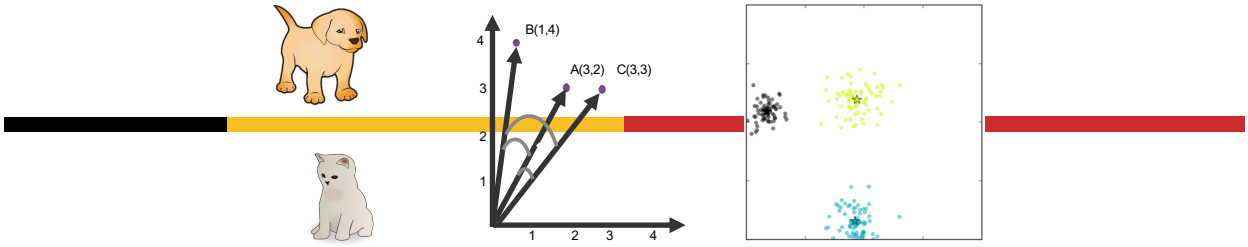
Clustering: *k*-means, Expectation-Maximization



Based partly on: M desJardins, T Oates, P Matuszek, RJ Mooney: www.cs.utexas.edu/~mooney/cs388/slides/TextClustering.ppt, and other sources as noted

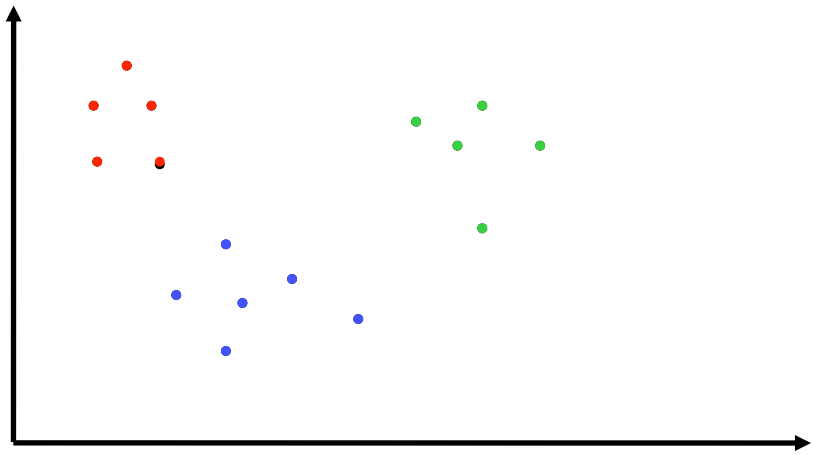
1

What is Clustering?

- Given a **set of points**, with a notion of **distance** between points, **group the points** into some number of ***clusters***, so that
  - Members of the same cluster are close/similar to each other
  - Members of different clusters are dissimilar
- **Usually:**
  - Points are in a high-dimensional space
  - Similarity is defined using a distance measure
  - Euclidean, Cosine, Jaccard, edit distance, …

2

Clustering Example



3

A Different Example

- How would you group
  - 'The price of crude oil has increased significantly'
  - 'Demand for crude oil outstrips supply'
  - 'Some people do not like the flavor of olive oil'
  - 'The food was very oily'
  - 'Crude oil is in short supply'
  - 'Oil platforms extract oil'
  - 'Canola oil is supposed to be healthy'
  - 'Iraq has significant oil reserves'
  - 'There are different types of cooking oil'

A note: you might or might not know how many clusters to look for.

4

A Different Example

- How would you group
  - 'The price of crude oil has increased significantly'
  - 'Demand for crude oil outstrips supply'
  - 'Some people do not like the flavor of olive oil'
  - 'The food was very oily'
  - 'Crude oil is in short supply'
  - 'Oil platforms extract oil'
  - 'Canola oil is supposed to be healthy'
  - 'Iraq has significant oil reserves'
  - 'There are different types of cooking oil'

