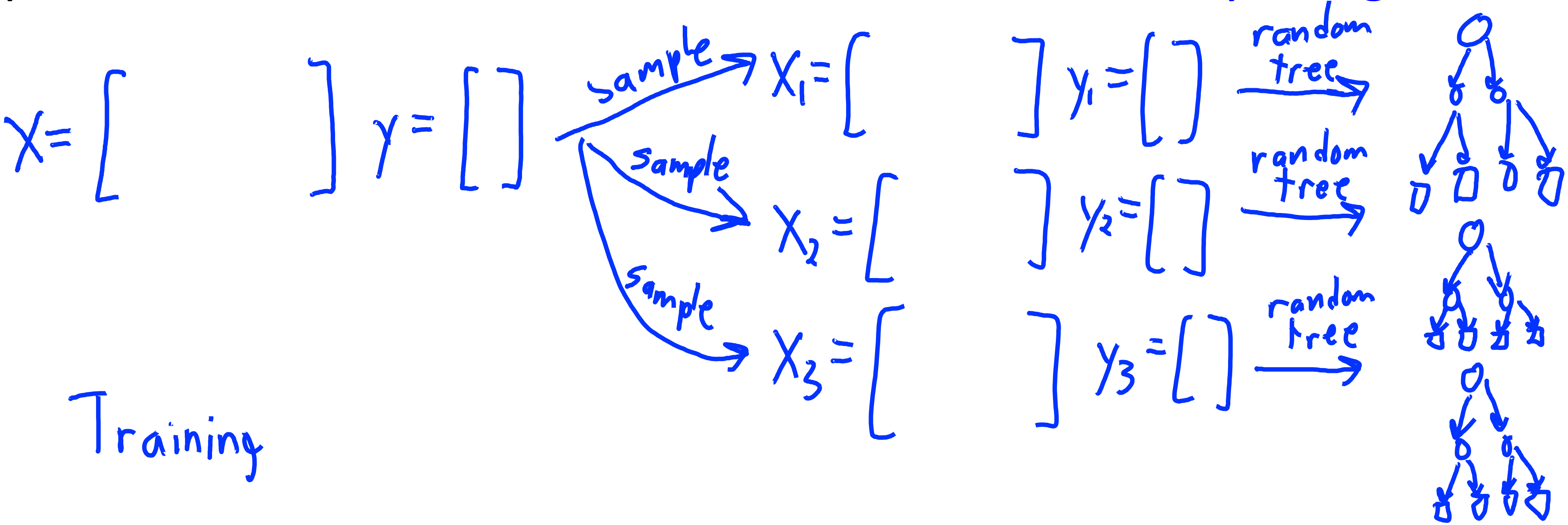
K-Means Clustering

Admin

- Assignment 1 is due Sunday night.
- The last possible time to submit is next Wednesday night.
  - Solutions will be posted after that.
  - You will be able to see each other's work after that.
- Assignment 2 coming soon.
- Reminder: midterm on March 1
  - If you need A&D provisions, please email me soon.

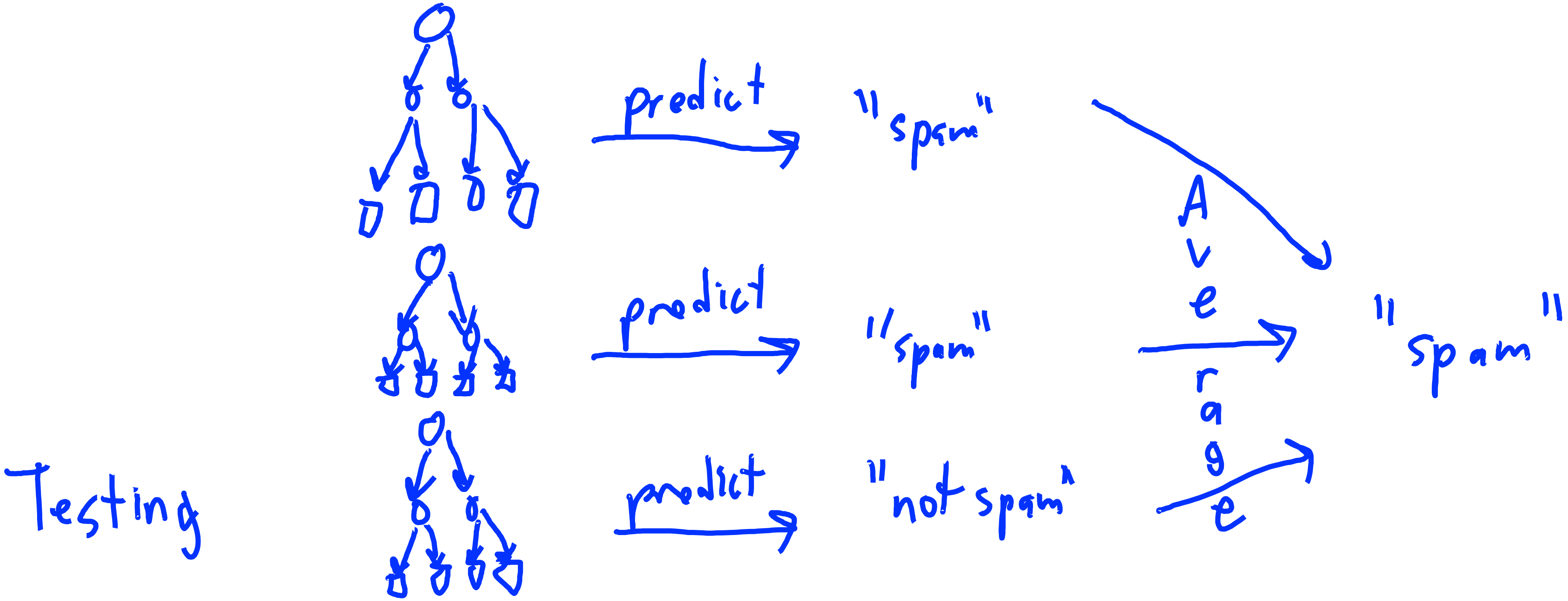
Random Forests

- Random forests are one of the best ‘out of the box’ classifiers.
- Fit deep decision trees to random bootstrap samples of data, base splits on random subsets of the features, and classify using mode.



Random Forests

- Random forests are one of the best ‘out of the box’ classifiers.
- Fit deep decision trees to random bootstrap samples of data, base splits on random subsets of the features, and classify using mode.



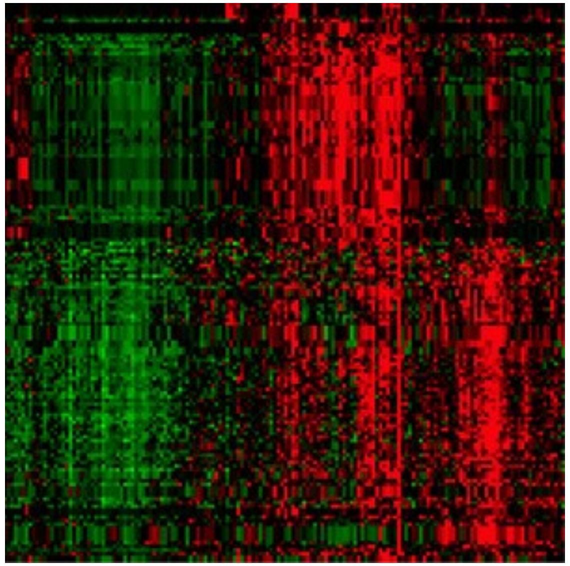
End of Part 1: Key Concepts

- Fundamental ideas:
  - Training vs. test error.
  - Golden rule of ML (test set should not influence training).
  - Fundamental trade-off (low training error vs. overfitting).
  - Validation sets and cross-validation.
  - Parametric vs. non-parametric.
  - No free lunch theorem (there is no “best” model).
  - Ensemble methods (combining models).
- Methods that we focused on:
  - Decision trees (greedy recursive splitting using decision stumps).
  - Naïve Bayes (generative classifier based on conditional independence).
  - K-nearest neighbours (non-parametric classifier with universal consistency).
  - Random forests (averaging plus randomization to reduce overfitting).

Application: Classifying Cancer Types

- "I collected gene expression data for 1000 different types of cancer cells, can you tell me the different classes of cancer?"

X =



- We are not given the class labels y, but want meaningful labels.
- An example of unsupervised learning.

https://corelifesciences.com/human-long-non-coding-rna-expression-microarray-service.html

