Clustering

Kenneth (Kenny) Joseph

University at Buffalo
Department of Computer Science
and Engineering
School of Engineering and Applied Sciences

3/23/22

The Midterm

- Cheating
  - 17% of you received an email about potential AI violations. Schedule your meeting within the timeframe, or you will automatically fail the class
  - Punishment will depend on what is determined at the meeting
- Difficulties
  - **If you scored below a 35 on the exam (and did not cheat)**
    - There will be a remediation session at some point after the AI violations have been addressed
    - If you attend and submit corrections to your exam, you will be able to receive up to 35 points

Other Announcements

- Quizzes will change moving forward
  - 1-2 questions, like those on the exam
  - Quiz 7 is out now
- Course Evals
  - I read them!
  - Tough to satisfy all complaints
    - Too fast, too slow
    - Too much math, not enough math
    - Not enough math, not enough coding
  - Things I will address:
    - I will slow down
    - Adding more math sometimes, and more code others

Other Announcements (cont.)

- PA3 is due Friday
  - You may use simpledorf
  - Do not count the final annotation (by 3rd party if the first two annotators disagree) in your calculation
  - There are no right answers. Make a decision on how you’re going to do the labeling, and then use it! Discuss what worked/didn’t in your write-up
    - Love the discussion on Piazza!
  - We will discuss as a class some time next week

Questions/Comments

OK. Moving along. First, a look back

- Evaluation metrics for classification: Precision, Recall, Accuracy, F1, Precision@K
- Evaluation curves: Precision/Recall, ROC
- Annotation: the basics, and measures of agreement

A look forward

- Topics we **must** cover
  - Clustering
    - K-means
    - Gaussian Mixtures & the EM Algorithm
    - Agglomerative clustering
  - Dimensionality reduction w/ PCA
  - Graphical Models
    - Bayesian statistics – the basics
    - Basic interpretation of graphical models
    - Bayesian learning for regression/ Mixture models
  - Deep Learning
    - Backprop
    - Basic models
  - Bias/Fairness
    - Measures
    - Concepts

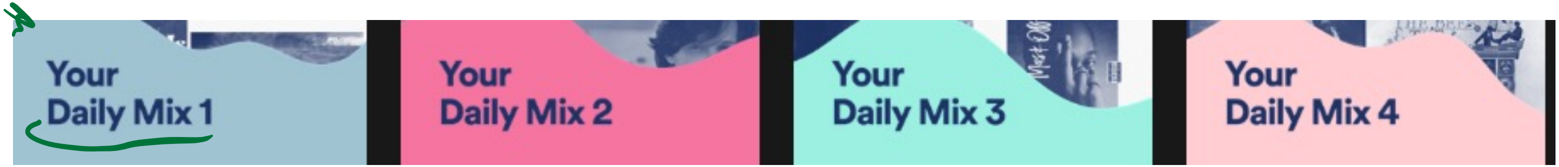
- Topics we **might** cover, in order
  - Kernels
  - SVMs
  - Dimensionality reduction w/ UMAP
  - Causal Inference
  - Non-IID data: HMMs, Networks, Spatiotemporal data
  - Other learning frameworks: Active Learning, RL, Ranking, human-in-the-loop

A look forward (cont.)

- Left:
  - 4 Quizzes
    - Remember, new format
  - 2 Programming Assignments
    - PA4: Unsupervised Learning (Dimensionality Reduction and Clustering)
    - PA5: Deep Learning
  - 1 Final
    - Same as midterm, except aforementioned concerns will be addressed.

