Introduction to Statistical Machine Learning
CSC/DSCC 265/465

Lecture 12: Unsupervised Learning – Part III

Cantay Caliskan

Notes and updates

Notes and updates

- Any questions?
- No Problem Set due next week
- Midterm review: Next week (Wednesday, March 2, 2022)

Plan for the next lectures

- *Kmeans*
- *DBSCAN*
- *OPTICS*
- ***GMM***
- *PCA*

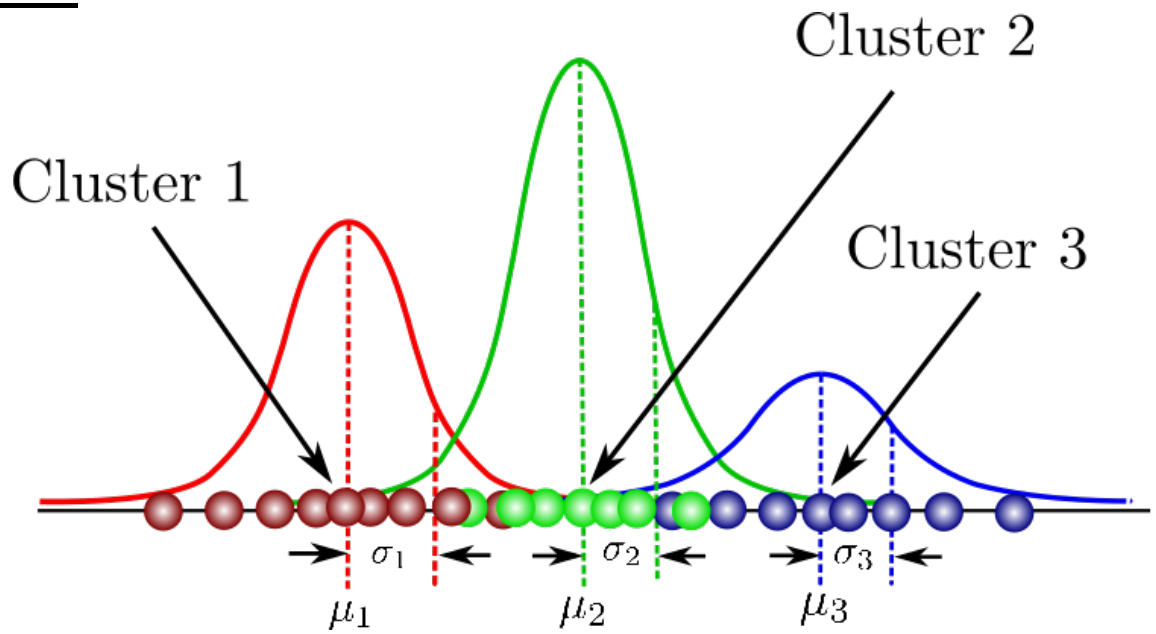
An example of distribution-based clustering: GMM

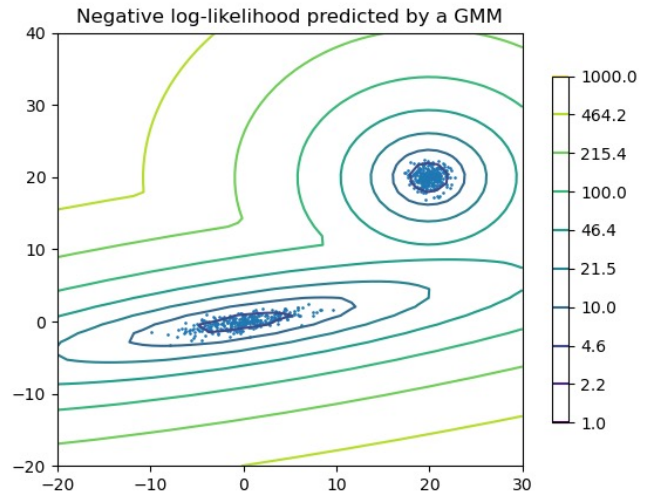
Reminder: Types of Clustering

- Goal: Find the underlying (latent) structures in a dataset
  - Question: How can we do that?
  - **Centroid-Based Clustering**: Find a center and associate all points with a center
  - **Distribution-Based Clustering**: Cluster points by the distributional differences in the data-generating function
  - **Density-Based Clustering**: Cluster points by the density of feature vectors
  - **Hierarchical Clustering**: Cluster points by building a tree

GMM: Gaussian Mixture Models

- <u>Idea</u>: Can we model ***K*** many clusters by using ***K*** many Gaussian distributions? (Dempster, 1977)
- And, can we do this by using Maximum Likelihood Estimation (***MLE***)?
- And, can we group data points belonging to a single distribution together?
- <u>Answer</u>: Yes!