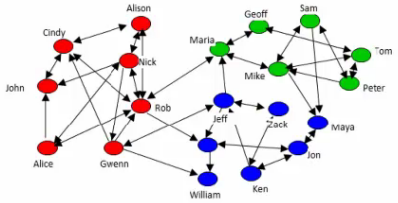
5

Another Example

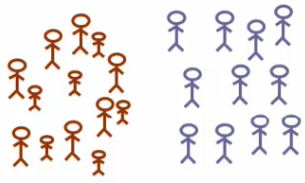
6

Some Example Uses

Organize computing clusters



Social network analysis



Market segmentation

Astronomical data analysis

7

Example: Image segmentation

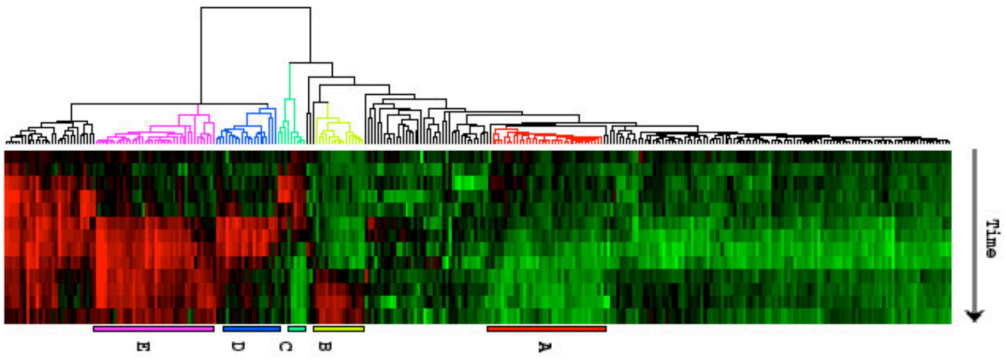
- Goal: Break up an image into perceptually similar regions

Slide: James Hays

8

Example: Genome data

- Goal: Cluster genes that express similar/related traits



Eisen et al., PNAS 1998

9

Sample problem: Music

- **Intuitively: Music can be divided into categories, and customers prefer a few categories**
  - But what are categories really?
- Represent a CD by a set of customers who bought it
- Similar CDs have similar sets of customers, and vice-versa
- But what are the **characteristics** of CDs that we care about?
  - Genre?
  - Popularity?
  - Cover color?

10

Music example

- Think of a space with one dimension for each **customer**
  - Values in a dimension may be 0 or 1 only
  - A CD is a "point" in this space (x_1, x_2,…, x_d), where x_i = 1 iff the i th customer bought the CD
- For Amazon, the dimension is tens of millions
- **Task:** Find clusters of similar CDs

11

Clustering Basics

- Collect examples
- Compute **similarity** among examples according to some metric
- Group examples together such that:
  - Examples within a cluster are similar
  - Examples in different clusters are different
- **Sometimes**: Summarize each cluster
- **Sometimes**: Assign new instances to the most similar cluster

12

Measures of Similarity

- To do clustering we need some measure of similarity.
- This is basically our "critic"
- Computed over a vector of values representing instances
- Types of values depend on domain:
  - Documents: bag of words, linguistic features
  - Purchases: cost, purchaser data, item data
  - Census data: most of what is collected
- Multiple different measures exist

13

Measures of Similarity

- Semantic similarity (but that's hard)
  - For example, olive oil *vs.* crude oil—these are semantically different
- Similar attribute counts
  - Number of attributes with the same value
  - Appropriate for large, sparse vectors
  - Bag-of-Words: BoW
- (Slightly) more complex vector comparisons:
  - Euclidean Distance
  - Cosine Similarity

14

Euclidean Distance

- Euclidean distance: distance between two measures summed across each feature

$$\text{dist}(x_i, x_j) = \text{sqrt}((x_{i1}\text{-}x_{j1})^2+(x_{i2}\text{-}x_{j2})^2+..+(x_{in}\text{-}x_{jn})^2)$$

- Squared differences give more weight to larger differences
  - $\text{dist}([1,2],[3,8]) = \text{sqrt}((1\text{-}3)^2+(2\text{-}8)^2) =$
    $\text{sqrt}((\text{-}2)^2+(\text{-}6)^2) =$
    $\text{sqrt}(4+36) =$
    $\text{sqrt}(40) = \sim 6.3$

15

Euclidean

- Calculate differences
  - Ears: pointy? [T/F → 0/1]
  - Muzzle: how many inches long? [1–4]
  - Tail: how many inches long? [2–6]