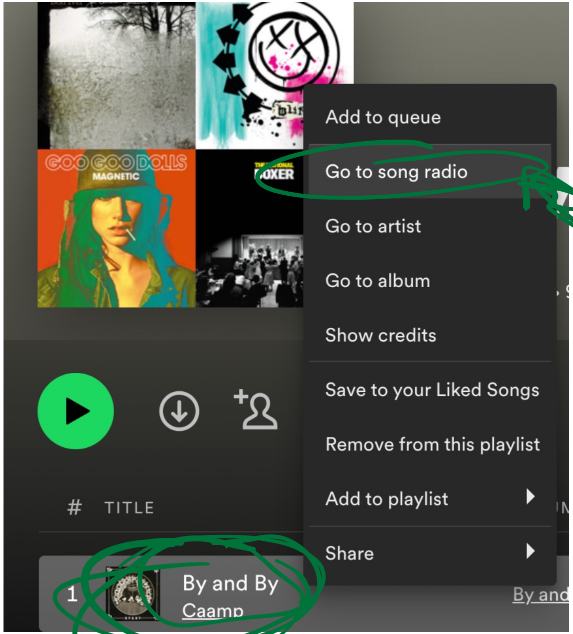
“Unsupervised” Learning

- High level – machine learning when you don’t have labels
- Good example: Spotify Daily Mixes



- How might you implement this?

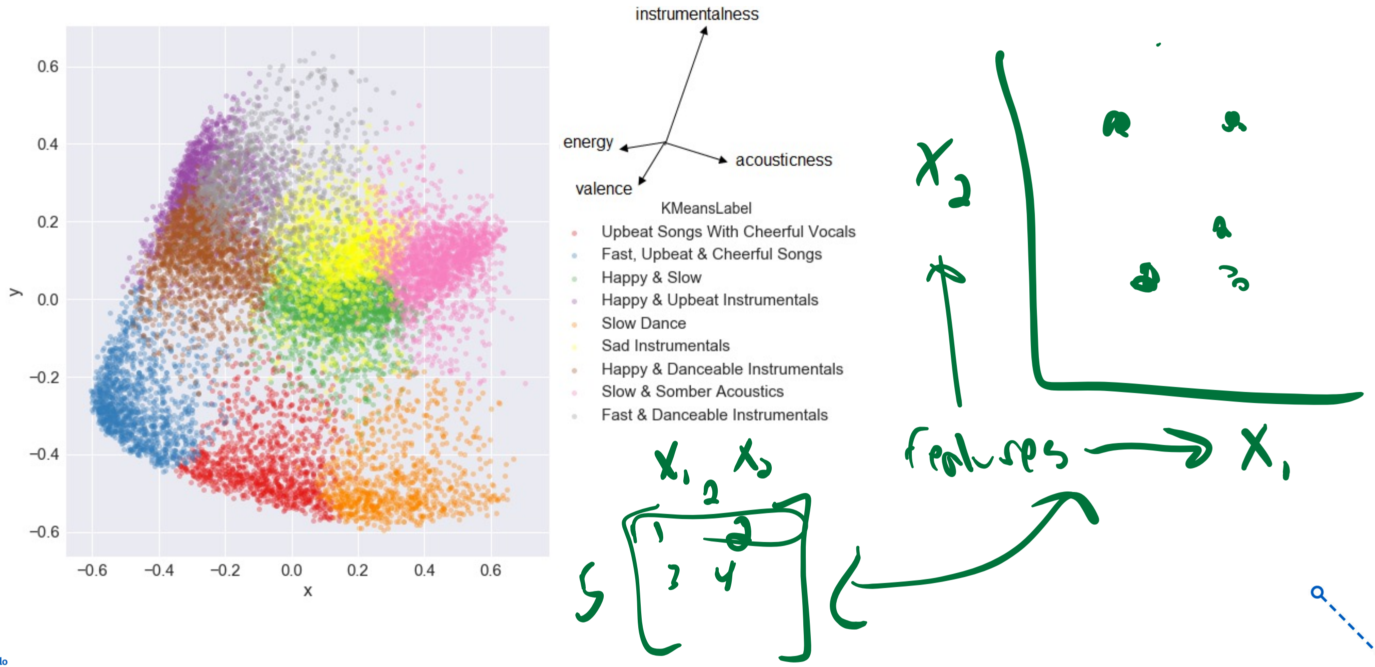
Supervised vs. Unsupervised is a Grey Zone



Add to queue
Go to song radio
Go to artist
Go to album
Show credits
Save to your Liked Songs
Remove from this playlist
Add to playlist
Share

- Supervised or unsupervised?
- Also:
  - *semi-supervised*
  - *Distantly supervised*