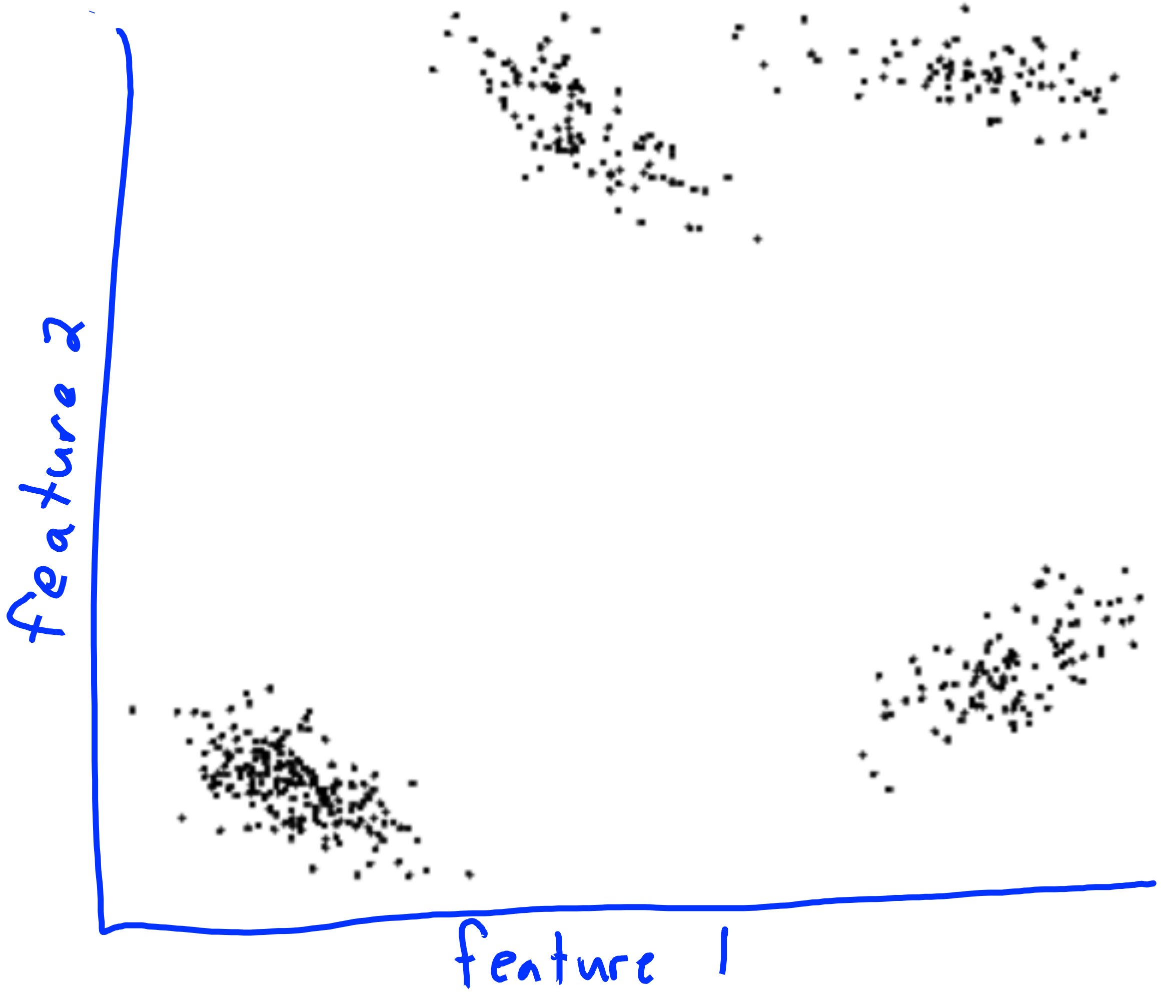
Unsupervised Learning

- Supervised learning:
  - We have features x_i and class labels y_i.
  - Write a program that produces y_i from x_i.
- Unsupervised learning:
  - We only have x_i values, but no explicit target labels.
  - You want to do ‘something’ with them.
- Some unsupervised learning tasks:
  - Outlier detection: Is this a ‘normal’ x_i?
  - Data visualization: What does the high-dimensional X look like?
  - Association rules: Which x_{ij} occur together?
  - Latent-factors: What ‘parts’ are the x_i made from?
  - Ranking: Which are the most important x_i?
  - Clustering: What types of x_i are there?

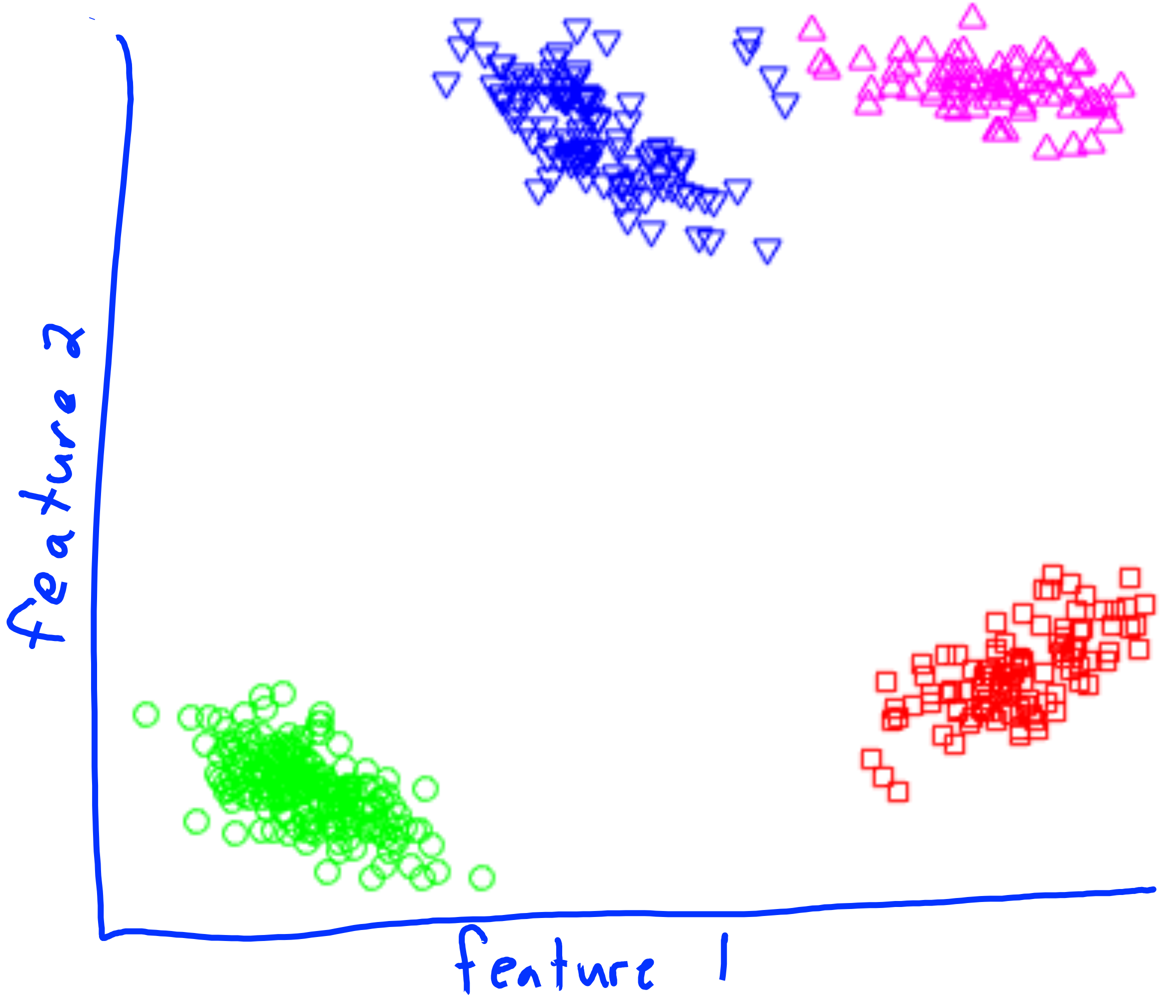
Clustering

- Clustering:
  - Input: set of objects described by features x_i.
  - Output: an assignment of objects to ‘groups’.
- Unlike classification, we are not given the ‘groups’.
  - Algorithm must discover groups.
- Example of groups we might discover in e-mail spam:
  - ‘Lucky winner’ group.
  - ‘Weight loss’ group.
  - ‘Nigerian prince’ group.

Clustering Example



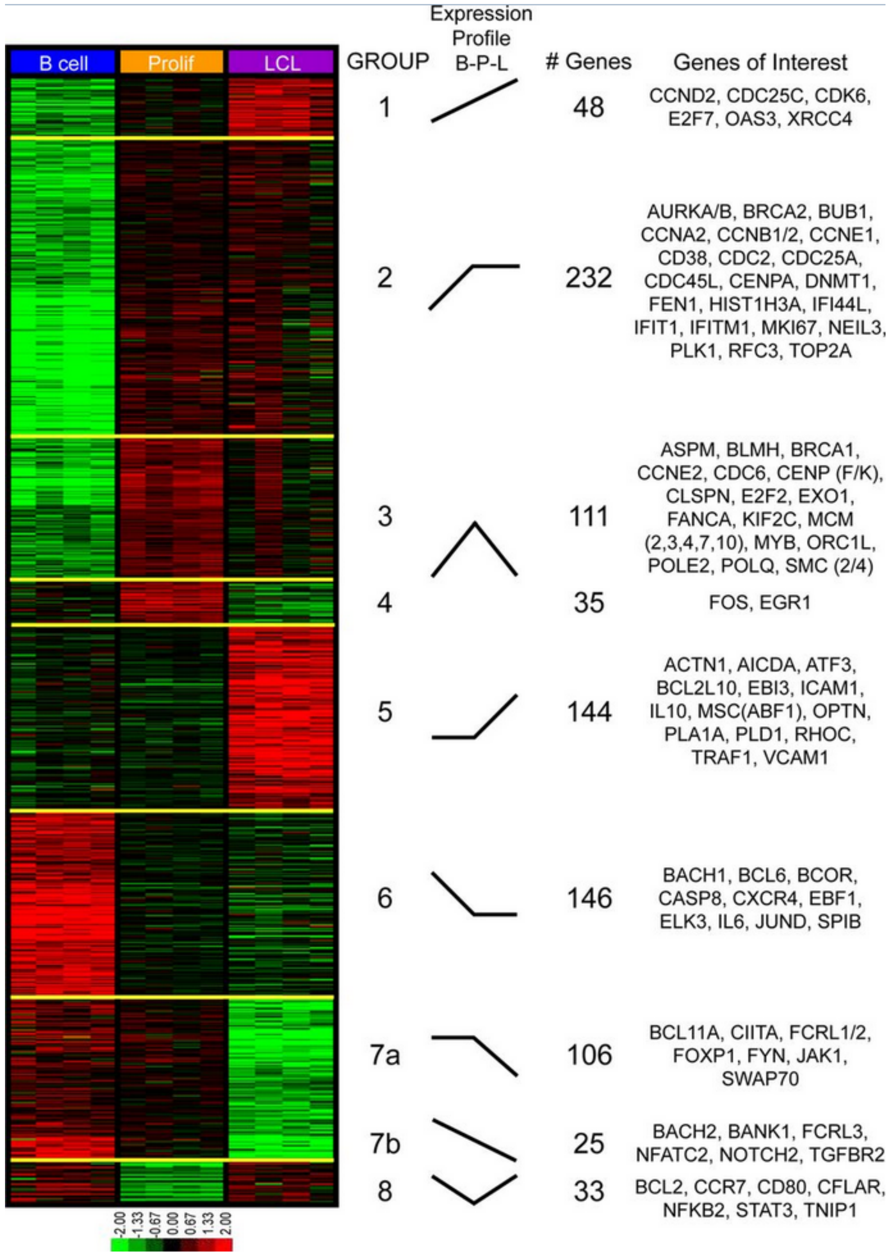
Clustering Example



Data Clustering

- General goal of clustering algorithms:
  - Objects in the same group should be 'similar'.
  - Objects in different groups should be 'different'.
- But the 'best' clustering is hard to define:
  - We don't have a test error.
  - Generally, there is no 'best' method in unsupervised learning.
    - Means there are lots of methods: we'll focus on important/representative ones.
- Why cluster?
  - You could want to know what the groups are.
  - You could want a 'prototype' example for each group.
  - You could want to find the group for a new example x.
  - You could want to find objects related to a new example x.

