COSC 220 Guest Lecture

Document-level Language Processing Using Unsupervised Learning

August 6, 2023

Timothy Chen

Overview

- Machine learning
  - Classification vs Regression
  - Supervised vs Unsupervised
- Unsupervised learning
  - K means
  - Expectation Maximization (EM)
  - Hierarchical Clustering
  - Spectral Methods
- Document Level Language Processing
  - SPECTER (Cohan et al. 2020)
  - Methodology
  - Results

Machine Learning

- Data is often represented in tables
- Each column represents one feature
- Each row represents one observation
- Each observation is associated with an output/label Y value
  - Classification (categorical)
  - Regression (continuous)

Input 1	Input 2
2	10
4	21
1	5
8	3

Output
20
85
5
25

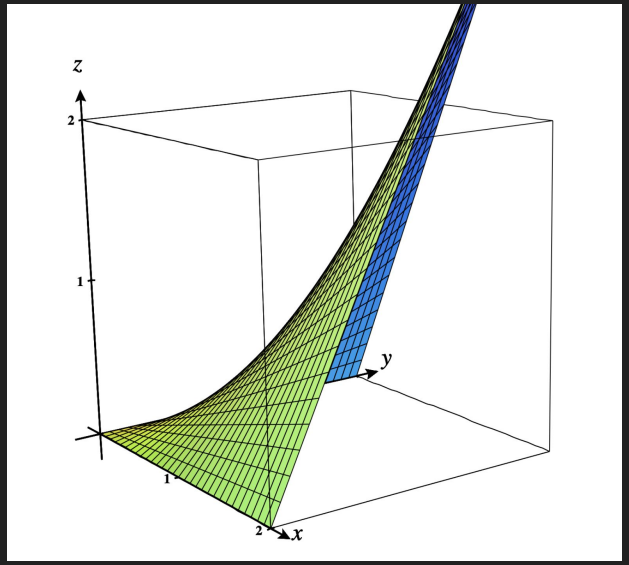
Classification or Regression?

Regression

x	y
2	10
4	21
1	5
8	3

Z
20
85
5
25

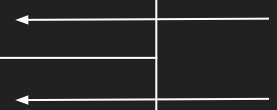
$Z \approx x * y$



Classification or Regression?

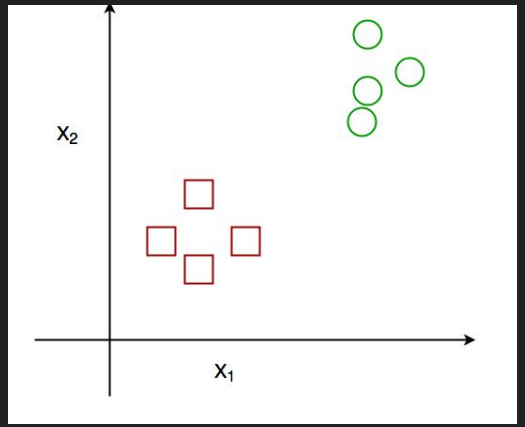
x1	x2
2	10
4	21
1	5
8	3

Y
0
1
1
0



Classification or Regression?

x1	x2	Y
2	10	0
4	21	1
1	5	1
8	3	0

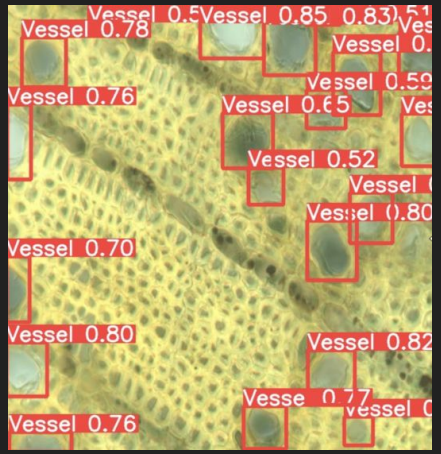


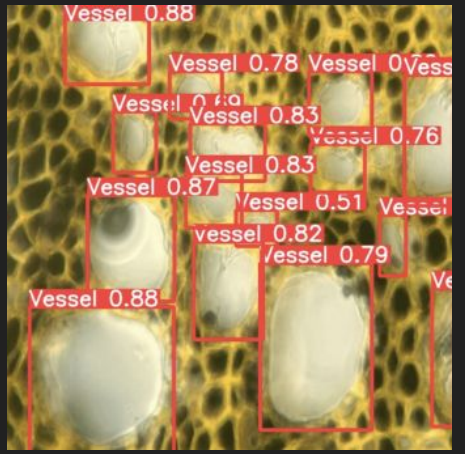
Classification - an example

Automatic Cell Detection for Plant-Based Xylem Tissue Applications

1st Sean Wu
Seaver College
Pepperdine University
Malibu, CA 90265
Sean.wu@pepperdine.edu

2nd Timothy Chen
Seaver College
Pepperdine University
Malibu, CA 90265
Timothy.chen5@pepperdine.edu



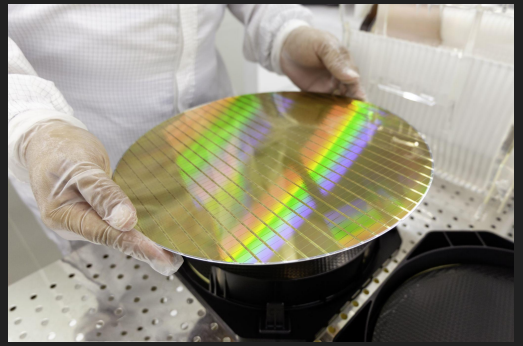


* this is an example object detection, which differs from classification

Another Example - Yield

- Suppose we have a machine that takes part in a certain process for the production of a semiconductor wafer
- Sometimes, the machine produces less defects than others times

- What is the predicted yield given a set of input conditions?



Yield Prediction

Wafer thickness, temperature, queue time, kw/h, etc.

	x1	x2	x3	x4	x5	x6	x7	x8	x9	Output (mm^2 / g)
Run 1	1	5.4	10	12	22	yes	8	8	1	689.654551222
Run 2	2	6.1	18	15	16	no	4	5	1	600.112153266
Run 3	3	3.1	15	1	14	no	7	4	2	590.735276323
Run 4	4	3.4	14	1	5	yes	15	5	1	664.298951367
…										
Run N	10	3.2	41	6	20	yes	7	8	3	???