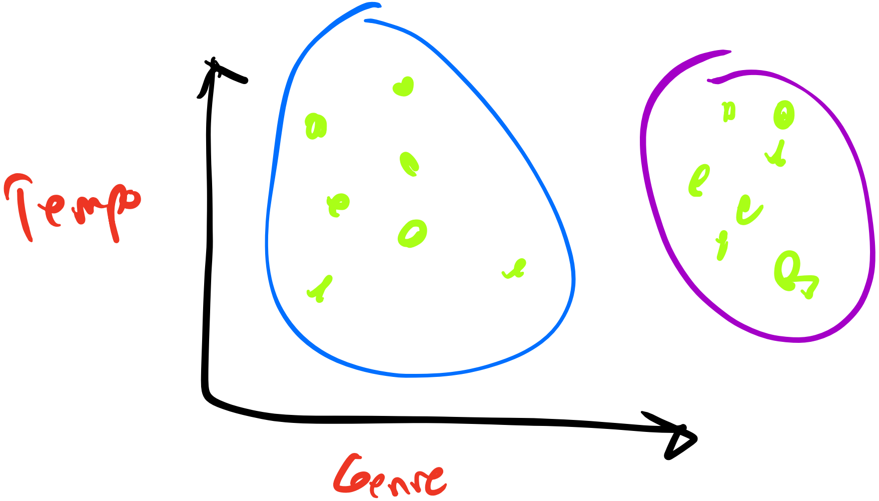
Reminder: We can treat our data as points in space



Our Class

- We will consider two forms of what is traditionally known as unsupervised learning:
  - C **Clustering** – You give me a set of points, I tell you how they fall into a set of groups
  - £ **Dimensionality Reduction** – You give me a set of points in D dimensions, I return to you data in K dimensions, K << D, where those points retain “similar content” to the points in D dimensions

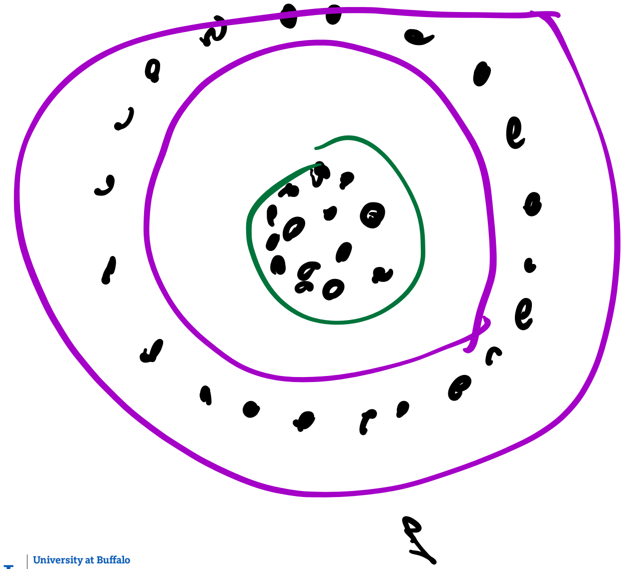
Visual Intuition: Clustering



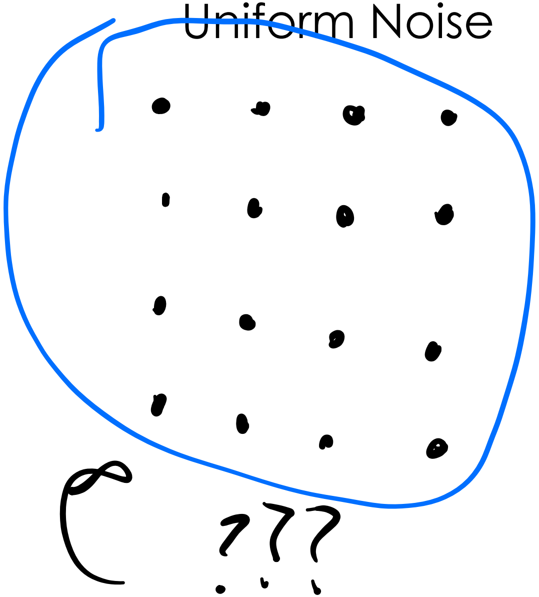
- Basic idea: Find the best partition of a set of points into a smaller set of groups
- What is a **cluster?**
- Intuitively, a set of points. Can also think of it as an area in the feature space

Quiz: What is the best clustering?