Clustering of Epstein-Barr Virus

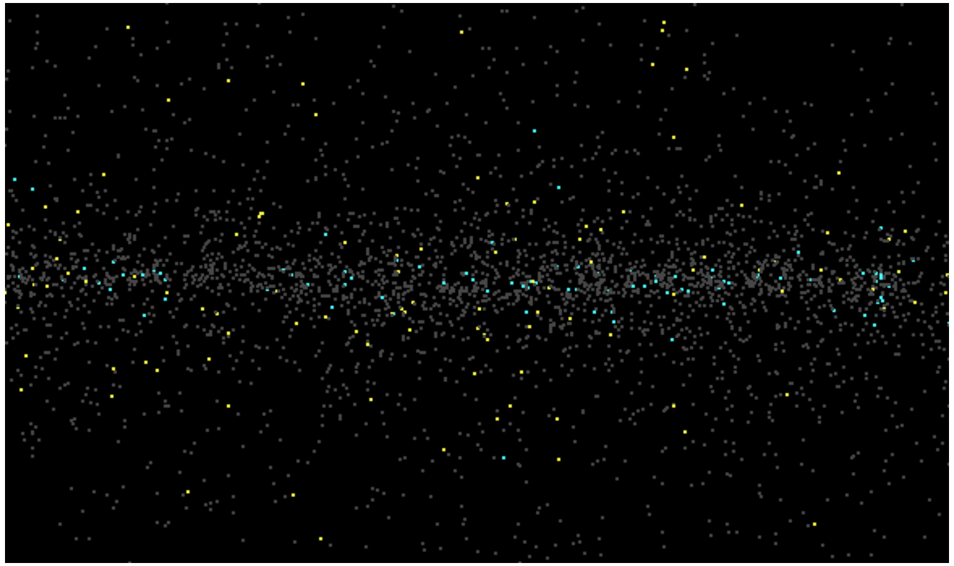


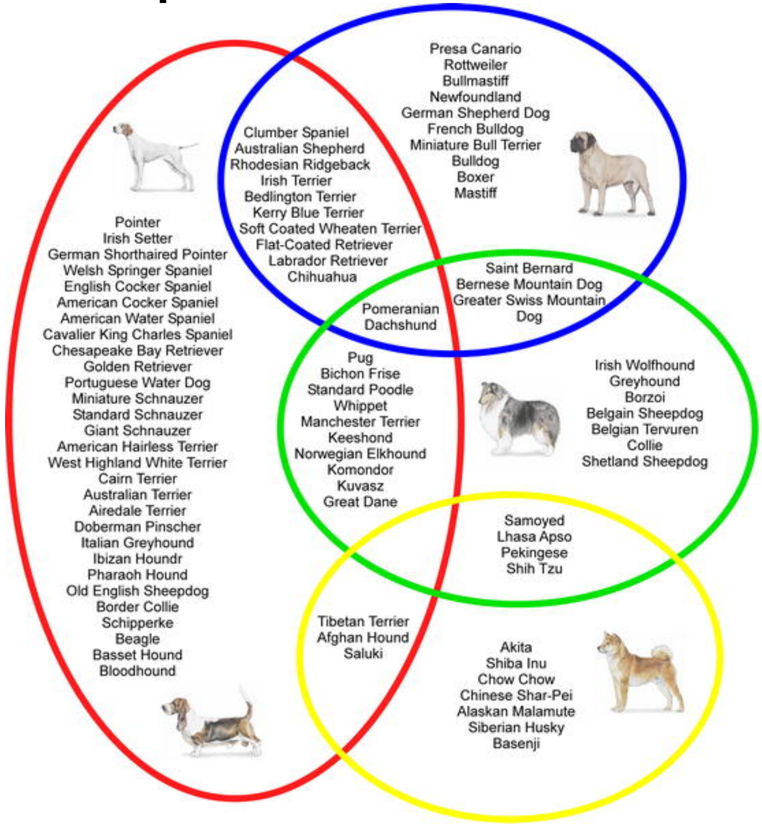
GROUP	Expression Profile B-P-L	# Genes	Genes of Interest
1		48	CCND2, CDC25C, CDK6, E2F7, OAS3, XRCC4
2		232	AURKA/B, BRCA2, BUB1, CCNA2, CCNB1/2, CCNE1, CD38, CDC2, CDC25A, CDC45L, CENPA, DNMT1, FEN1, HIST1H3A, IFI44L, IFIT1, IFITM1, MKI67, NEIL3, PLK1, RFC3, TOP2A
3		111	ASPM, BLMH, BRCA1, CCNE2, CDC6, CENP (F/K), CLSPN, E2F2, EXO1, FANCA, KIF2C, MCM (2,3,4,7,10), MYB, ORC1L, POLE2, POLQ, SMC (2/4)
4		35	FOS, EGR1
5		144	ACTN1, AICDA, ATF3, BCL2L10, EBI3, ICAM1, IL10, MSC(ABF1), OPTN, PLA1A, PLD1, RHOC, TRAF1, VCAM1
6		146	BACH1, BCL6, BCOR, CASP8, CXCR4, EBF1, ELK3, IL6, JUND, SPIB
7a		106	BCL11A, CIITA, FCRL1/2, FOXP1, FYN, JAK1, SWAP70
7b		25	BACH2, BANK1, FCRL3, NFATC2, NOTCH2, TGFBR2
8		33	BCL2, CCR7, CD80, CFLAR, NFKB2, STAT3, TNIP1

http://jvi.asm.org/content/86/20/11096.abstract

Other Clustering Applications

- NASA: what types of stars are there?
- Biology: are there sub-species?
- Documents: what kinds of documents are on my computer?
- Commercial: what kinds of customers do I have?



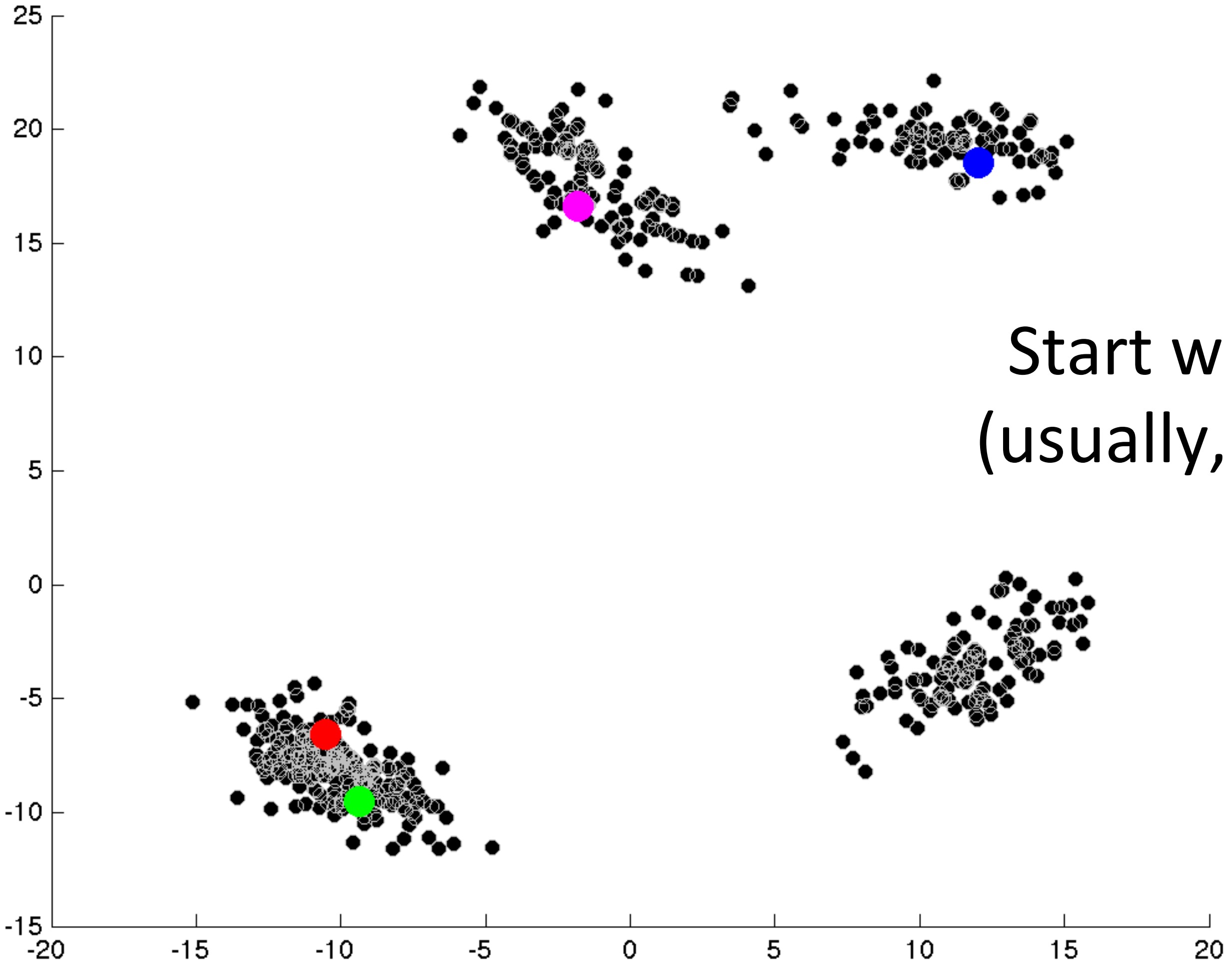


http://www.eecs.wsu.edu/~cook/dm/lectures/l9/index.html
http://www.biology-online.org/articles/canine_genomics_genetics_running/figures.html