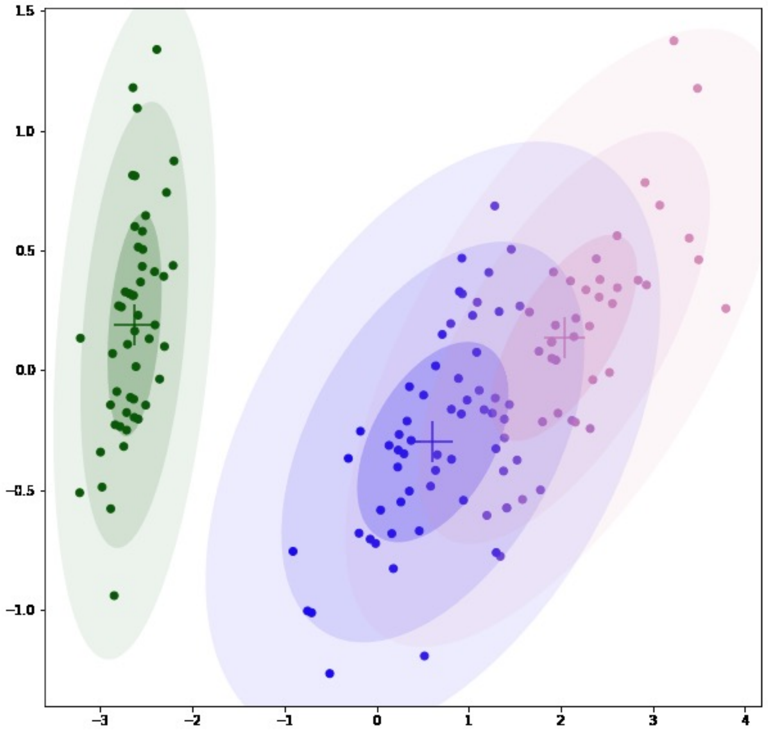




UNIVERSITY of ROCHESTER

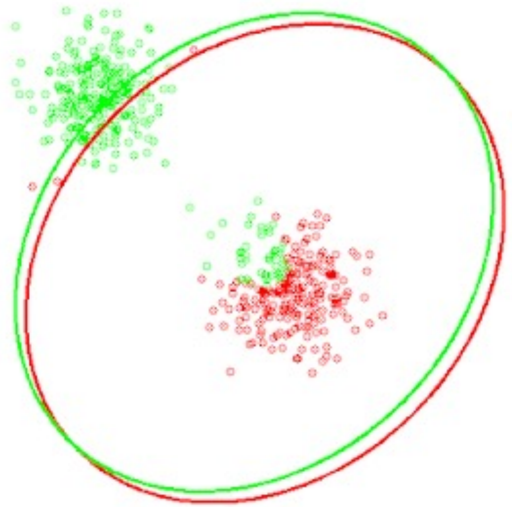
GMM: Summary

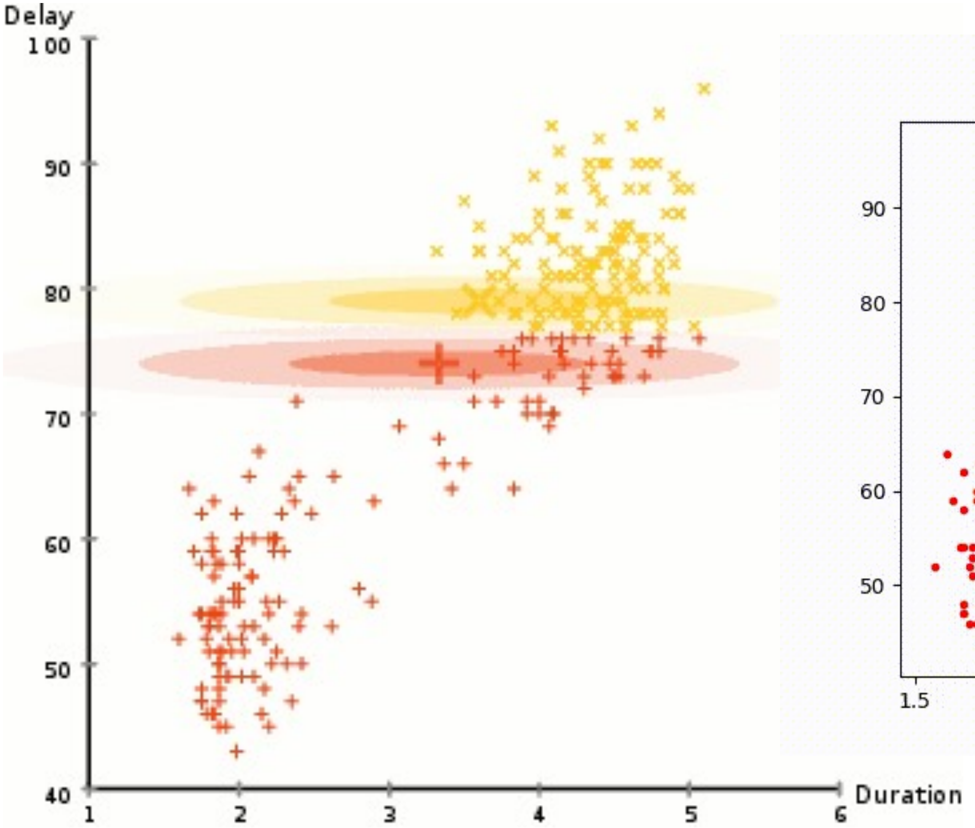
- Specify the number of clusters (***k***) in advance
- Initialize the parameter estimates according to their ***priors***
- Run an Expactation-Maximization (***EM***) algorithm until convergence