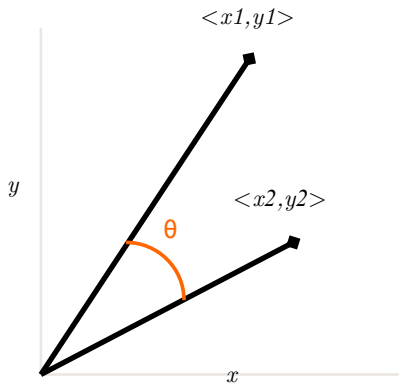

$$\text{dist}(x_1, x_2) = \text{sqrt}((0\text{-}1)^2+(3\text{-}1)^2+..+(2\text{-}4)^2)=\text{sqrt}(9)=3$$

$$\text{dist}(x_1, x_3) = \text{sqrt}((0\text{-}0)^2+(3\text{-}3)^2+..+(2\text{-}3)^2)=\text{sqrt}(1)=1$$

16

Cosine Similarity

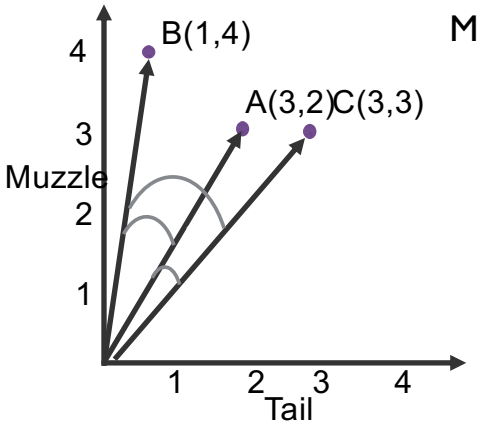
- A measure of similarity between vectors
  - Find **cosine of the angle** between them
  - Cosine = 1 when angle = 0
  - Cosine < 1 otherwise
- As angle between vectors shrinks, θ approaches 1
  - Meaning: the two vectors are getting closer
  - Meaning: the **similarity** of whatever is represented by the vectors **increases**
- Vectors can have any number of dimensions



Based on home.iitk.ac.in/~mfelixor/Files/non-numeric-Clustering-seminar.ppt

17

Cosine Similarity



Most similar?

18

Euclidean Distance vs Cosine Similarity vs Other

- Cosine Similarity:
  - Measures **relative** proportions of various features
  - Ignores magnitude
  - When all the correlated dimensions between two vectors are in proportion, you get maximum similarity
- Euclidean Distance:
  - Measures **actual** distance between two points
  - More concerned with absolutes
- Either can deal with many dimensions
- Often similar in practice, especially on high dimensional data
- Consider meaning of features/feature vectors **for your domain**

Justin Washtell @ semanticvoid.com/blog/2007/02/23/similarity-measure-cosine-similarity-or-euclidean-distance-or-both/

19

What about non-Euclidean space?

- Is the space Euclidean or non-Euclidean?
- In Euclidean:
  - Points are vectors of real numbers, i.e. coordinates
  - It is possible to summarize a collection of points as their average ("centroid")
- Non-Euclidean:
  - There is no notion of location or centroid
  - We summarize a collection of points differently
  - Distance measures: Jaccard, Hamming, cosine

20

Clustering Algorithms

- Flat:
  - K means
- Hierarchical:
  - Bottom up
  - Top down (not common)
- Probabilistic:
  - Expectation Maximization (E-M)

21

Partitioning (Flat) Algorithms

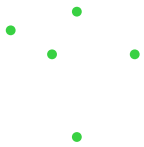
- Partitioning method
  - Construct a **partition** of *n* instances into a set of *k* clusters
- Given: a set of documents and the number *k*
- Find: a partition of *k* clusters that optimizes the chosen partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions.
  - Usually too expensive.
  - Effective heuristic methods: k-means algorithm.

www.csee.umbc.edu/~nicholas/676/MRSslides/lecture17-clustering.ppt

22

k-means Clustering

- Simplest hierarchical method, widely used
- Create clusters based on a centroid; each instance is assigned to the closest centroid
  - Centroid means "approximate center"
- K is given as a parameter
- Heuristic and iterative