Concentric Circles

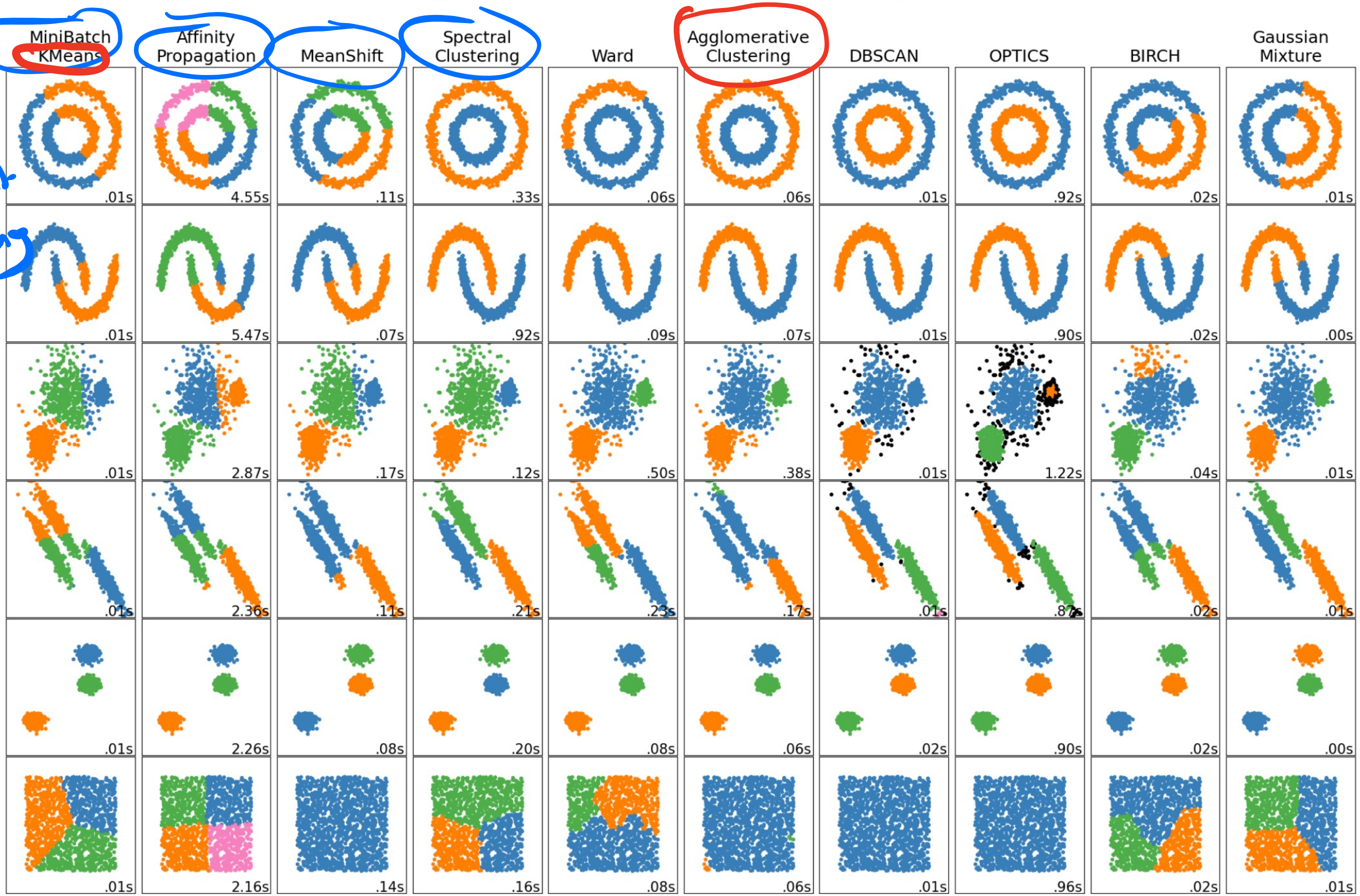


Uniform Noise

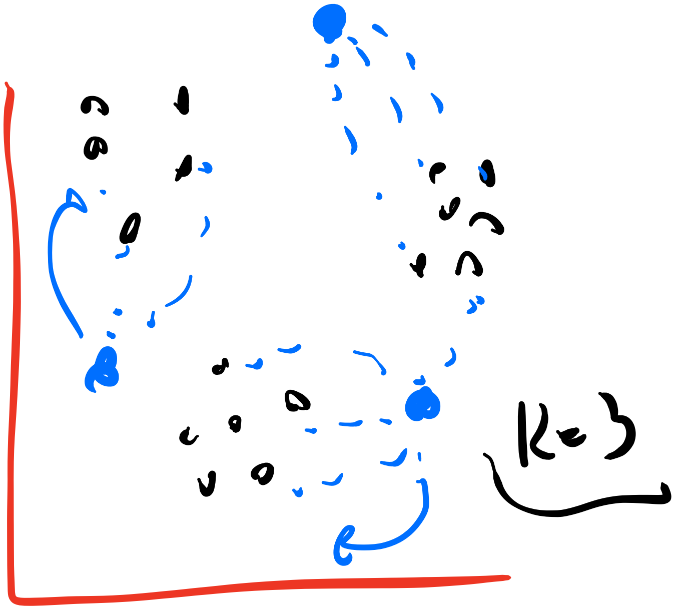


How do you evaluate your clustering alg (relative to other algs?)

Different clustering alg.



Today: Clustering with the K-Means Algorithm



- A 3-step Algorithm
  1. Initialize a set of *k* cluster centers

Iterate until convergence

2. Assign each point to the closest center
3. Update the position of the centers

Example w/ Visual Tool

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

- Building Intuitions
  - How might you figure out the right k?
  - What dataset does kmeans not work well on? Why?
  - What is the benefit of assigning clusters vs random initialization?
  - What might be a good way to assign clusters?