Supervised learning

- Each sample x is associated with a label Y
- We train the model using previous knowledge of what label Y is based on inputs
- Test the model against test/validation sets

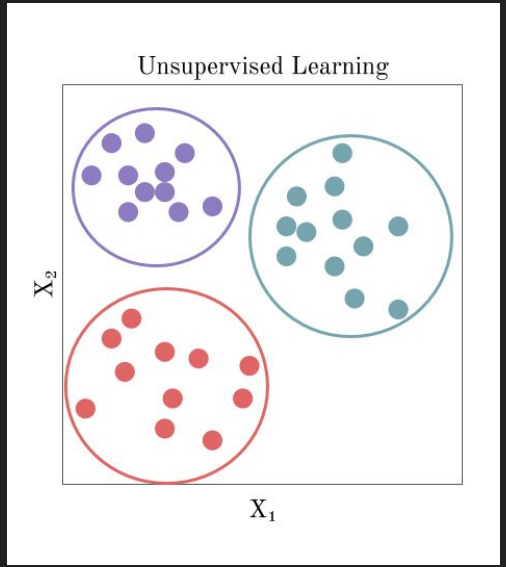
Output (mm^2 / g)
689.654551222
600.112153266
590.735276323
664.298951367

- Use the model to predict unknown Y

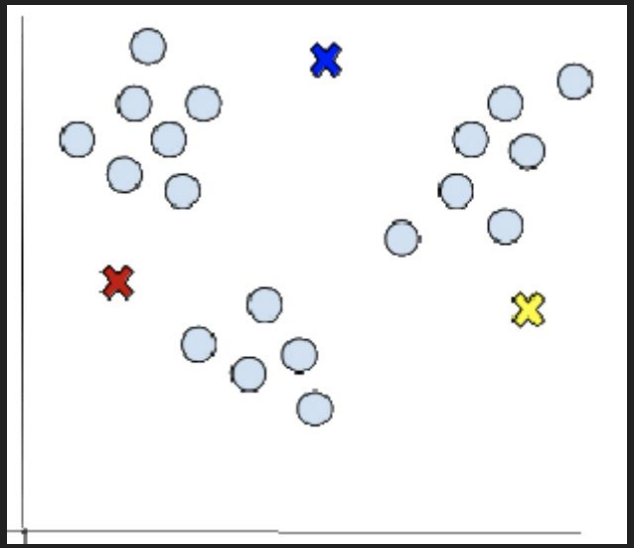
???

Unsupervised Learning

- No labels are provided
- Clustering: observations (data points) are grouped based on input
- Algorithms for unsupervised learning
  - K means
  - Expectation Maximization (EM)
  - Hierarchical Clustering
  - Spectral Methods

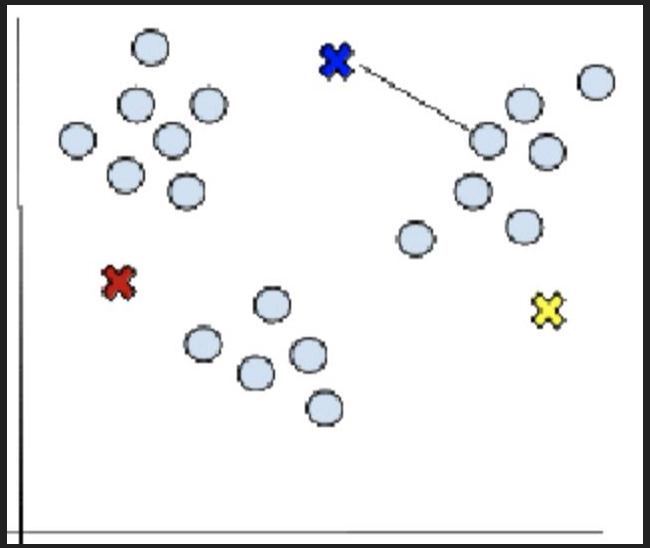


K means algorithm



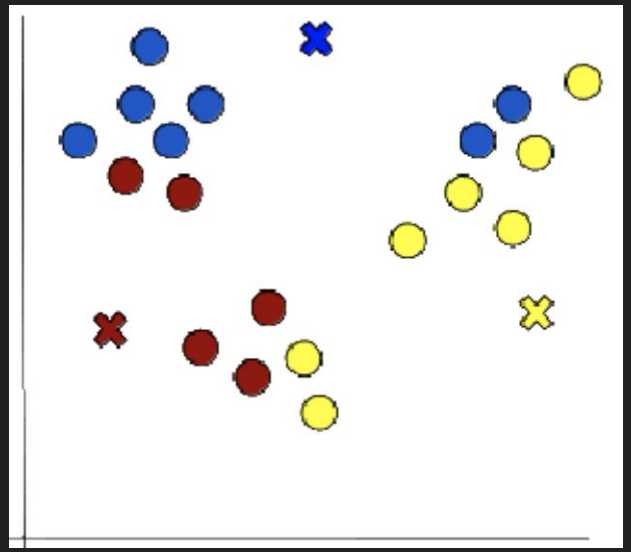
- original data set with uncategorized points
  - Initialize cluster centroids randomly

K means algorithm



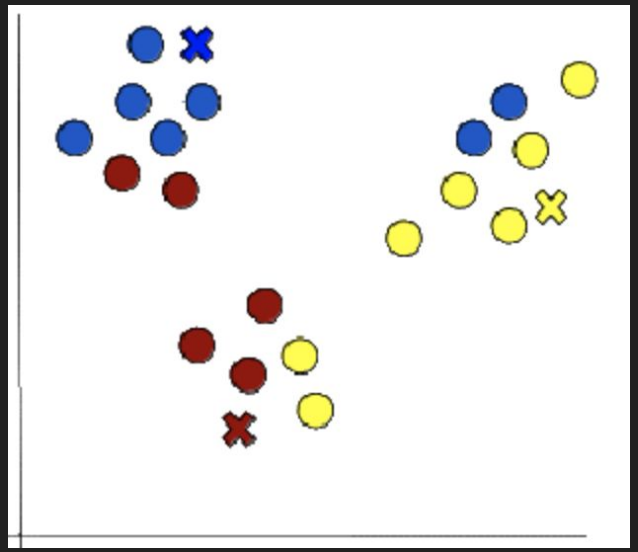
- Assign data points to the closest cluster by distance
- Using Euclidean distance: $d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

K means algorithm



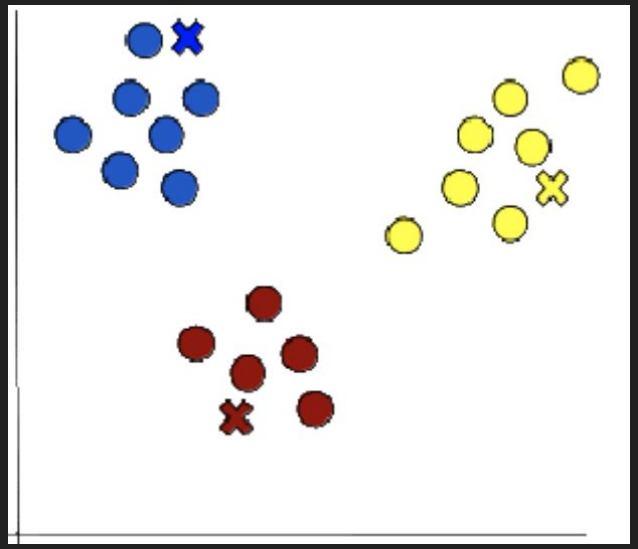
- Assign data points to the closest cluster by distance
- Using Euclidean distance: $d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