23

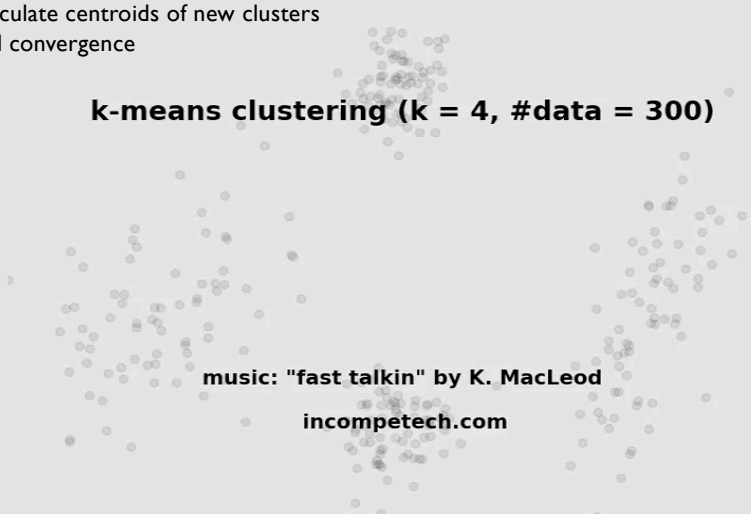
k-means Algorithm

- Choose k (the number of clusters)
- Randomly choose k instances to center clusters on
- Assign each point to the centroid it's closest to, forming clusters
- Recalculate centroids of new clusters
- Reassign points based on new centroids
- Iterate until…
- Convergence (no point is reassigned) or after a fixed number of iterations.

24

k-M

1. randomly place centroids
2. iteratively:
   - assign points to closest centroid, forming clusters
   - calculate centroids of new clusters
3. until convergence



This (happens to be) a pretty good random initialization!

www.youtube.com/watch?v=5I3Ei69I40s

25

k-means

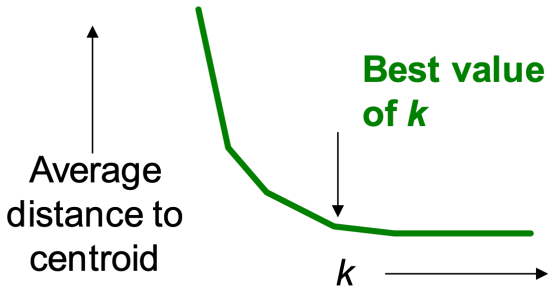
- Tradeoff between having more clusters (better focus within each cluster) and having too many clusters.
  - Overfitting is a possibility with too many!
- Results depend on random seed selection.
  - Some seeds can result in slow convergence or convergence to poor clusters
- Algorithm is sensitive to outliers
  - Data points that are very far from other data points
  - Could be errors, special cases, …

www.csee.umbc.edu/~nicholas/676/MRSslides/lecture17-clustering.ppt

26

Finding *k* automatically

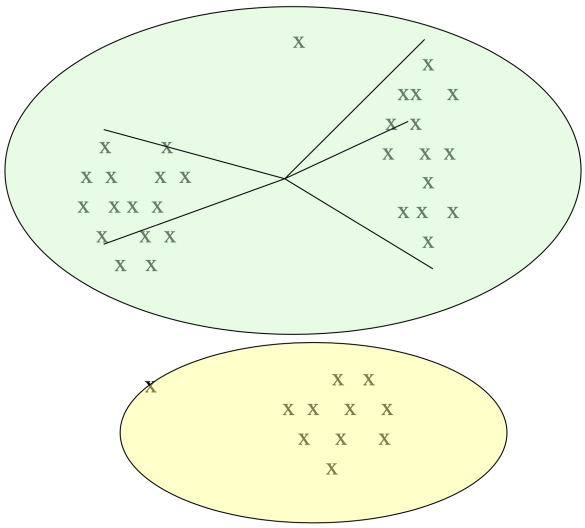
- Try different **k**, looking at the change in the average distance to centroid as **k** increases
- Average falls rapidly until right **k**, then changes little



27

Example: Picking *k*

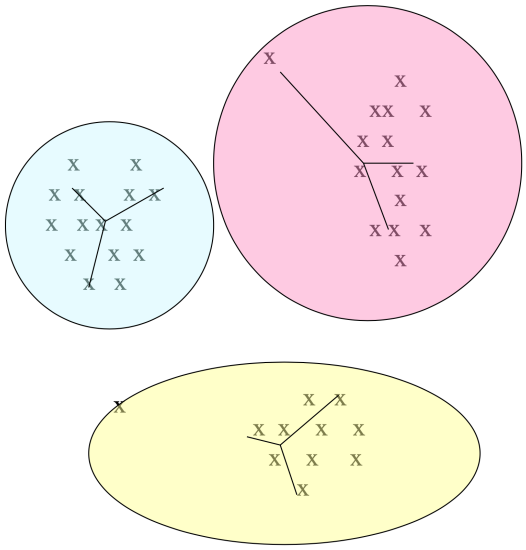
- Too few *k*: many long distances to centroid