Formally

$\sqrt{(x_n - \mu_n)^2}$

where is the nth point

$\rightarrow r_{10} = 1$

$\rightarrow r_{11} = 0$

$\rightarrow r_{12} = 0$

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

where is the kth center

nth point

kth cluster; whether or not this point is in this cluster

Step 1: $r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$

Step 2:

$$2 \sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0$$

for $\boldsymbol{\mu}_k$ to give

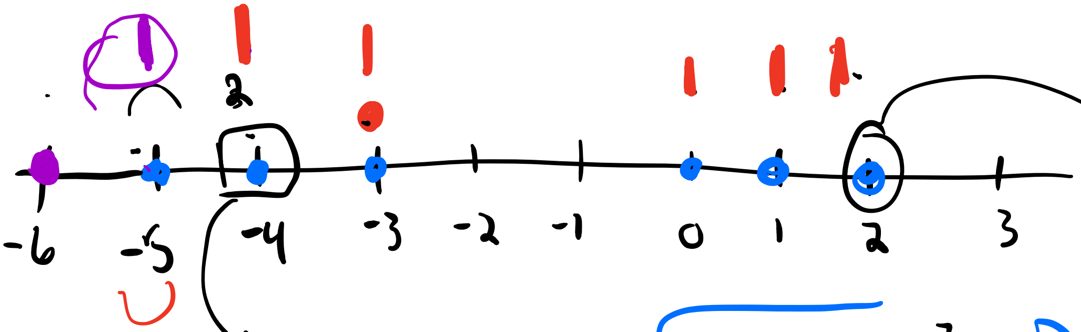
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}.$$

Hand-Drawn Example

$$\sum_{n}^{N} \sum_{k}^{K} r_{nk} \|x_n - \mu_k\|^2$$

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|x_n - \mu_j\|^2 \\ 0 & \text{else} \end{cases}$$