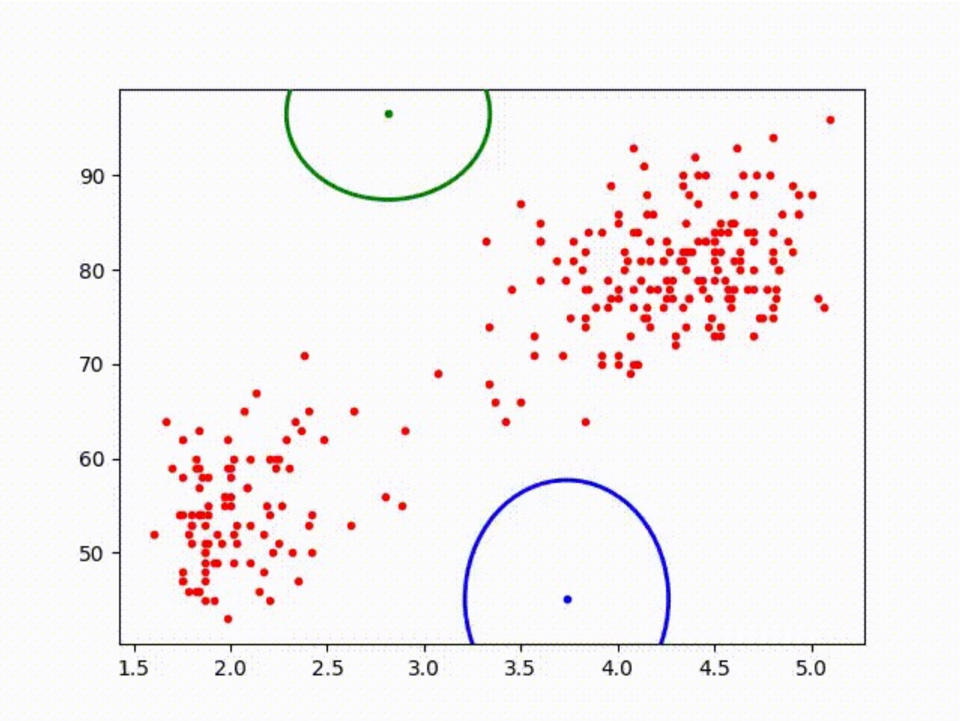


- <u>Advantages:</u>
  - Simple
  - Allows classification of new points
- <u>Disadvantages:</u>
  - Must know the number of clusters in advance
  - Errors in cluster assignments propagate

GMM-Clustering







GMM-Clustering

- Definition: A ***probabilistic*** model that assumes that all data points are generated from a ***mixture*** of a finite number of ***Gaussian*** distributions with unknown parameters

- Idea: Let's say we have an n-dimensional feature vector ***X*** = [X_1, X_2, … , X_n] where each dimension is represented by an **i.i.d** random variable coming from an *unknown* distribution
  - **i.i.d**: independent and identically distributed

- ***GMM*** demands a specific form of density:

$$p(x) = \sum_{j=1}^{k} \pi_j \phi(x; \mu_j, \Sigma_j)$$

Question: What do we have here?

GMM: Technical Background

$$p(x) = \sum_{j=1}^{k} \pi_j \phi(x; \mu_j, \Sigma_j)$$

- Idea: This is a mixture of k component multivariate Gaussian distributions:

$$\phi(x; \mu_j, \Sigma_j) = \frac{1}{|2\pi\Sigma_j|^{1/2}} \exp\left(-\tfrac{1}{2}(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)\right)$$

A multivariate Gaussian density with unknown parameters

- where we have the unknown parameters: (μ_j, Σ_j)
- and the unknown probability of selecting component *j*: π_j $\quad \sum_{j=1}^{k} \pi_j = 1$
- GMM has the same representation as a generative model:

$$z_i \overset{\text{iid}}{\sim} \text{Mult}(\pi, 1)$$
$$x_i | z_i \overset{\text{ind}}{\sim} \mathcal{N}(\mu_{z_i}, \Sigma_{z_i})$$

- z_i represents the latent class

This is a ***probabilistic approach***

UNIVERSITY of ROCHESTER

Clustering with GMM

- Idea: GMM with known parameters defines a joint distribution over (x_i, z_i)
- Using the Bayes' Formula, we have: $p(z_i = j|x_i) \;=\; \dfrac{p(z_i = j)p(x_i|z_i = j)}{p(x_i)}$

Question: What does this formula mean?