28

Example: Picking *k*

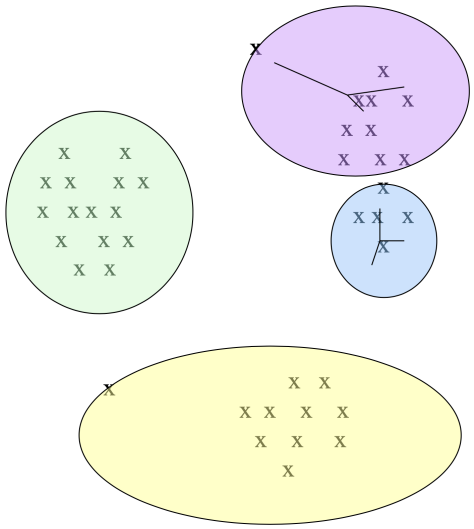
- Just right: Distances are fairly short



29

Example: Picking *k*

- Too many *k*: Minimal improvement in average distance



30

Properties of *k*-means

- Guaranteed to converge in a finite number of iterations
- Running time per iteration:
- Assign data points to closest centroid: O(KN)
- Change the cluster centroid to the average of points: O(N)

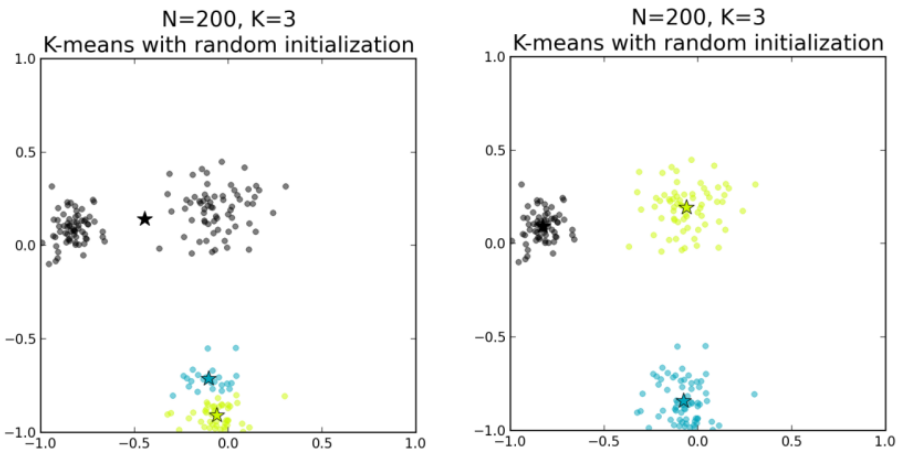
31

Example: *k*-means for image segmentation

- Goal: Partition an image into regions with fairly homogenous visual characteristics

32

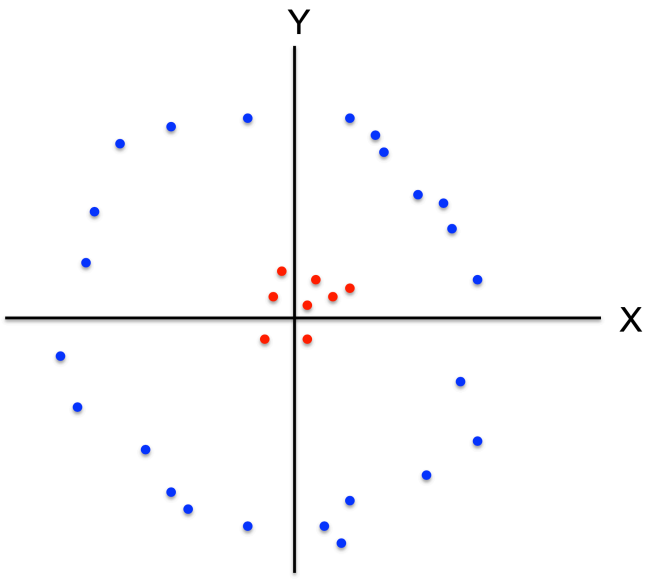
Problem: Bad Initial Seeds



datasciencelab.wordpress.com/2014/01/15/improved-seeding-for-clustering-with-k-means/