K means algorithm



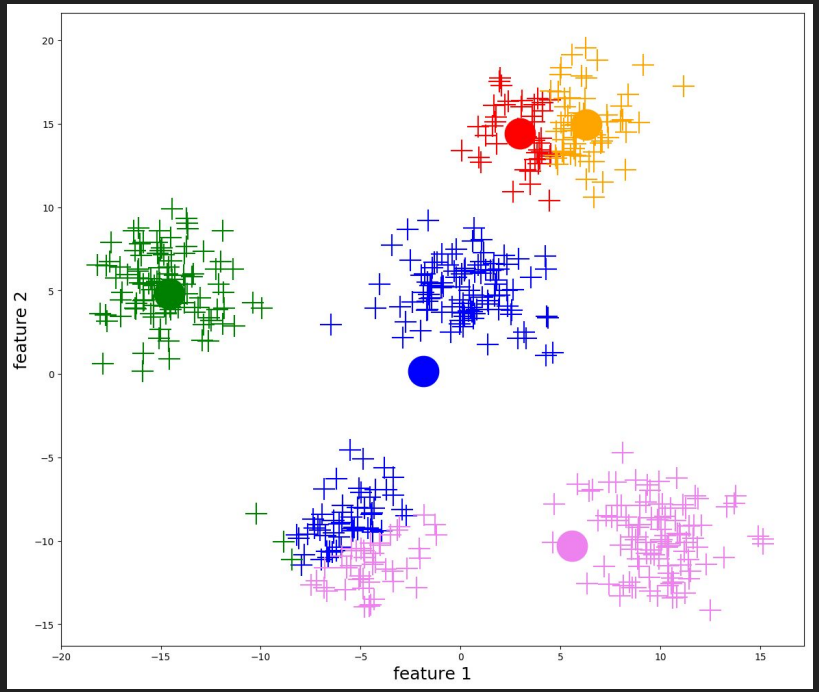
- Reinitialize centroids using the mean of all data points that were assigned in previous step

K means algorithm



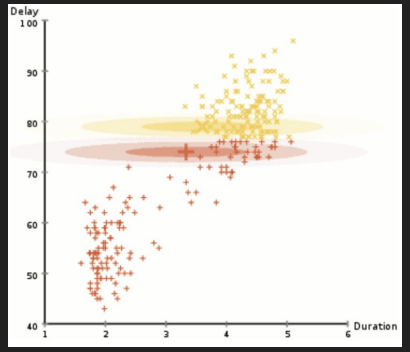
- Repeat previous two steps until convergence

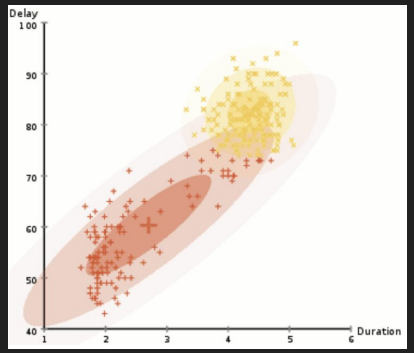
K means algorithm

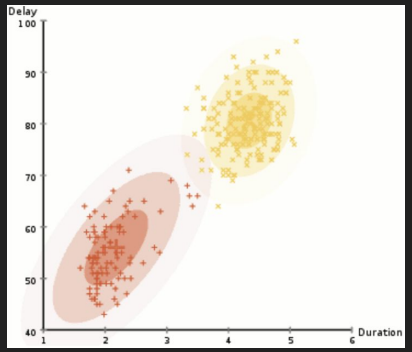


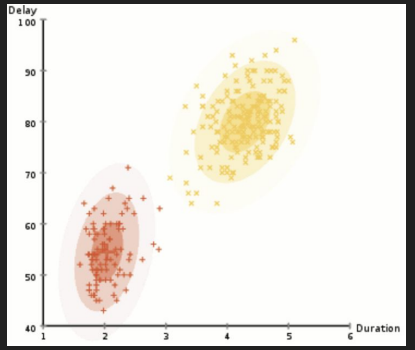
Expectation Maximization (EM)

- Gaussian mixture models
  - Clusters modeled as Gaussians instead of just mean
- Assigns data to cluster *with probability*





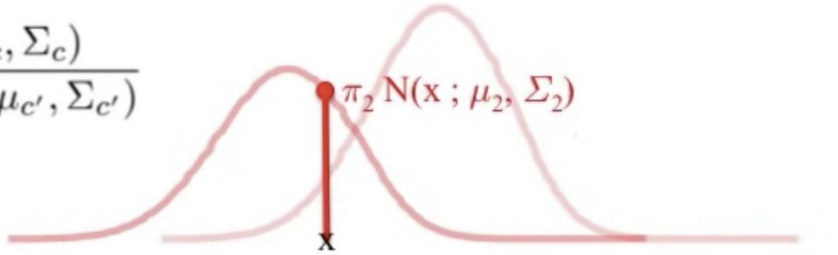




EM Algorithm: E-step

- Start with each cluster represented by mean u_c, covariance Σ_c, and size π_c
- For each data point
  - Calculate the probability that it belongs to cluster using the parameters of each cluster

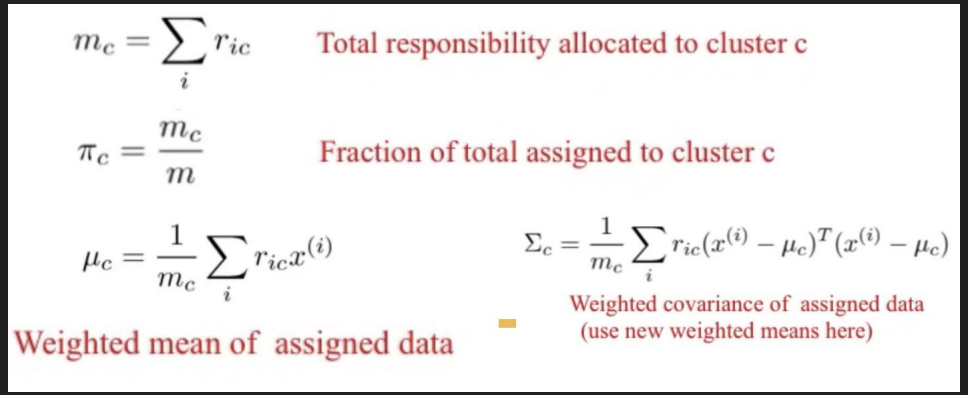
$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