K-Means

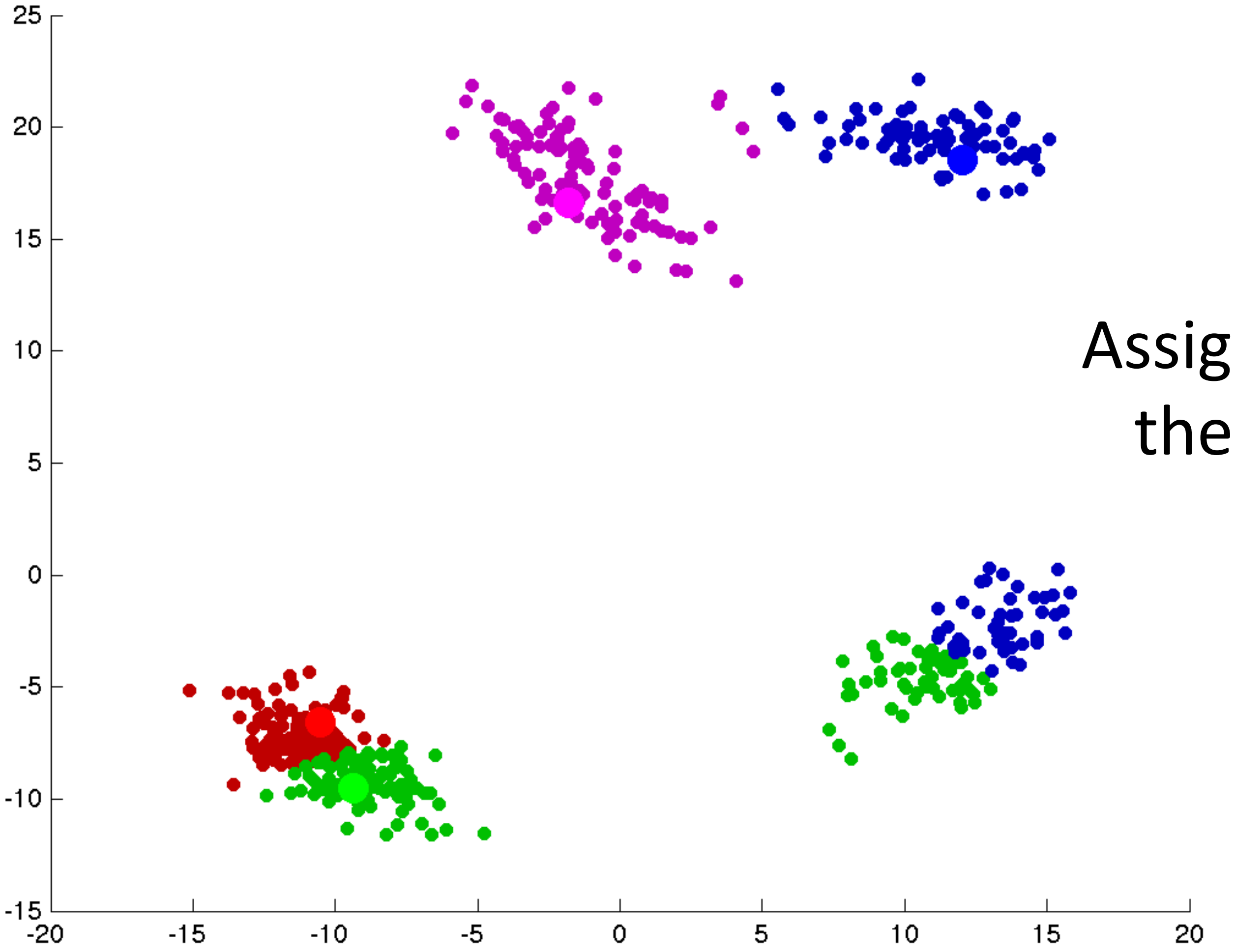
- Most popular clustering method is k-means.
- Input:
  - The number of clusters ‘k’.
  - Initial guesses of the center (“mean”) of each cluster.
- Algorithm:
  - Assign each x_i to its closest mean.
  - Update the means based on the assignment.
  - Repeat until convergence.
- Note: not to be confused with k-NN, which is supervised learning.
  - But both require a notion of “distance”.

K-Means Example



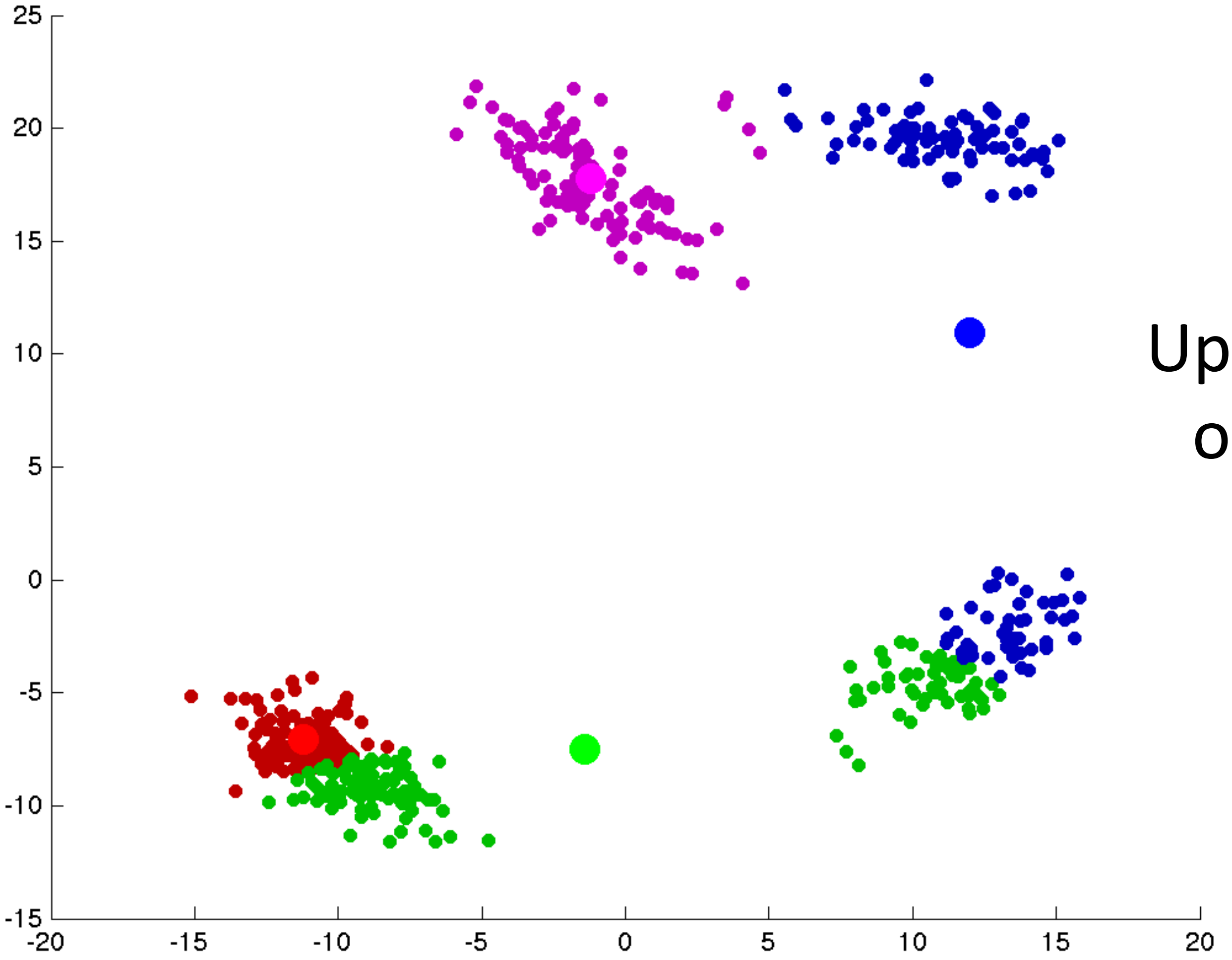
Start with ‘k’ initial ‘means’
(usually, random data points)