- Before we observe x_i, we have the belief that it belongs to cluster j with probability π_j ; after observing x_i we update this belief in accordance with the likelihood of x_i
- Question: What does this imply?
  - We have to run ***Maximum Likelihood Estimation (MLE)***
  - We have to make a probabilistic inference
- Question: Is this soft clustering or hard clustering?
- Answer: Depends on how you make the assignment (raw probability vs. argmax)

Clustering with GMM

- What is left: We don't know the parameters of the probability distribution(!)

$$p(z_i = j|x_i) = \frac{p(z_i = j)p(x_i|z_i = j)}{p(x_i)} = \frac{\pi_j \phi(x_i; \mu_j, \Sigma_j)}{\sum_{l=1}^{k} \pi_l \phi(x_i; \mu_l, \Sigma_l)}$$

Here are the ***parameters*** to estimate

- Let's derive the ***maximum likelihood estimates*** (MLEs) of GMM parameters:

$$\sum_{i=1}^{n} \log(p(x_i)) = \sum_{i=1}^{n} \log(\sum_{j=1}^{k} \pi_j \phi(x_i; \mu_j, \Sigma_j))$$

- Goal: Maximize ***log-likelihood*** (example: when ***k = 1***):

$$\sum_{i=1}^{n} \log(\phi(x_i; \mu_j, \Sigma_j)) = \sum_{i=1}^{n} \left[-\frac{1}{2}(x_i - \mu_1)^T \Sigma_1^{-1}(x_i - \mu_1) - \log|2\pi\Sigma_1|^{1/2} \right]$$

$$(\mu_1^*, \Sigma_1^*) = (\frac{1}{n}\sum_{i=1}^{n} x_i, \frac{1}{n}\sum_{1=1}^{n}(x_i - \mu_1^*)(x_i - \mu_1^*)^T)$$

Question: Do we need to run 'gradient descent' for this solution?

Answer: No, this is a closed-form solution.

UNIVERSITY of ROCHESTER

Clustering with GMM

- What happens when k > 1?

$$\sum_{i=1}^{n} \log(p(x_i)) = \sum_{i=1}^{n} \log(\sum_{j=1}^{k} \pi_j \phi(x_i; \mu_j, \Sigma_j))$$

This cannot be simplified to have closed-form solutions.

(1) We can use ***numerical optimizers***

- Examples: Gradient descent, line search, stochastic optimization etc.

(2) We can use the ***expectation-maximization algorithm (EM)***

Let's take a look at ***expectation-maximization algorithm (EM)!***

Expectation-Maximization Algorithm (EM)

- **Step 1**: Initialize the parameters arbitrarily $\pi, (\mu_{1:k}, \Sigma_{1:k})$

Probabilities associated with cluster membership, distribution *means* and *variances* for each cluster

- **Step 2 (Expectation step – *E*)**: Compute soft class memberships, given the current parameters: $\tau_{ij} = P(z_i = j | x_{ij}, \pi, (\mu_\ell, \Sigma_\ell))$.

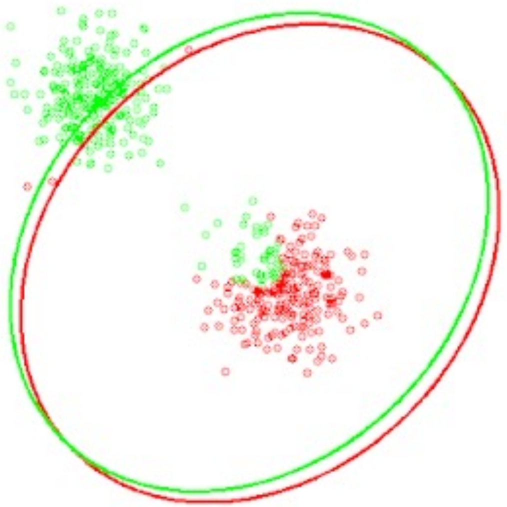
- **Step 3 (Maximization step – *M*)**: Update the parameters by plugging in τ_{ij} (our guess) for the unknown probability of $z_i = j$ which gives us:

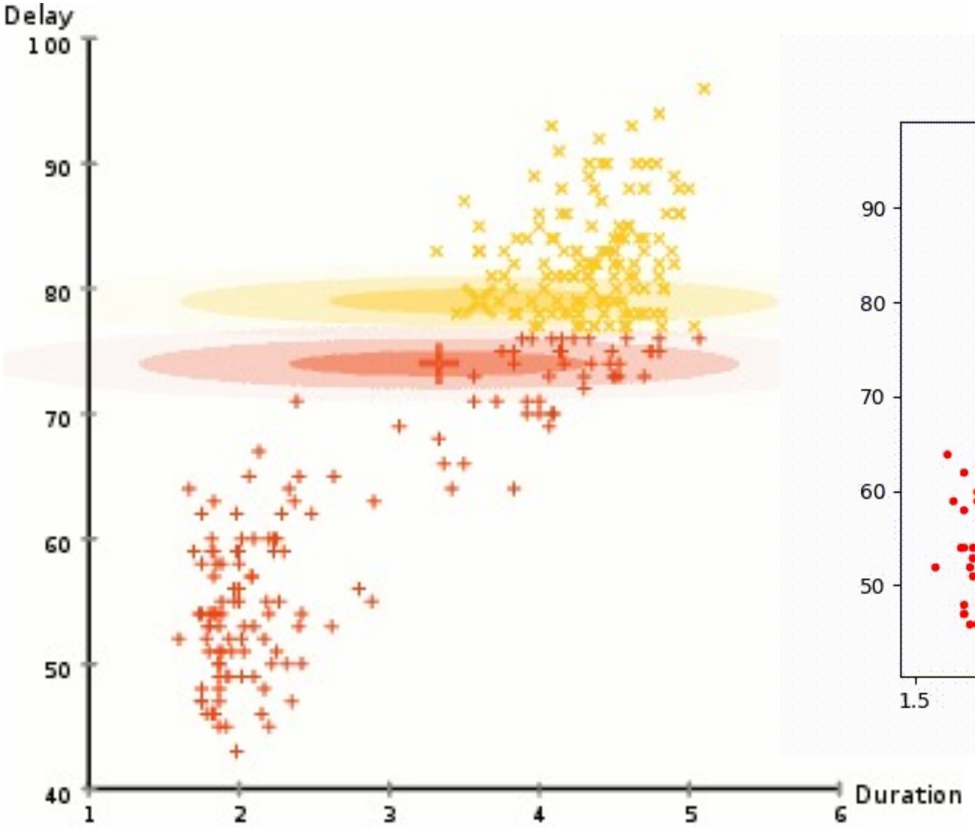
$$\pi_j = \frac{1}{n} \sum_{i=1}^{n} \tau_{ij} \qquad \mu_j = \frac{\sum_{i=1}^{n} \tau_{ij} x_i}{\sum_{i=1}^{n} \tau_{ij}} \qquad \Sigma_j = \frac{\sum_{i=1}^{n} \tau_{ij} (x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^{n} \tau_{ij}}$$

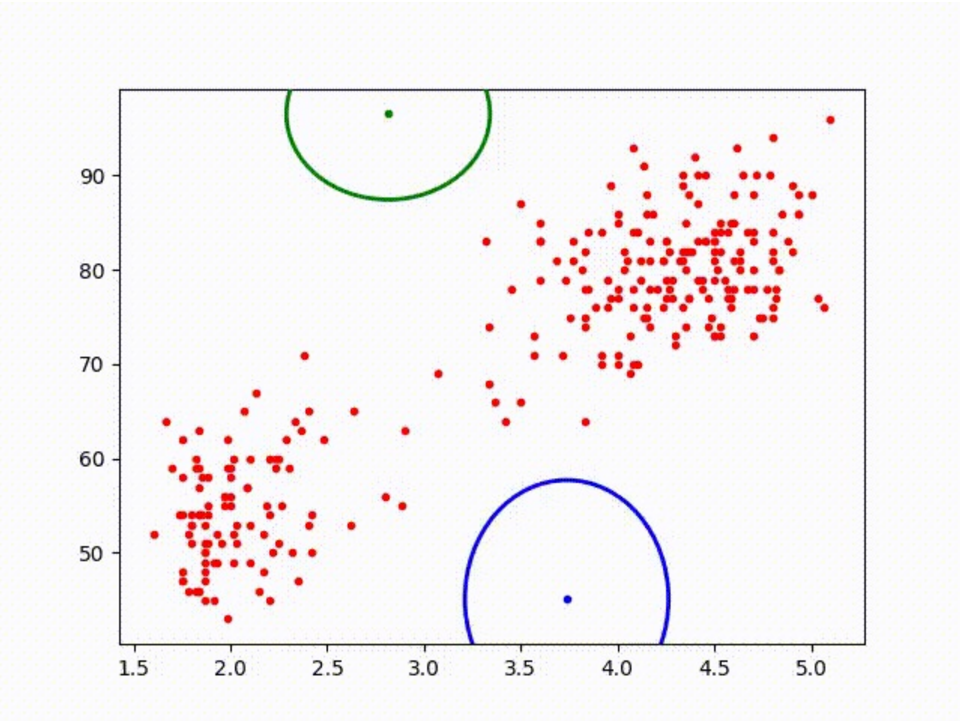
  - Now, each x_i is partially assigned to each cluster j through the conditional probability that $z_i = j$.

  - Question: How is this similar to K-Means?