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$



Distance to ● $\sqrt{(-4 - -6)^2} = 2$

" " ● $\sqrt{(-4 - -3)^2} = 1$

$r_{2●} = 0$

$r_{2●} = 1$

$D_{●} = \sqrt{(2 - -6)^2}$

$= 8$

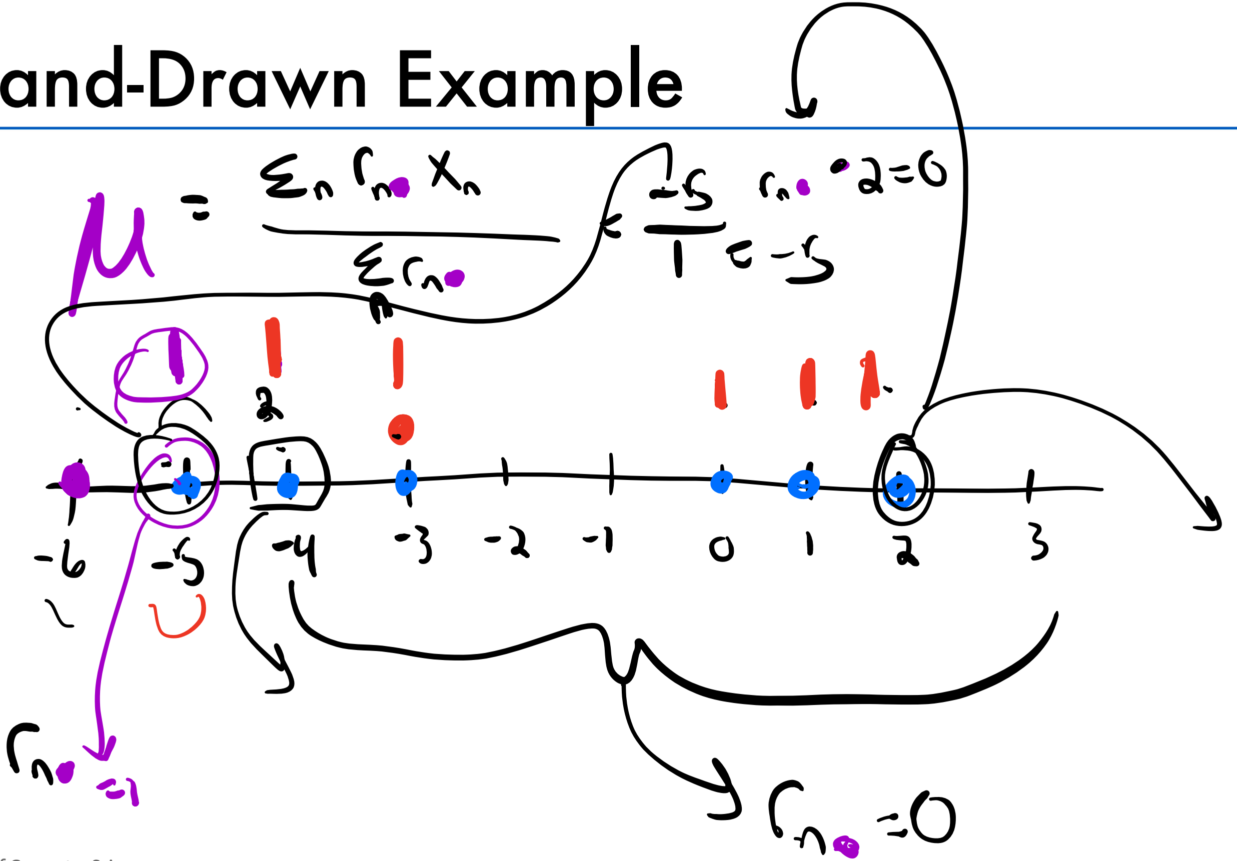
$D_{●} = \sqrt{(2 - -3)^2}$

$= 5$

$r_{6●} = 0$

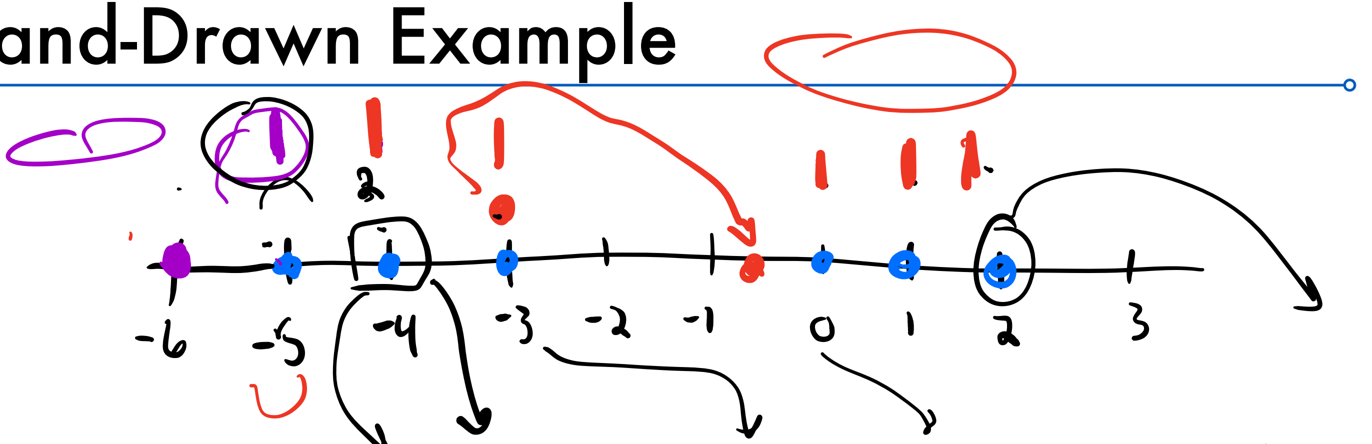
$r_{6●} = 1$

Hand-Drawn Example



$\mu = \frac{\Sigma_n r_{n\bullet} x_n}{\Sigma r_{n\bullet}}$
$\frac{-5}{1} = -5$