K-Means Example



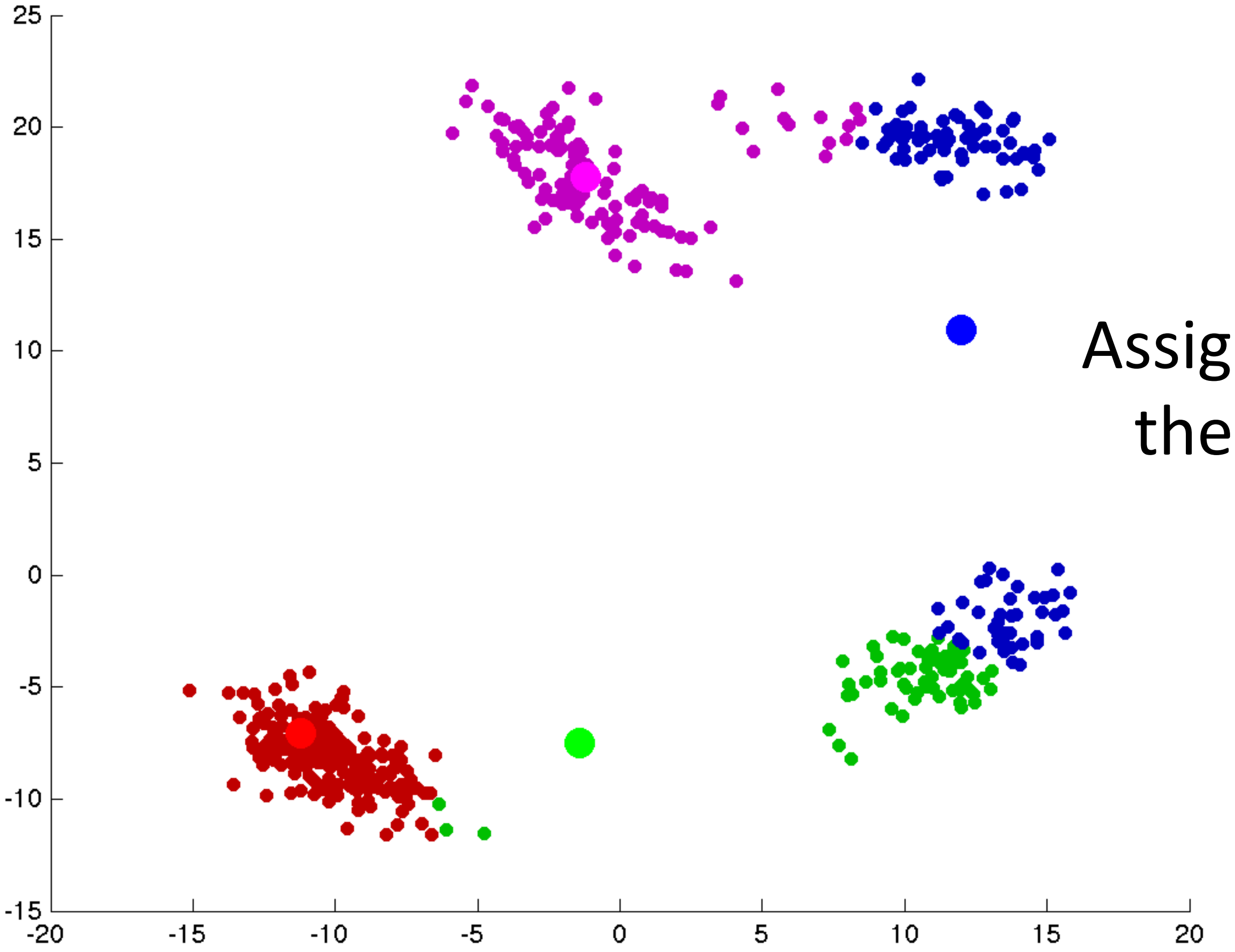
Assign each object to the closest mean.

K-Means Example



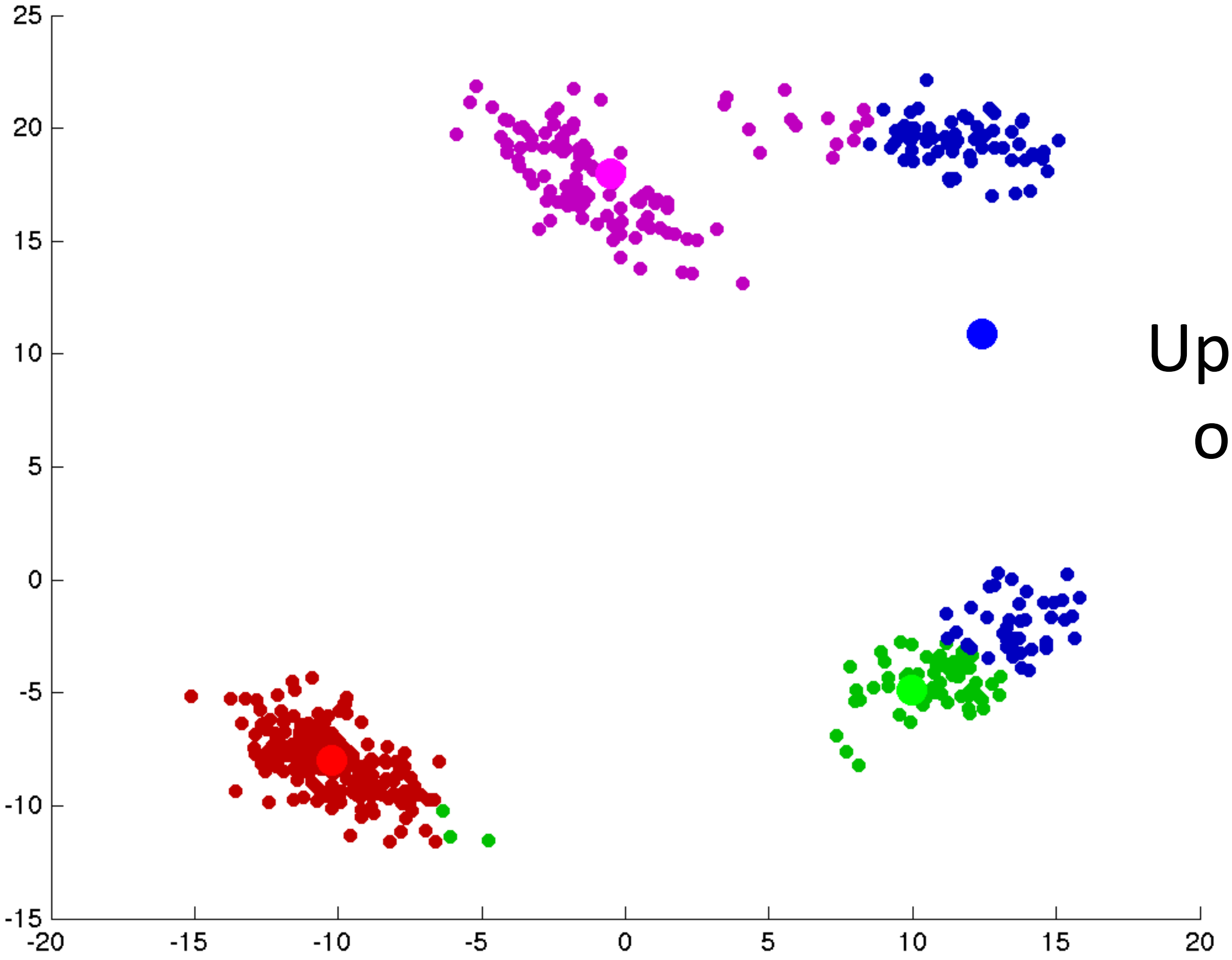
Update the mean of each group.

K-Means Example



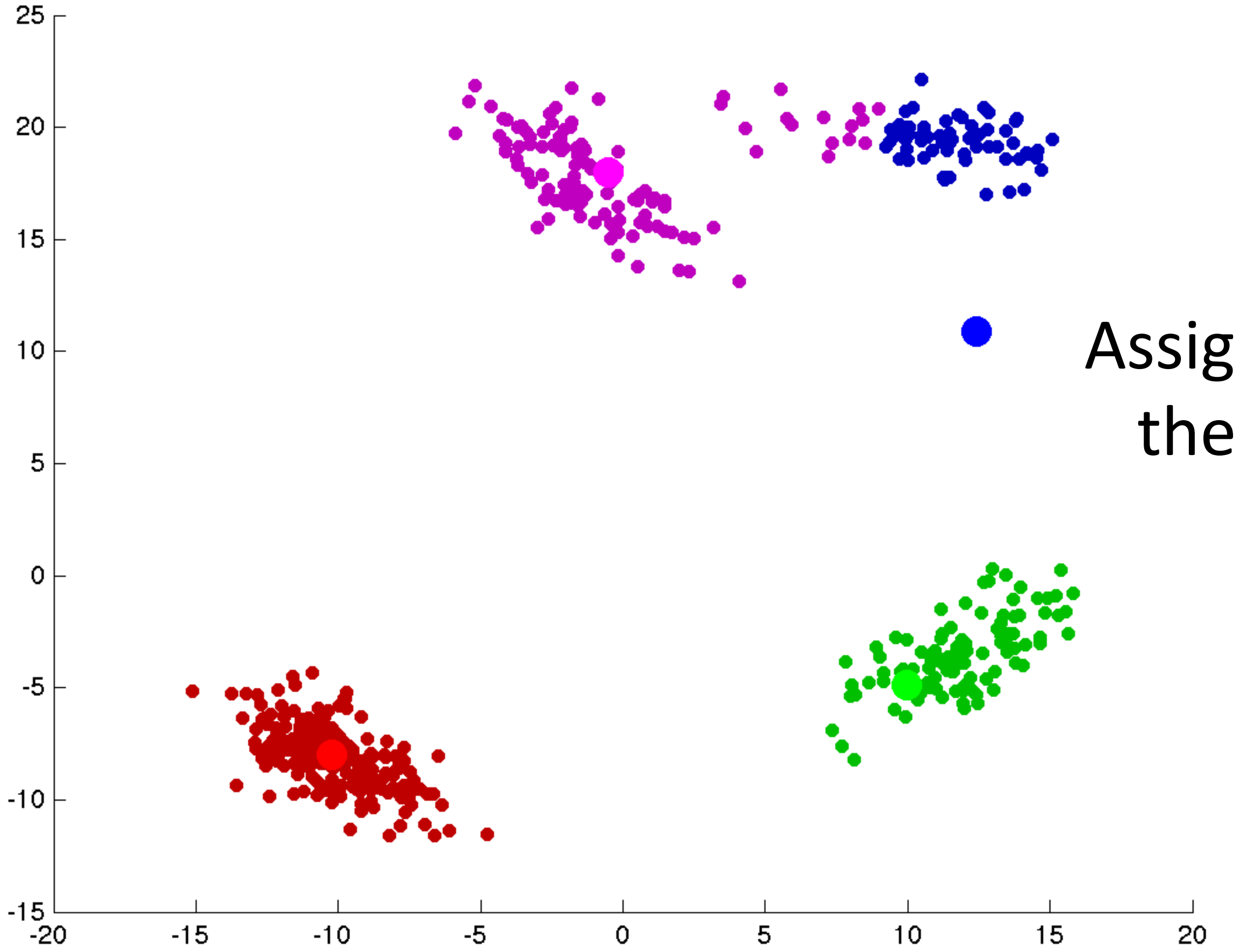
Assign each object to the closest mean.