GMM-Clustering







GMM vs. KMeans

- **Kmeans:**
  - We use an algorithm called Lloyd’s algorithm
  - Makes (usually) hard assignments in each iteration
    - Each point assigned to one class

- **GMM:**
  - We model the mixture proportions and covariance structure in the data
  - Makes (usually) soft assignments in each iteration
    - Each point is assigned a probability of belonging to a cluster

Questions: GMM

- **Does it converge?** ⟵ **Yes**
- **Do we know the rates of convergence?** ⟵ **Depends on the density of data**
- **Are the solutions optimal?** ⟵ **Usually sub-optimal**
- **Can we bound the sub-optimality of the solutions?** ⟵ **Yes**
- **What is its relationship to the maximum likelihood estimation?** ⟵ **Find parameters to maximize the likelihood of our data**

Plan for the next lectures

- *Kmeans*
- *DBSCAN*
- *OPTICS*
- *GMM*
- ***PCA***

Dimensionality Reduction

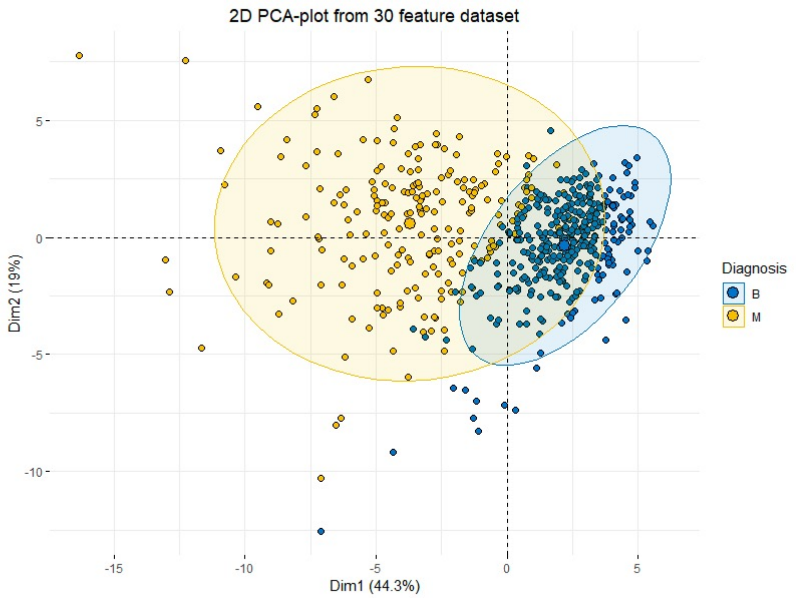
Dimensionality Reduction

- Definition: Transformation of data from a ***high-dimensional space*** into a ***low-dimensional*** space
  - Goal: Low-dimensional space should retain some meaningful properties of the original data

  - **Idea: High-dimensional data may be 'problematic'**
    - ***Curse of dimensionality***
    - But: Many domains are using high dimensional data

Dimensionality Reduction

- <u>Motivation for dimensionality reduction:</u>
  - Sparse data (= a lot of zeros or NA’s)
  - (High/extremely high) number of features
  - Highly correlated/redundant features
  - Noisy features
  - Features that are hard to describe
  - The need to know which feature is important
  - Interpretation / visualization
  - Computational burden
  - ***Curse of dimensionality***

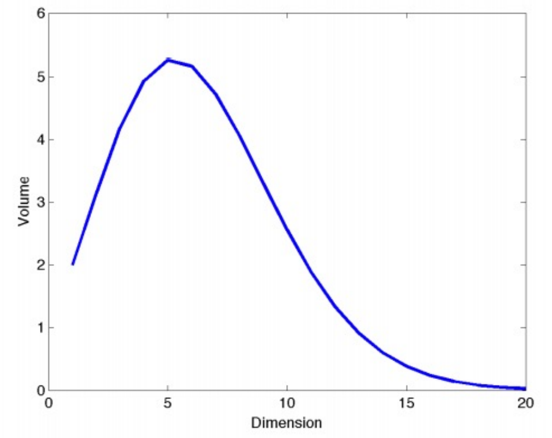


The Curse of Dimensionality

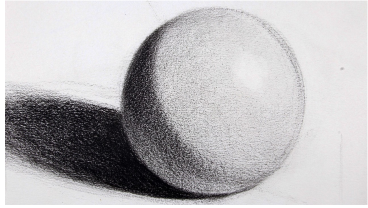
- ***"Many algorithms that work fine in low dimensions become intractable when the input is high-dimensional."***

Bellman, 1961

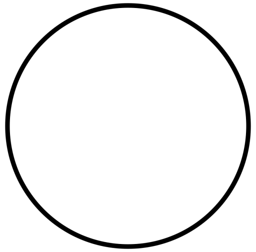
- Refers to various phenomena that arise when analyzing and organizing data in *high-dimensional spaces*
  - Original definition: As dimensionality increases, the volume of the space increases so fast that the available data become sparse. Sparsity is a problem for any method that requires statistical signific
  - Example: Volume of a unit sphere in higher dimensions is **zero**.
  - Another example: #observations vs. #features



Volume of a Unit Sphere



- Question: How do we calculate the ***volume*** of a sphere in **3D**?
- Formula: **V =** $\frac{4}{3}\pi r^3$



- Question: How do we calculate the ***volume*** of a sphere in **2D**?
- Formula: **V =** $(1) * \pi r^2$

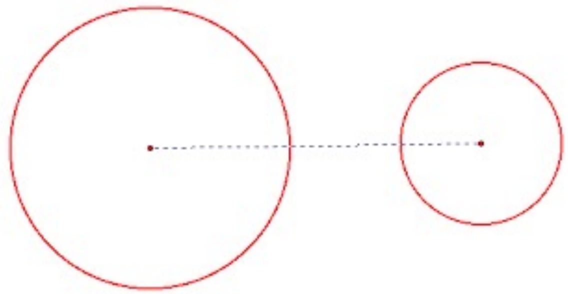
- Question: How do we calculate the ***volume*** of a sphere in **1D**?
- Formula: **V =** $(2) * r^1$

- Question: How do we calculate the ***volume*** of a sphere in **n-D**?
- Formula: **V = Constant*** r^n

- Question: What observation can we make here?