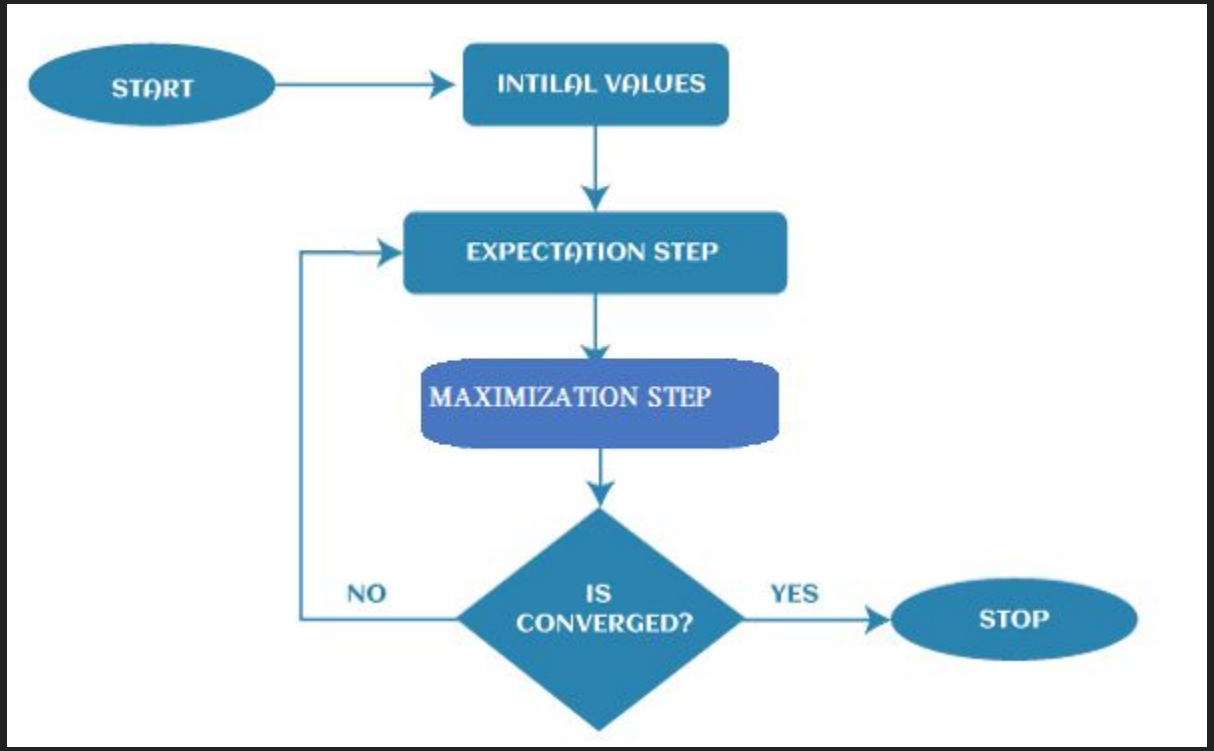


EM Algorithm: M-step

- Using the probabilities calculated in previous step:

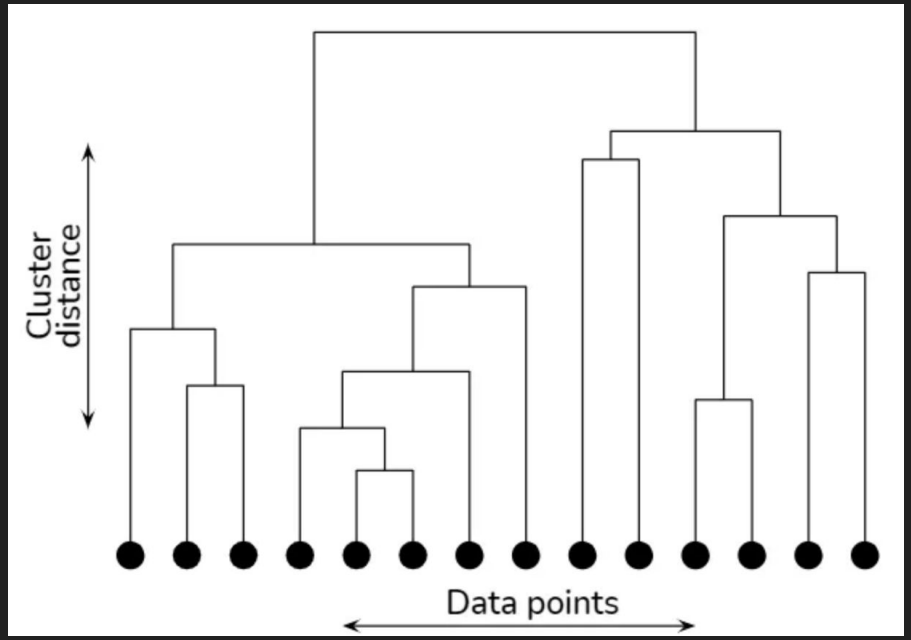
  Update the input parameters (mean, covariance, size) with weighted data points



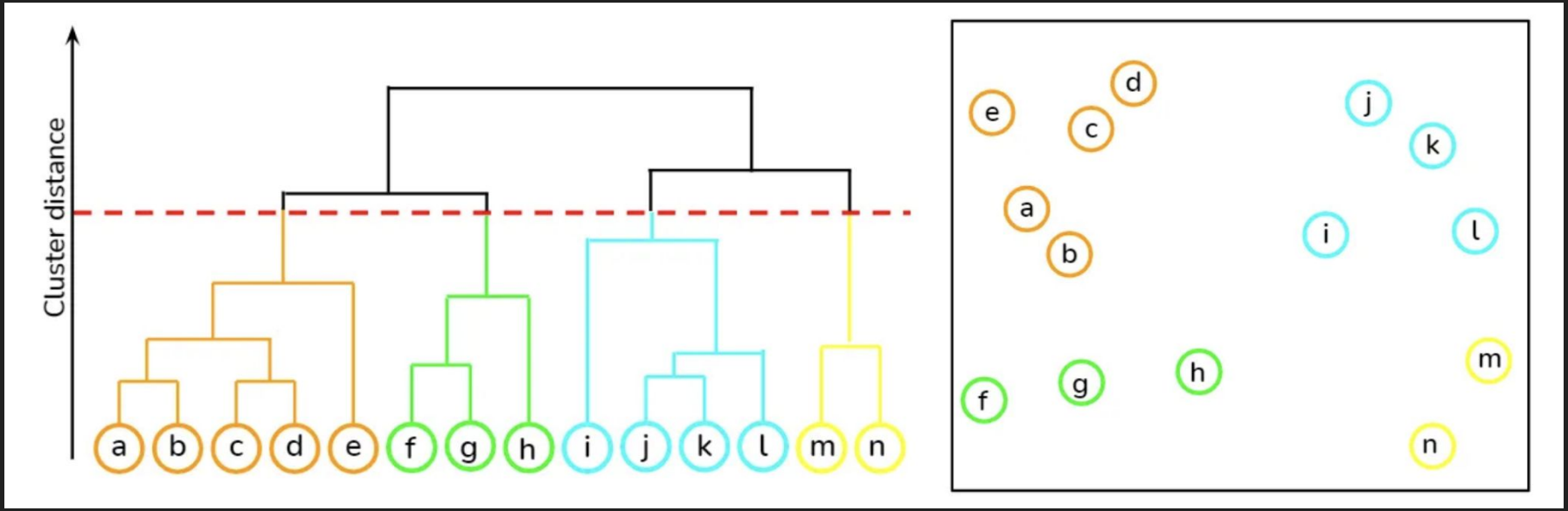

START
INTILAL VALUES
EXPECTATION STEP
MAXIMIZATION STEP
IS CONVERGED?
NO
YES
STOP

Hierarchical clustering

- Utilizes **dendrogram**
- Chooses arbitrary height in the dendrogram as a cutoff to determine clusters
- For every data point, it can tell how close it is to another data point