K-Means Example



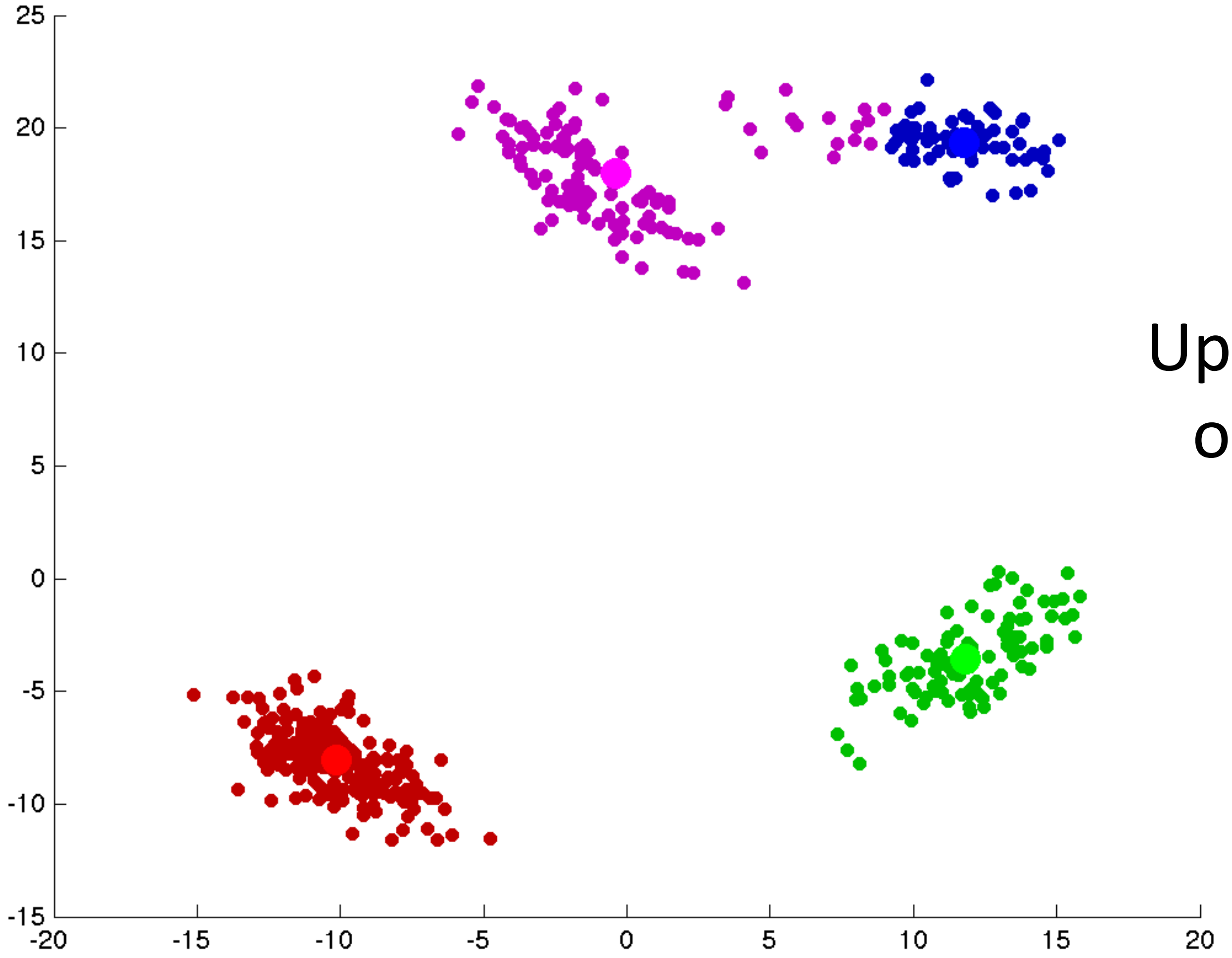
Update the mean
of each group.

K-Means Example



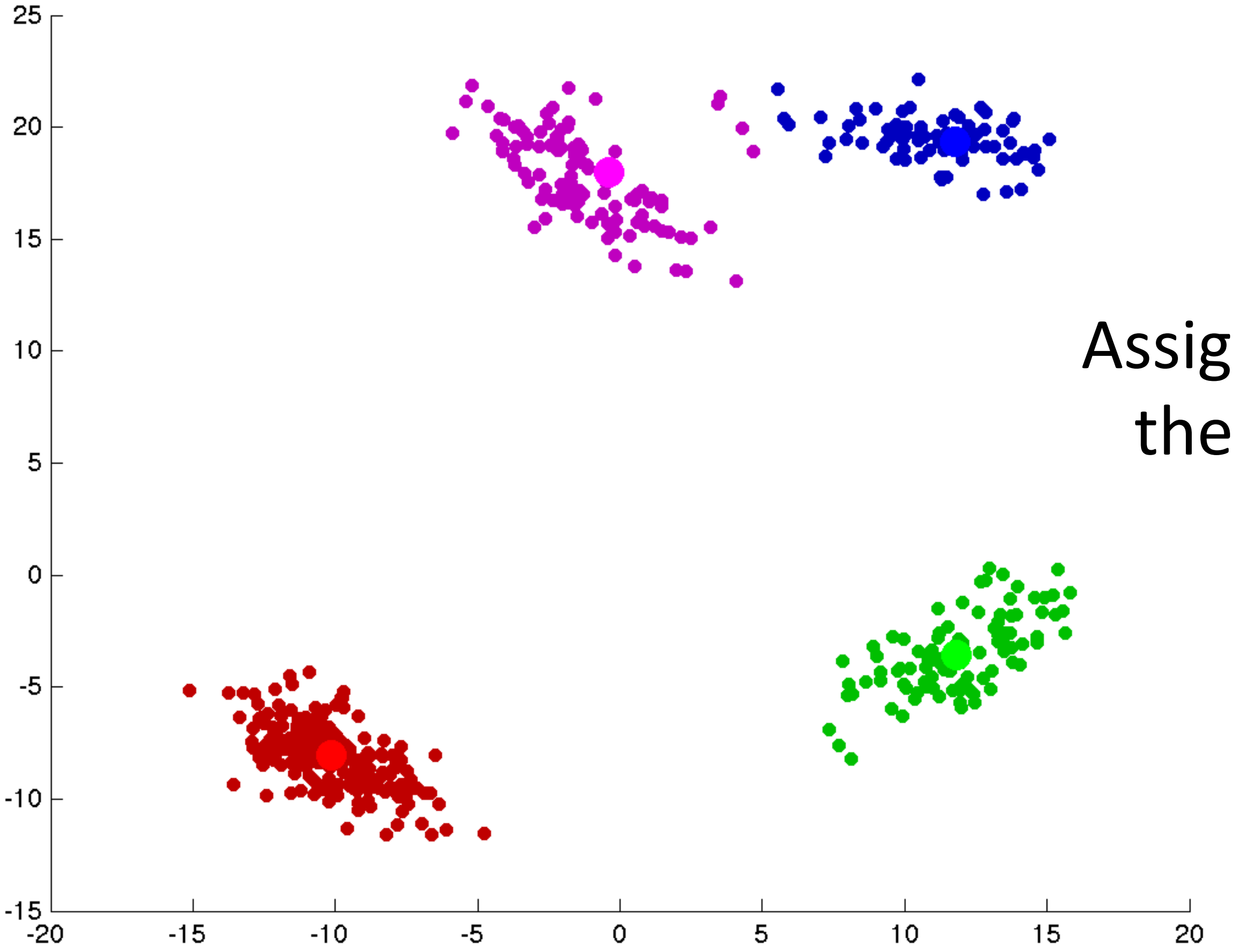
Assign each object to the closest mean.

K-Means Example



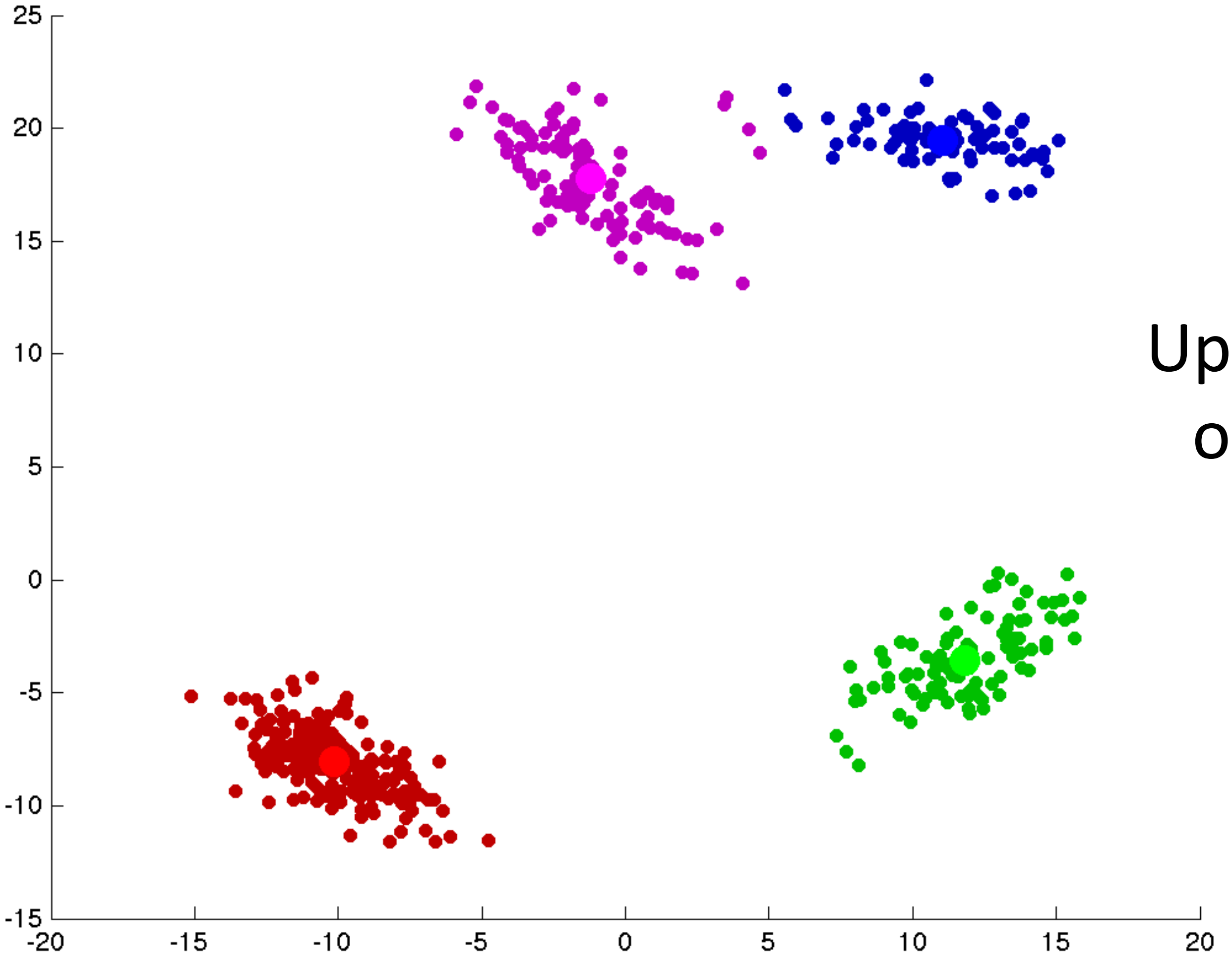
Update the mean of each group.