Volume of a Unit Sphere

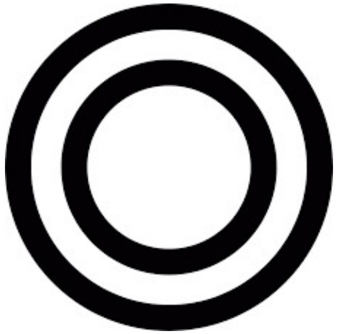
- <u>Observation</u>: What if we find the **difference in volume** between two spheres?
  - Let's say $\mathbf{r_1 = 2}$ and $\mathbf{r_2 = 1}$



- <u>Question</u>: The **difference in volume** in **1D**?
- <u>Formula</u>: **V-diff = 2 $\ast$ 2 - 2 $\ast$ 1 = 2**

- <u>Question</u>: The **difference in volume** in **2D**?
- <u>Formula</u>: **V-diff =** $\boldsymbol{\pi 2^2 - \pi 1^2 = 3\,\pi}$

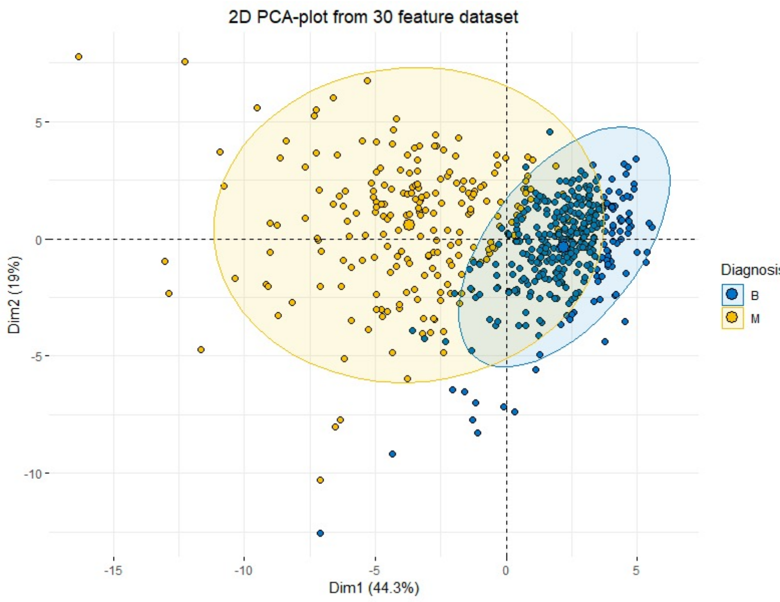
- <u>Question</u>: The **difference in volume** in **3D**?
- <u>Formula</u>: **V-diff =** $\boldsymbol{\frac{4}{3}\pi 2^3 - \frac{4}{3}\pi 1^3 = \frac{28}{3}\pi = 9.33\,\pi}$



- <u>Question</u>: What happens when n -> ∞?
- <u>Answer</u>: Volume concentrates around the ring between two spheres(!)

Dimensionality Reduction

- <u>Different techniques:</u>
  - Missing Values Ratio
  - Low Variance Filter
  - High Correlation Filter
  - Random Forests / Ensemble Trees
  - Backward Feature Elimination
  - Forward Feature Construction
  - Linear Discriminant Analysis (LDA)
  - Generalized Discriminant Analysis (GDA)
  - T-Stochastic Neighbor Embedding (t-SNE)
  - **Principal Component Analysis (PCA)**



Big and High-Dimensional Data

- **Document classification**
  - Features per document =
    - Thousands of words / unigrams
    - Millions of bigrams
    - *Possibly*: billions of trigrams

- **Surveys - Netflix**
  - 524564 users x 5579 movies and TV shows

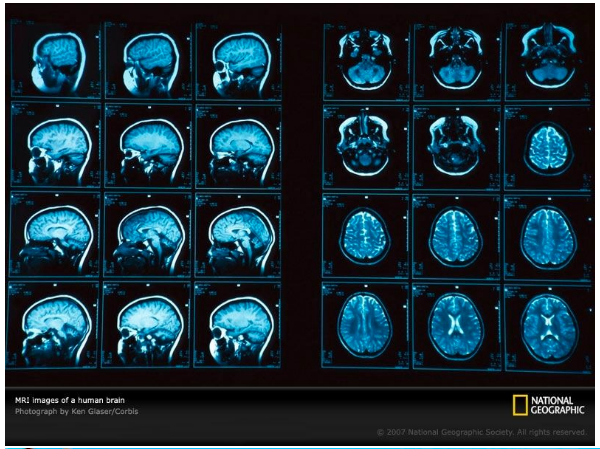
Users	M1	M2	M3	M4
U1	2	4	3	1
U2	0	0	4	4
U3	3	2	2	3
U4	2	?	3	?