Hierarchical clustering



Agglomerative vs Divisive

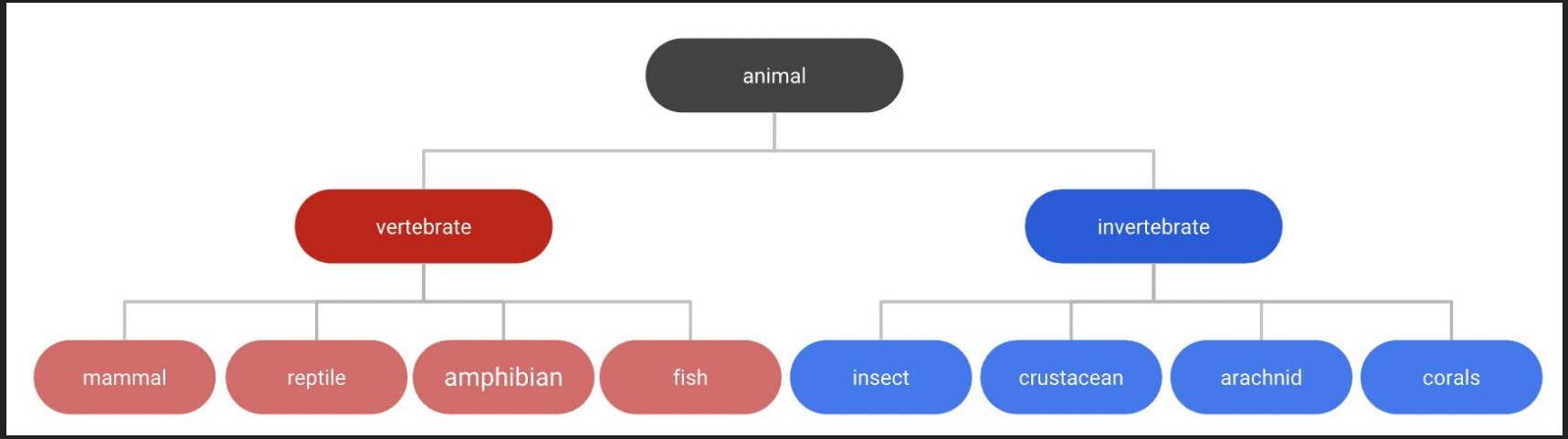
Agglomerative: the more common approach

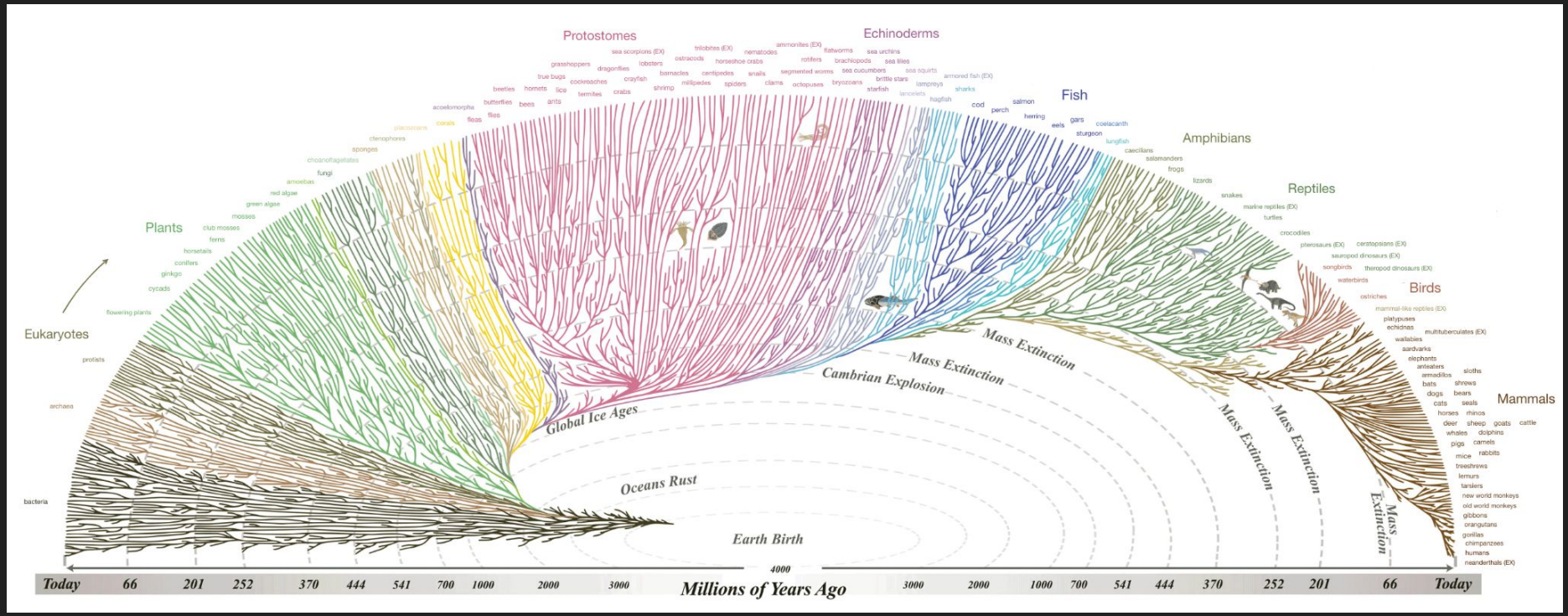
- Bottom up approach where data points are joined iteratively based on similar characteristics until one point is reached

Divisive: less common

- Top down approach where single point is split repetitively into multiple groups

Example of cluster hierarchy




Protostomes
Echinoderms
Fish
Amphibians
Reptiles
Birds
Mammals
Plants
Eukaryotes
bacteria
archaea
protists
fungi
sponges
ctenophores
placozoans
choanoflagellates
amoebas
red algae
green algae
mosses
club mosses
ferns
horsetails
conifers
ginkgo
cycads
flowering plants
sea scorpions (EX)
trilobites (EX)
ammonites (EX)
nematodes
flatworms
sea urchins
ostracods
horsehoe crabs
rotifers
brachiopods
sea lilies
grasshoppers
dragonflies
lobsters
barnacles
centipedes
snails
segmented worms
sea cucumbers
sea squirts
armored fish (EX)
true bugs
cockroaches
crayfish
millipedes
spiders
clams
octopuses
bryozoans
brittle stars
lampreys
sharks
beetles
hornets
lice
termites
crabs
shrimp
starfish
lancelets
hagfish
acoelomorpha
butterflies
bees
ants
cod
perch
salmon
herring
eels
gars
coelacanth
sturgeon
lungfish
corals
fleas
flies
caecilians
salamanders
frogs
lizards
snakes
marine reptiles (EX)
turtles
crocodiles
pterosaurs (EX)
ceratopsians (EX)
sauropod dinosaurs (EX)
songbirds
theropod dinosaurs (EX)
waterbirds
ostriches
mammal-like reptiles (EX)
platypuses
echidnas
multituberculates (EX)
wallabies
aardvarks
elephants
anteaters
armadillos
bats
dogs
cats
horses
deer
whales
pigs
mice
treeshrews
lemurs
tarsiers
new world monkeys
old world monkeys
gibbons
orangutans
gorillas
chimpanzees
humans
neanderthals (EX)
sloths
shrews
bears
seals
rhinos
sheep
goats
cattle
dolphins
camels
rabbits
Mass Extinction
Mass Extinction
Cambrian Explosion
Mass Extinction
Mass Extinction
Mass Extinction
Global Ice Ages
Oceans Rust
Earth Birth
4000
Today
66
201
252
370
444
541
700
1000
2000
3000
Millions of Years Ago
3000
2000
1000
700
541
444
370
252
201
66
Today