K-Means Example



Assign each object to the closest mean.

K-Means Example



Update the mean of each group.

Stop if no objects change groups.

Interactive demo:
https://www.naftaliharris.com/blog/visualizing-k-means-clustering

K-Means Issues

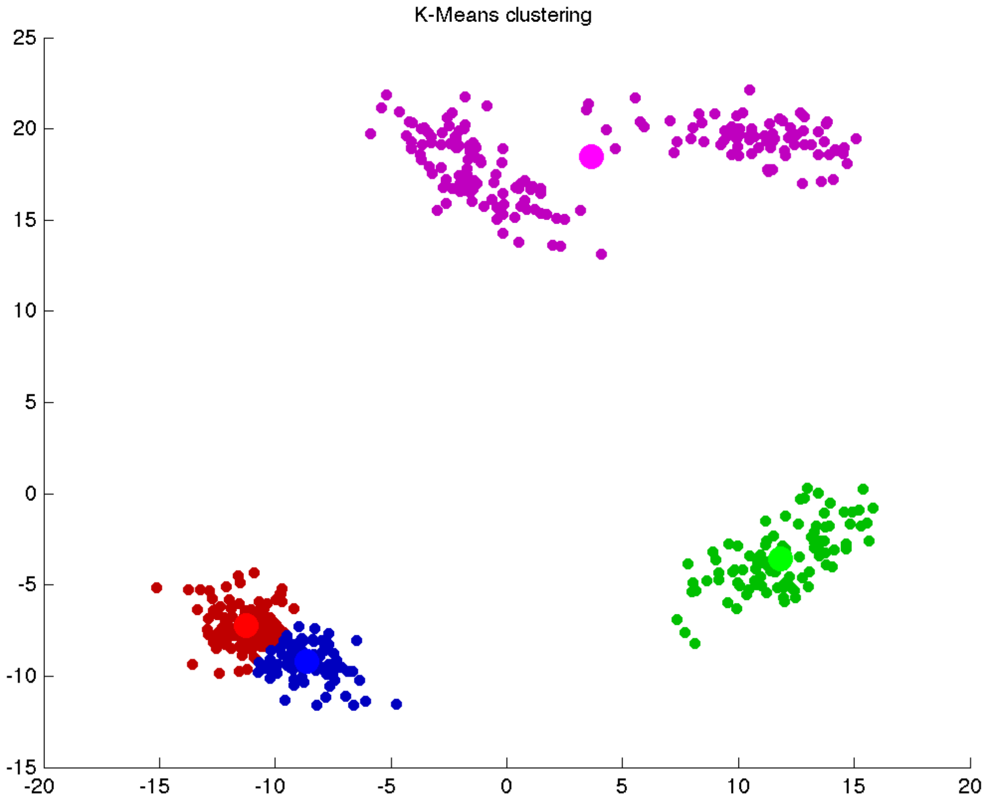
- Guaranteed to converge when using Euclidean distance.
- New object are assigned to nearest mean to cluster them.

- Assumes you know number of clusters ‘k’.
  - Lots of heuristics to pick ‘k’, none satisfying:
    - https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set

- Each object is assigned to one (and only one) cluster:
  - No possibility for overlapping clusters or leaving objects unassigned.

- It may converge to sub-optimal solution…

K-Means Clustering with Different Initialization



- Classic approach to dealing with sensitivity to initialization:
  - Try several different random starting points, choose the 'best'.
- We'll see a more clever approach next time…

Cost of K-means

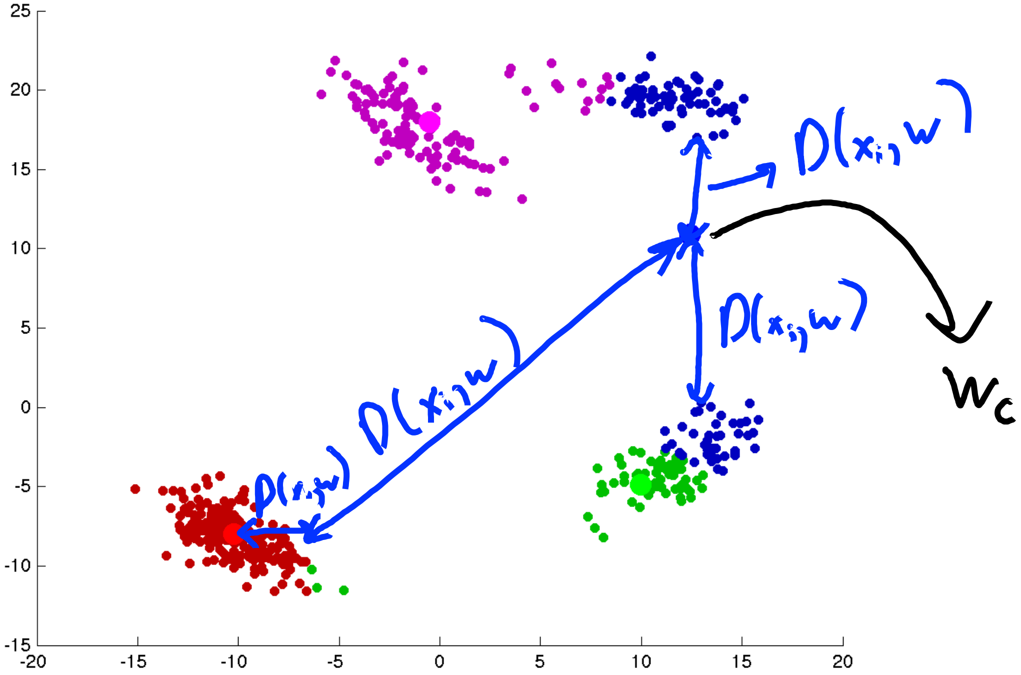
- Bottleneck is calculating distance from each x_i to each mean w_c:

$$D(x_i, w_c) = \sqrt{\sum_{j=1}^{d} (x_{ij} - w_{cj})^2}$$

vector

center of cluster "c"

scalar



$w_c = [13, 11]$

Cost of K-means

- Bottleneck is calculating distance from each x_i to each mean w_c:

$$D(x_i, w_c) = \sqrt{\sum_{j=1}^{d} (x_{ij} - w_{cj})^2}$$

vector

center of cluster "c"

scalar

- Each time we do this costs O(d) to go through all features.
- For each of the 'n' objects, we compute the distance to 'k' clusters.
- Total cost of assigning objects to clusters is O(ndk).
  - Fast if k is not too large.
- Updating means is cheaper: O(nd).

– For each cluster 'c', compute $w_c = \frac{1}{n_c} \sum_{i \in C} x_i$

Object in cluster.

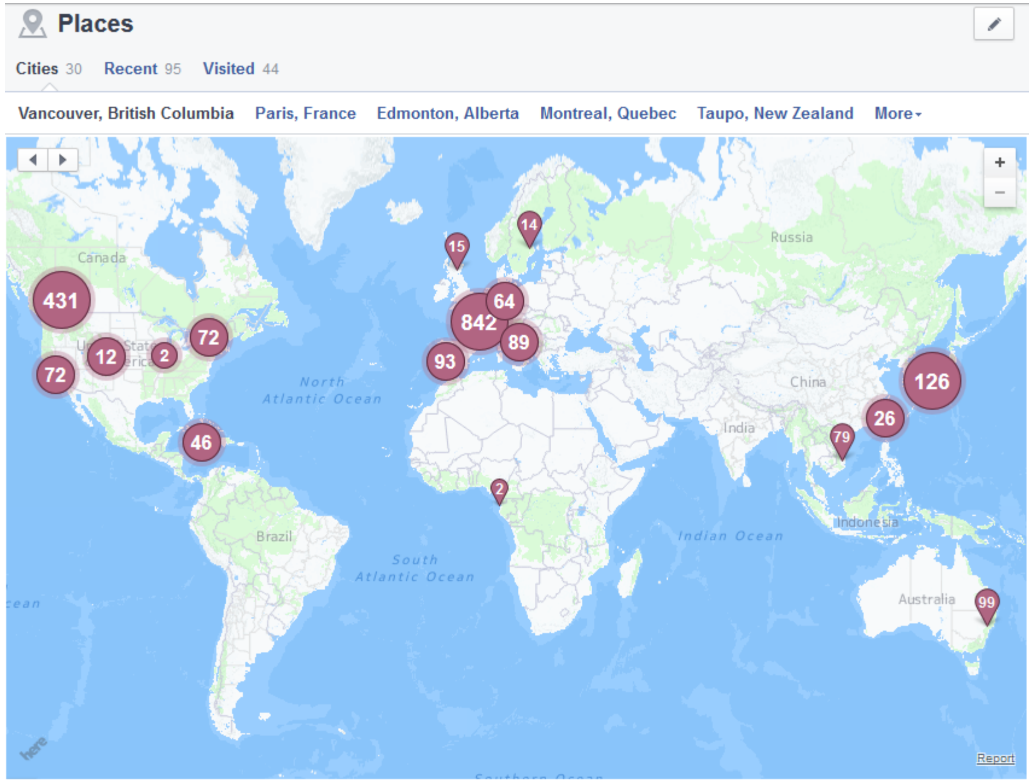
Loop over objects in cluster.