$r_{n\bullet} \cdot 2 = 0$
2
-6
-5
-4
-3
-2
-1
0
1
2
3
$r_{n\bullet} = 1$
$r_{n\bullet} = 0$

Hand-Drawn Example



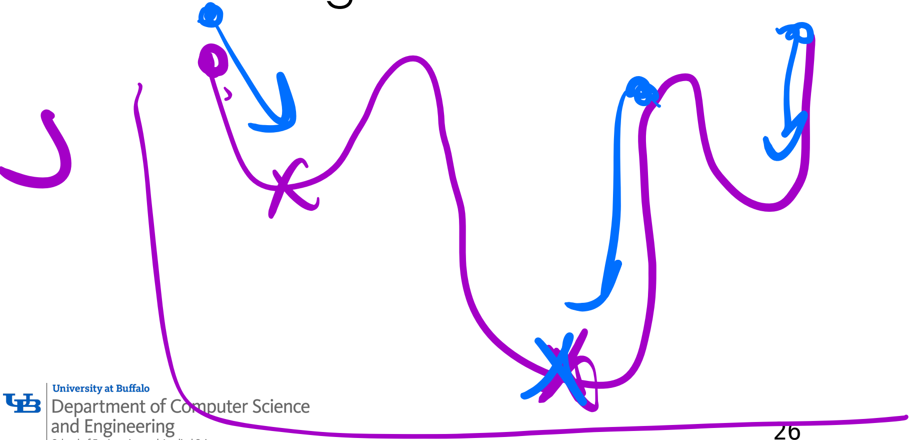
$$\mu = \frac{0 \cdot -5 + \underbrace{1}_{r_0=1} \cdot -4 + 1 \cdot -3 + 1 \cdot 0 + 1 \cdot 1 + 1 \cdot 2}{5}$$

$$= -.8$$

Other Important Concepts - Convergence

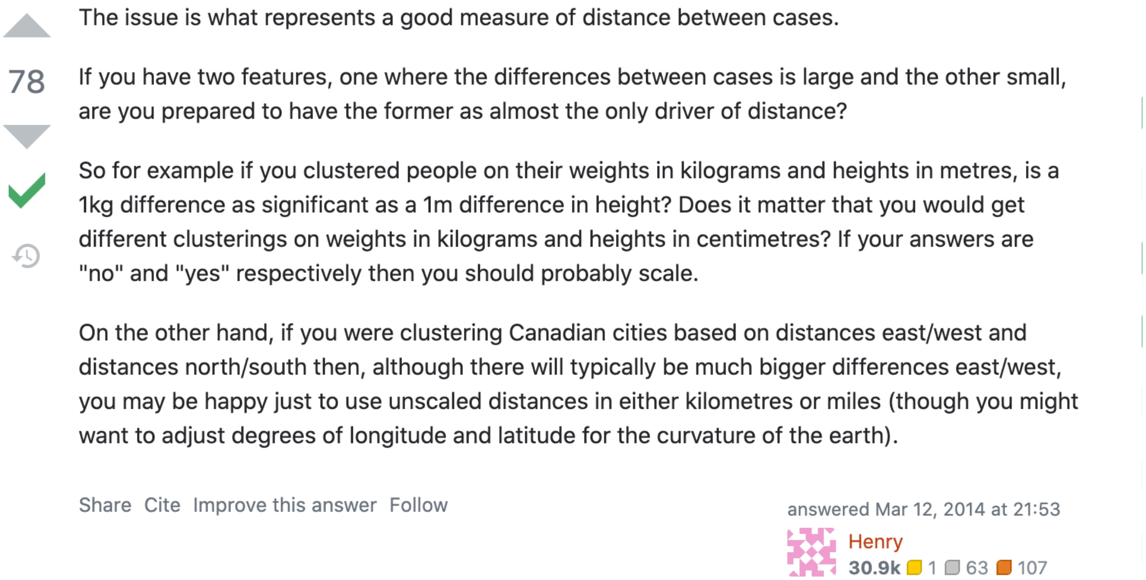
- Does this converge?
  - Yes! But only to **local minima**
  - More in the code example