Spectral Clustering

Unnormalized spectral clustering

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

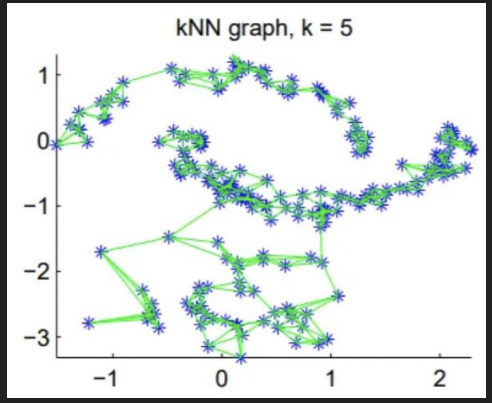
- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $u_1, \ldots, u_k$ of L.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

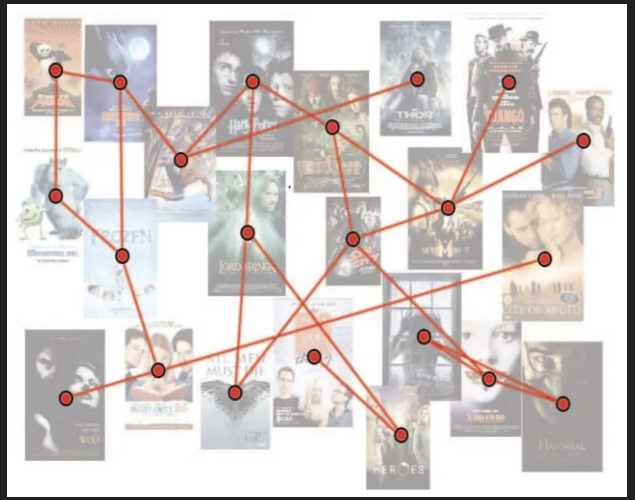
Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | \, y_j \in C_i\}$.

Spectral Methods - Similarity Matrix

- Represent similarity between points in a graph
- Each node represents a point and each edge represents a weighted similarity
- Use the graph to create similarity matrix A where Aij represents the weight of the edge between points i and j

$$w_{ij} = \exp\left\{-\frac{\|x_i - x_j\|}{\sigma^2}\right\}$$





Spectral Methods - Laplacian

- Calculate Laplacian matrix
- L = D - W
- W is the similarity matrix
- D is diagonal matrix whose entries are column sums of W
- D serves as a normalization factor to balance cluster affinities: makes sure a single cluster does not dominate or skew the results

Labeled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$

Spectral Methods - Eigenvectors

- Final step is to calculate the eigenvectors of the Laplacian. First K eigenvectors (associated with smallest eigenvalues) -> columns of matrix
- K = dimensionality: chosen using spectral gap or elbow method
- Take the rows of resulting matrix and run K means

- Result: reduces dimensionality
  - Eigenvectors associated with smaller eigenvalues of the Laplacian matrix capture essential, structural information in the data

$$\mathbf{A} \cdot \mathbf{v}_1 = \lambda_1 \cdot \mathbf{v}_1$$

$$(\mathbf{A} - \lambda_1) \cdot \mathbf{v}_1 = 0$$

$$\begin{bmatrix} -\lambda_1 & 1 \\ -2 & -3-\lambda_1 \end{bmatrix} \cdot \mathbf{v}_1 = 0$$

$$\begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix} \cdot \mathbf{v}_1 = \begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix} \cdot \begin{bmatrix} v_{1,1} \\ v_{1,2} \end{bmatrix} = 0$$

Why Spectral Clustering?

- Focuses more on structure and similarity of data
- Can adapt to unique shapes of data and outliers, since it accounts for structure
- Dimensionality reduction
- No initial cluster centers unlike kmeans, making it less sensitive to initial input parameters

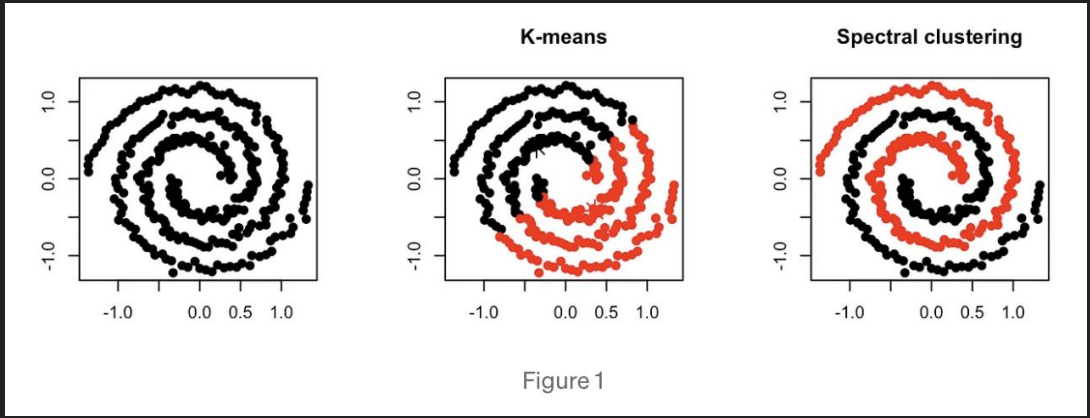


Figure 1

Publication Sorting Using Document-level Language Processing

Problem: Updates in Pepperdine University research and new publications are not often known across different departments.

Proposal: Pepperdine research social network!