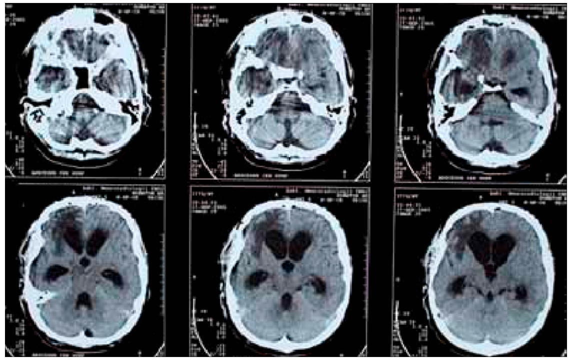
Users may watch a movie more than once.

Big and High-Dimensional Data

- **MEG Brain Imaging**
  - 120 locations x 500 time points x 20 objects
- **Image Data**
  - Full HD: 1920 x 1080
  - QHD: 2560 x 1440
  - UHD: 3840 x 2160

Any image data is high-dimensional data





Big and High-Dimensional Data

- **Spam or not spam**
  - Title
  - Content of the e-mail
  - Template

love Sandra love_sandra_05@yahoo.com wrote:

Hello,
how are you doing today hope all is well with you, now i want to
let you know that i have tell my company that i have
saw a room but is $750dollars and them said that
they will issue you a check of $4950 dollars, so as
soon as you receive the check and cash it you will
take your own money there which is $750 dollars and
send the balanec to me, so that i will use it to pay
for my ticket, i need it urgent .
Becasue the wedding will be comeing up soon.
This is all i needed from you so that my company
will send the check for you immediatley.
Your full name
Your cantcat address
Your country code and name.
Your telephone number.

Please i will like you to send this now so that i
will forward it to my company okay.
Looking forward to hear from you now.
Thanks and god bless you.
Love Sandra

Spam message

- **Music Data**

# bits	SNR	Possible integer values (per sample)	Base-ten signed range (per sample)
4	24.08 dB	16	–8 to +7
8	48.16 dB	256	–128 to +127
11	66.22 dB	2048	–1024 to +1023
12	72.24 dB	4096	–2048 to +2047
16	96.33 dB	65,536	–32,768 to +32,767
20	120.41 dB	1,048,576	–524,288 to +524,287
24	144.49 dB	16,777,216	–8,388,608 to +8,388,607
32	192.66 dB	4,294,967,296	–2,147,483,648 to +2,147,483,647
48	288.99 dB	281,474,976,710,656	–140,737,488,355,328 to +140,737,488,355,327
64	385.32 dB	18,446,744,073,709,551,616	–9,223,372,036,854,775,808 to +9,223,372,036,854,775,807

Principal Component Analysis

- *Dimensionality Reduction by Google*
- https://www.youtube.com/watch?v=wvsE8jm1GzE

- *What if machines can see music?*
- https://www.youtube.com/watch?v=yGl5KFIfSsY

Dimensionality Reduction

1) **Feature elimination**
2) **Feature extraction**

- ***High Dimensions = Lots of Features***

- Dimensionality reduction is *helpful*:
  - Often too many features to do a final classification
  - Higher #features -> more difficult to visualize
  - Higher #features -> more difficult to make classification

UNIVERSITY *of* ROCHESTER

Kaggle Competition and Team Building

Kaggle Competition

- Goal: Classifying ***fake news*** by topic (**multi-class classification** problem)
  - Original dataset contains ~10,000 fake news
  - Plan: You will be provided with ~5,000 observations
  - Expectation: Predict the topics for the rest of the news!
  - You can work in a team of two (2) people
    - And, you are encouraged to work in a team!
  - Dataset contains:
    - ***Date***
    - ***Origin*** (Country)
    - ***Origin*** (Media Source)
    - ***Brief information***
    - ***Long information***
    - ***Topic class*** (~30 different topics)

Kaggle Competition

- ***Expectations***
  - A ***descriptive analysis***
    - <u>One section</u> for undergraduate students
    - <u>Two sections</u> for graduate students
  - A ***prediction challenge***
    - <u>Plan</u>: You will be able to choose any classification model you would like
      - Two separate lists of ranking for undergraduates and graduates
  - A ***report***
    - Summarize your findings and strategies with a final report

Team Building

- Have you met any people from the classroom?
- Would you like to get to know more people?
- Have you checked the class Facebook?

Not this Facebook! ☺

- There is a link on BlackBoard that you can use to identify your team
  - Content Menu -> Kaggle Teams
- Let’s use the remainder of the time for team building.