- What might we do to address our local minima problem?



Other Important Concepts – Feature Scaling

- Important to scale your features! Why?



78

The issue is what represents a good measure of distance between cases.

If you have two features, one where the differences between cases is large and the other small, are you prepared to have the former as almost the only driver of distance?

So for example if you clustered people on their weights in kilograms and heights in metres, is a 1kg difference as significant as a 1m difference in height? Does it matter that you would get different clusterings on weights in kilograms and heights in centimetres? If your answers are "no" and "yes" respectively then you should probably scale.

On the other hand, if you were clustering Canadian cities based on distances east/west and distances north/south then, although there will typically be much bigger differences east/west, you may be happy just to use unscaled distances in either kilometres or miles (though you might want to adjust degrees of longitude and latitude for the curvature of the earth).

Share Cite Improve this answer Follow

answered Mar 12, 2014 at 21:53

Henry

30.9k ● 1 ● 63 ● 107

Other Important Concepts – Non-Euclidean Distance

$$\widetilde{J} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\mathcal{V}(\mathbf{x}_n, \boldsymbol{\mu}_k)$$

Kernelized KMeans

$$\widetilde{J} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\mathcal{V}(\mathbf{x}_n, \boldsymbol{\mu}_k)$$

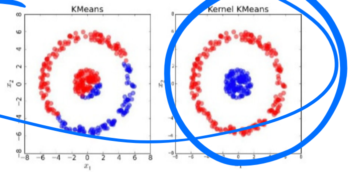
https://cse.iitk.ac.in/users/piyush/courses/ml_autumn16/771A_lec10_slides.pdf

- Basic idea: Replace the Euclidean distance/similarity computations in K-means by the kernelized versions. E.g., $d(\boldsymbol{x}_n, \boldsymbol{\mu}_k) = ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||$ by

$$\begin{aligned} ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||^2 &= ||\phi(\boldsymbol{x}_n)||^2 + ||\phi(\boldsymbol{\mu}_k)||^2 - 2\phi(\boldsymbol{x}_n)^\top\phi(\boldsymbol{\mu}_k) \\ &= k(\boldsymbol{x}_n, \boldsymbol{x}_n) + k(\boldsymbol{\mu}_k, \boldsymbol{\mu}_k) - 2k(\boldsymbol{x}_n, \boldsymbol{\mu}_k) \end{aligned}$$

- Here $k(.,.)$ denotes the kernel function and ϕ is its (implicit) feature map
- Note: ϕ doesn't have to be computed/stored for data $\{\boldsymbol{x}_n\}_{n=1}^N$ or the cluster means $\{\boldsymbol{\mu}_k\}_{k=1}^K$ because computations only depend on kernel evaluations

Kernel K-means vs. K-means



Pyclust: Open Source Data Clustering Pckage

- A small technical note: When computing $k(\boldsymbol{\mu}_k, \boldsymbol{\mu}_k)$ and $k(\boldsymbol{x}_n, \boldsymbol{\mu}_k)$, remember that $\phi(\boldsymbol{\mu}_k)$ is the average of ϕ's the data points assigned to cluster k