This network must have some document searching functionality

Goal: A novel approach to automated, dynamic document sorting based on publication *content*, not just title and keywords, using clustering

SPECTER: Document-level Representation Learning using Citation-informed Transformers

Arman Cohan[†*] **Sergey Feldman**[†*] **Iz Beltagy**[†] **Doug Downey**[†] **Daniel S. Weld**[†,‡]

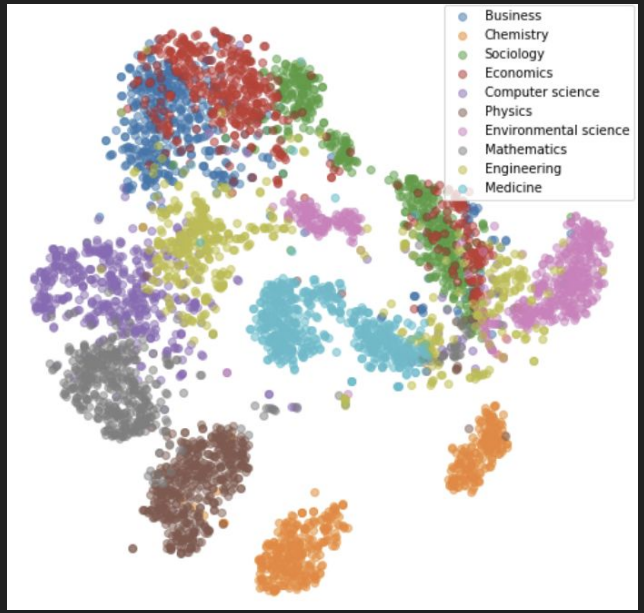
[†]Allen Institute for Artificial Intelligence

[‡]Paul G. Allen School of Computer Science & Engineering, University of Washington

{armanc,sergey,beltagy,dougd,danw}@allenai.org

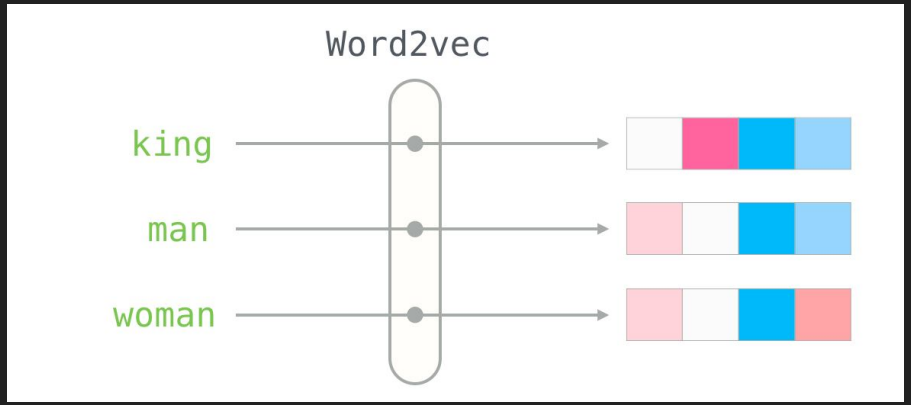
Summary:

- Analyzes citations within a publication and the *impact* that the citation has on the paper
- Groups documents by interdocument relatedness
- Stores data as a SPECTER filed, a vector of floats that represents content



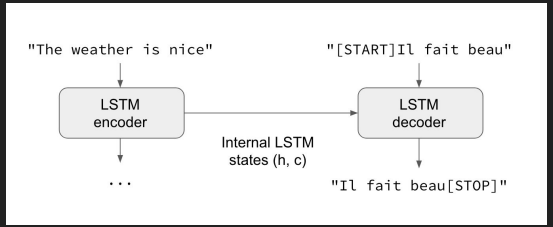
Specter - Representation Learning

- Goal is to extract *features*, important characteristics of the raw data, that represent the data
- Transforms raw data into a more compact and expressive format that captures the underlying structure
- Learn based on these features



Specter - Transformer model architecture

- Transformer:
  - Self attention: weigh impact of different parts of input based on relationship between words
  - Encoder decoder
  - "Attention is All You Need" Vaswani et al. in 2017
- In SPECTER
  - Transformer encodes research paper
  - Pretrained on citation data to learn document relatedness
  - Citation links affect similarity of papers, as shown in output of Transformer



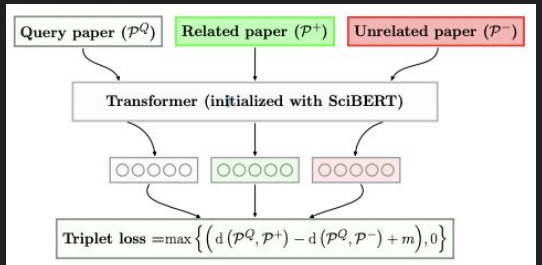


Figure 1: Overview of SPECTER.

How are SPECTER embeddings generated?

- Embedding = [2,1,4,5……]
- Each element represents certain features
  - Semantic content, contextual information, citation relatedness, structure information, etc.

- Designs loss function to learn to embed these papers closer together when one cites another, and further when they don't
  - Pq = paper, P+ = cited paper, P- = not cited paper, d = distance function, m is loss margin hyperparameter

- Pretrains Transformer using this document relatedness, as represented by the function

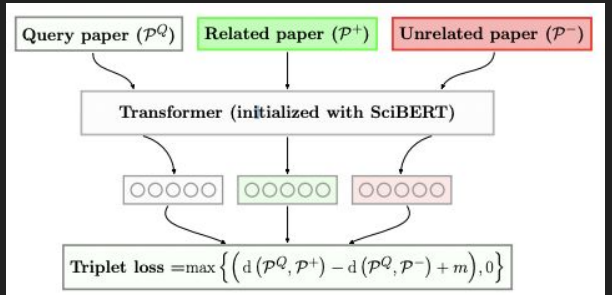


Figure 1: Overview of SPECTER.

$$\mathcal{L} = \max\left\{\left(\mathrm{d}\left(\mathcal{P}^Q, \mathcal{P}^+\right) - \mathrm{d}\left(\mathcal{P}^Q, \mathcal{P}^-\right) + m\right), 0\right\}$$

Task →	Classification		User activity prediction				Citation prediction			
Subtask →	MAG	MeSH	Co-View		Co-Read		Cite		Co-Cite	
Model ↓ / Metric →	F1	F1	MAP	nDCG	MAP	nDCG	MAP	nDCG	MAP	nDCG
Random	4.8	9.4	25.2	51.6	25.6	51.9	25.1	51.5	24.9	51.4
Doc2vec (2014)	66.2	69.2	67.8	82.9	64.9	81.6	65.3	82.2	67.1	83.4
Fasttext-sum (2017)	78.1	84.1	76.5	87.9	75.3	87.4	74.6	88.1	77.8	89.6
SIF (2017)	78.4	81.4	79.4	89.4	78.2	88.9	79.4	90.5	80.8	90.9
ELMo (2018)	77.0	75.7	70.3	84.3	67.4	82.6	65.8	82.6	68.5	83.8
Citeomatic (2018)	67.1	75.7	81.1	90.2	80.5	90.2	86.3	94.1	84.4	92.8
SGC (2019a)	76.8	82.7	77.2	88.0	75.7	87.5	**91.6**	**96.2**	84.1	92.5
SciBERT (2019)	79.7	80.7	50.7	73.1	47.7	71.1	48.3	71.7	49.7	72.6
Sent-BERT (2019)	80.5	69.1	68.2	83.3	64.8	81.3	63.5	81.6	66.4	82.8
SPECTER (Ours)	**82.0**	**86.4**	**83.6**	**91.5**	**84.5**	**92.4**	88.3	94.9	**88.1**	**94.8**