Number of objects in cluster 'c'

Vector Quantization

- K-means originally comes from signal processing.
- Designed for vector quantization:
  - Replace ‘vectors’ (objects) with a set of ‘prototypes’ (means).

- Example:
  - Facebook places.
  - What sizes of clothing should I make?



http://wannabite.com/wp-content/uploads/2014/10/ragu-pasta-sauce-printable-coupon.jpg

Vector Quantization: Image Colors

- Usual RGB representation of a pixel’s color: three 8-bit numbers.
  - For example, [241 13 50] = 🟥.
  - Can apply k-means to find set of prototype colours.

Original:
(24-bits/pixel)

$$X = \left.\underbrace{\begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}}_{\substack{3\\(\text{all 8-bit})}}\right\}\#\text{pixels}$$

K-Means Quantized:
(6-bits/pixel)

$$y = \left.\begin{bmatrix}1\\2\\3\\2\\2\\1\\1\\3\end{bmatrix}\right\}\text{cluster for each pixel}$$

1 6-bit number lets us have 2^6 clusters

$$W = \left.\underbrace{\begin{bmatrix} \bigcirc \\ \\ \\ \end{bmatrix}}_{3}\right\} 2^6 \rightarrow 64 \text{ colours}$$

Average red, green, and blue values in cluster 1.

Vector Quantization: Image Colors

- Usual RGB representation of a pixel’s color: three 8-bit numbers.
  - For example, [241 13 50] = 🟥 .
  - Can apply k-means to find set of prototype colours.

Original:
(24-bits/pixel)

$$X = \underbrace{\begin{bmatrix} \quad \\ \\ \\ \end{bmatrix}}_{\substack{3 \\ \text{(all 8-bit)}}} \Big\} \text{\# pixels}$$

K-Means Quantized:
(3-bits/pixel)

$$y = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 2 \\ 1 \\ 1 \\ 3 \end{bmatrix} \Big\} \text{cluster for each pixel}$$

1 3-bit number lets us have 2^3 clusters

Average red, green, and blue values in cluster 1.

$$W = \underbrace{\begin{bmatrix} \quad \\ \\ \\ \end{bmatrix}}_{3} \Big\} 2^3 = 8$$

Vector Quantization: Image Colors

- Usual RGB representation of a pixel’s color: three 8-bit numbers.
  - For example, [241 13 50] = 🟥 .
  - Can apply k-means to find set of prototype colours.

Original:
(24-bits/pixel)

$$X = \underbrace{\begin{bmatrix} \quad & \quad & \quad \\ \\ \\ \\ \end{bmatrix}}_{\substack{3 \\ \text{(all 8-bit)}}} \Bigg\} \# \text{pixels}$$

K-Means Quantized:
(2-bits/pixel)

$$y = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 2 \\ 1 \\ 1 \\ 3 \end{bmatrix} \Bigg\} \text{cluster for each pixel}$$

1 2-bit number lets us have 2^2 clusters.

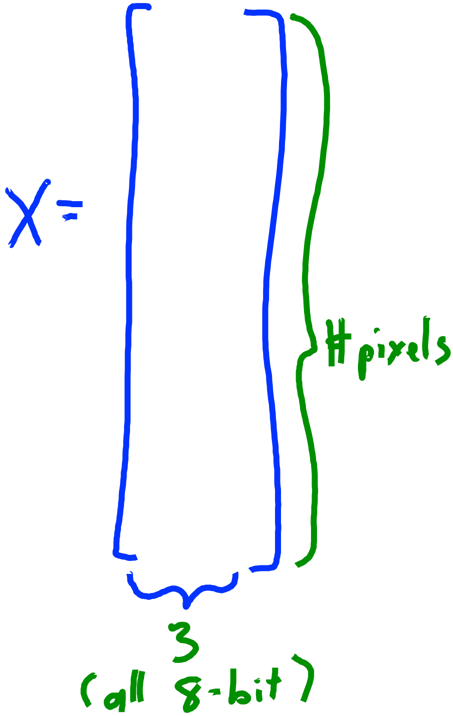
$$W = \underbrace{\begin{bmatrix} \quad & \quad & \quad \\ \\ \\ \\ \end{bmatrix}}_{3} \Bigg\} 4$$

Average red, green, and blue values in cluster 1.

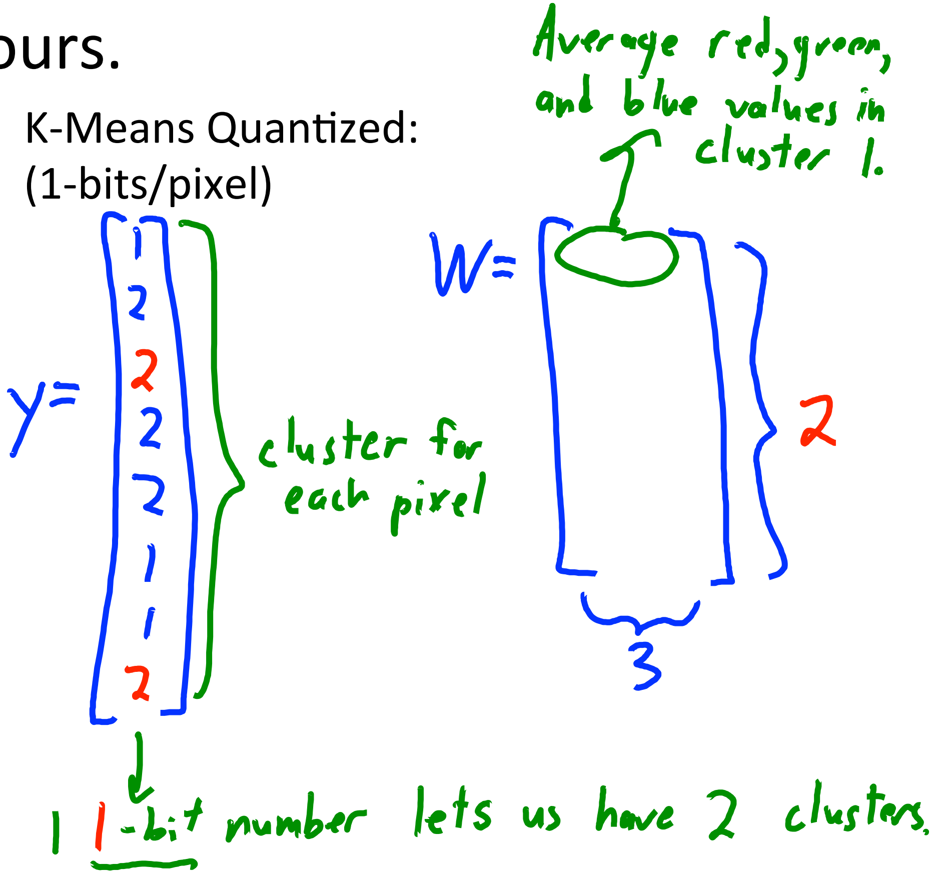
Vector Quantization: Image Colors

- Usual RGB representation of a pixel’s color: three 8-bit numbers.
  - For example, [241 13 50] = 🟥 .
  - Can apply k-means to find set of prototype colours.

Original:
(24-bits/pixel)



K-Means Quantized:
(1-bits/pixel)



Unsupervised Learning and Compression

- What we were doing is *compressing* the data
- Unsupervised learning and data compression have a lot in common
  - Both require understanding the "patterns" in the data
  - The clustering objective is called "distortion" in lossy compression parlance
- There is also a connection with supervised learning
  - We want a model that is simpler than the whole data set
  - kNN is an exception because there is no compression/simplification.

What is K-Means Doing?

- We can interpret K-Means as trying to minimize an objective:
  - Total sum of squared distances from object x_i to their centers $w_{c(i)}$:

$$f(w_1, w_2, \dots, w_k, c(1), c(2), \dots, c(n)) = \sum_{i=1}^{n} \sum_{j=1}^{d} (x_{ij} - \underbrace{w_{c(i)j}}_{\text{Cluster of example 'i'}})^2$$

- The k-means steps:
  - Optimally update cluster assignments c(i).
  - Optimally update means w_c.
- Convergence follows because:
  - Each step does not increase the objective.
  - There are a finite number of assignments to k clusters.

K-Medians Clustering

- With other distances, k-means may not converge.
- However, changing objective function gives convergent algorithms.
- E.g., we can use the L1-norm: $\sum_{i=1}^{n}\sum_{j=1}^{d}|x_{ij} - w_{c(i)j}|$
- A 'k-medians' algorithm based on the L1-norm:
  - Cluster assignment based on the L1-norm (nearest median).
  - Update 'medians' as median value (dimension-wise) of each cluster.
- This approach is more robust to outliers.

Summary

- Unsupervised learning: fitting data without explicit labels.
- Clustering: finding ‘groups’ of related objects.
- K-means: simple iterative clustering strategy.
- Vector quantization: replacing measurements with ‘prototypes’.
- K-medians: generalization to other distance functions.

- Next time:
  - Non-parametric clustering.