33

Problem: *k*-means is a poor choice



34

Evaluation of k-means

- Advantages:
  - Easy to understand, implement
  - Most popular clustering algorithm
  - Efficient, almost linear
    - Time complexity: O(tkn)
      - n = number of data points
      - k = number of clusters
      - t = number of iterations.
  - In practice, **performs well** (especially on text)
- Disadvantages:
  - Must choose *k* beforehand
    - Bad *k* → bad clusters
    - Sometimes we don't know *k*
  - Sensitive to initialization
    - One fix: run several times with different random centers and look for agreement
  - Sensitive to outliers, irrelevant features

35

Expectation Maximization Clustering

- Expectation-Maximization is a core ML algorithm
  - Not just for clustering!
- Basic idea: assign instances to clusters **probabilistically** rather than **absolutely**
  - Instead of assigning membership in a group, learn a probability function for each group
- Instead of absolute assignments, output is probability of each instance being in each cluster

38

EM Clustering Algorithm

- **Goal:** maximize overall probability of data
- Iterate between:
  - Expectation: **estimate probability** that each instance belongs to each cluster
  - Maximization: **recalculate parameters** of probability distribution for each cluster
- Until convergence or iteration limit.

39

Expectation Maximization (EM)

- **Probabilistic method for soft clustering**
- Idea: learn k classifications from **unlabeled** data
- Assumes k clusters:$\{c_1, c_2, \ldots c_k\}$
- "Soft" version of k-means
- Assumes a probabilistic model of categories (such as Naive Bayes)
- Allows computing $P(c_i \mid I)$ for each category, c_i, for a given instance I

40

(Slightly) More Formally

- Iteratively learn **probabilistic categorization model** from **unsupervised data**
- Initially assume random assignment of examples to categories
  - "Randomly label" data
- Learn initial probabilistic model by estimating **model parameters θ** from randomly labeled data
- Iterate until convergence:
  - **Expectation (E-step):**
    - Compute $P(c_i \mid I)$ for each instance (example) given the current model
    - Probabilistically re-label the examples based on these posterior probability estimates
  - **Maximization (M-step):** Re-estimate model parameters, θ, from re-labeled data

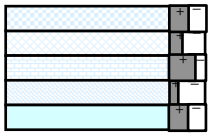
41

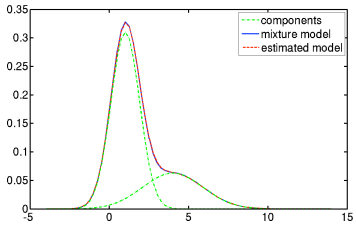
EM

Initialize:

Assign random probabilistic labels to unlabeled data

Unlabeled Examples





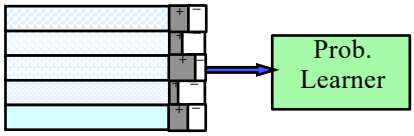
https://www.mathworks.com/matlabcentral/fileexchange/24867-gaussian-mixture-model-m

42

EM

Initialize:

Give soft-labeled training data to a probabilistic learner

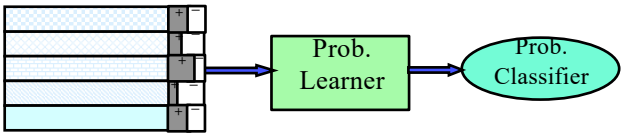


43

EM

Initialize:

Produce a probabilistic classifier

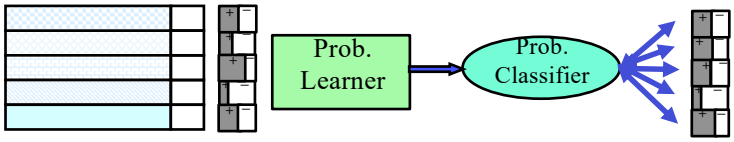


44

EM

E Step:

Relabel unlabled data using the trained classifier



45

EM

M step:

Retrain classifier on relabeled data

Continue EM iterations until probabilistic labels on unlabeled data converge.

46

EM Summary

- Basically a probabilistic k-means.
- Has many of same advantages and disadvantages
  - Results are easy to understand
  - Have to choose k ahead of time
- Useful in domains when we want likelihood that an instance belongs to more than one cluster
  - Natural language processing for instance

47