Application of SPECTER

SPECTER Field

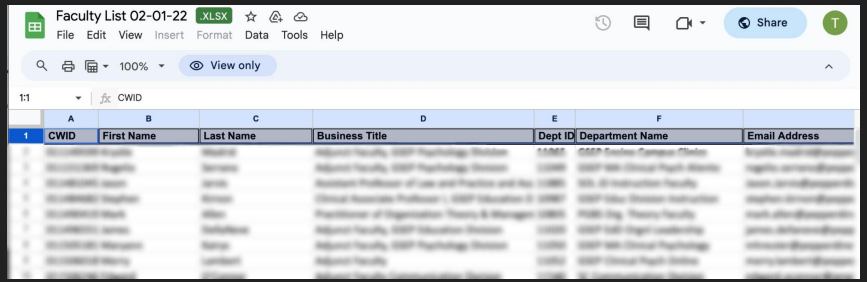
{"model": "specter@v0.1.1", "vector": [-8.82082748413086, -2.6610865592956543, 0.8936576843261719, 0.26382559537887573, 1.513496994972229, -1.0313653945922852, 0.7256110906600952, 2.080476999282837, 0.8230894804000854, 4.205087661743164, -0.9822302460670471, -0.03232526779174805, -2.4075639247894287, -0.9867752194404602, -1.921100378036499, 0.7259652018547058, -0.7707503437995911, -3.645012140274048, 3.874178647994995, 1.7716397047042847, 3.568206787109375, 0.44507932662963867, -4.567941188812256, -6.441675662994385, 2.8164267539978027, -5.735928535461426, 4.397401332855225, 1.8640488386154175, -2.0923988819122314, -2.627096176147461, -0.7487516403198242, -2.0460338592529297, 4.951084613800049, 0.1367805004119873, 0.11137333512306213, -4.1136651039123535, -2.594398021697998, 3.030726671218872, 0.3192863166332245, 0.5330687761306763, 2.1357359886169434, -5.069549560546875, 3.3368606567382812, -0.8448878526687622, 0.03290879726409912, 3.0217368602752686, -2.1936895847320557, -2.591491937637329, -2.7961978912353516, 2.9690139293670654, 1.8845709562301636, 2.0488815307617188, -0.3929055333137512, 3.2013416290283203, -0.02199321985244751, 0.6550448536872864, 4.03271033549308777, 0.9349490404129028, 2.9326210021972656, -0.9087234735488892, 1.6680347919464111, -1.7342610359191895, 5.191349506378174, 0.8320249915122986, 0.632623553276062, -4.146793842315674, -1.180707573890686, 3.647608757019043, 2.951371669769287, 3.1728854179382324, 1.2579790353775024, -2.1136271953582764, -0.9828354716300964, -3.1169707775115967, 2.203051805496216]}}

Clustering
(e.g. K-means)

Publications Sorted by Content

Data collection

- Obtained directory of all active Pepperdine faculty
- Queried each faculty's author id using the Semantic Scholar API
- Used each author id to find all publications
- Queried the SPECTER fields of each publication



Data collection

```
[ ] #Get all SPECTER embeddings
    spct = []

    #loops through each author
    for _paperIds in df["Paper Ids"]:
      curr_spct = []

      #loops through each paper of current author and appends that paper's spct embedding to curr_spct
      for id in _paperIds:
        response = requests.get(f'https://api.semanticscholar.org/graph/v1/paper/{id}?fields=embedding').text
        x = json.loads(response)
        if x["embedding"] != None:
          specter = x['embedding']['vector']
        else:
          specter = []
        curr_spct.append(specter)

      #appends all specter vectors for all publications of current author to spct
      spct.append(curr_spct)
```

```
[ ] df["Specter Embedding"] = spct
```

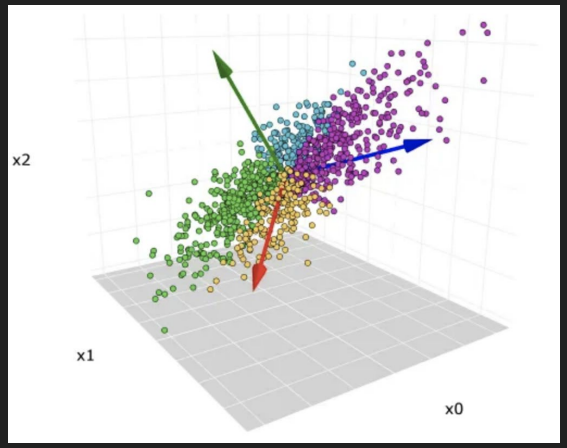
```
print(df)
```

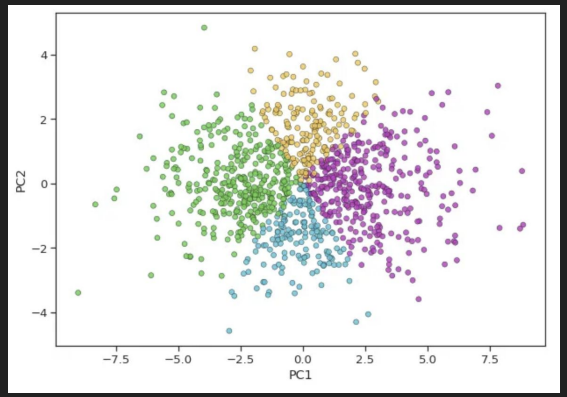
```
             0           1   Author ID  \
0      Krystle      Madrid   118985833
1      Rogelio     Serrano  2057791680
2        Jason      Jarvis   121754802
3      Stephen      Kirnon   107707217
4         Mark       Allen    48344112
...        ...         ...         ...
1014     Parsa      Peykar          -1
1015  Michelle     Leccese          -1
1016  Samantha       Hamed          -1
1017       Tad  Machrowicz  2047754792
1018   Olaseni      Hazzan          -1

                                                Paper Ids  \
0            [0b698cd2269d9b52c75dbdf1994dbc9a98fb16c8]
1     [459751e8ecc4e33070679fa866136d84ffadf19d, c01...
2     [2cd1ef8a86f772cbd177f8a10ad90b64a84ec0fd, c88...
3     [81d64b6c5d39011614d3115a0acde2157fffdd78, bca...
4     [ef79750b5adcdc7f82be728c5bee99af7ecf99fc, ac1...
...                                                 ...
1014                                                 []
1015                                                 []
1016                                                 []
1017  [1e3596b19f8ec88c747320f0f81a80c421d5acdd, e98...
1018                                                 []

                                        Specter Embedding
0     [[-1.9917640686035156, -2.0737364292144775, -2...
1     [[-2.988846778869629, -3.4957807064056396, 1.2...
2     [[0.41440069675445557, -4.595344066619873, 1.8...
3     [[-1.057356834411621, -2.260591745376587, 0.95...
4     [[-3.595524787902832, -1.4567368030548096, -0....
...                                                 ...
1014                                                 []
1015                                                 []
1016                                                 []
1017  [[2.803157091140747, -3.5736403465270996, 3.76...
1018                                                 []

[1019 rows x 5 columns]
```

Dimensionality Reduction

Principal Component Analysis (PCA):

- Reduces dimension of data for visualization purposes while retaining most of the information
- Chooses important features with high importance
- Feature with greatest variance becomes the 1st principal component





Model Performance

Silhouette Scores:

- Measures how well the model performs, specifically, how close an object is to its own cluster compared to other clusters
- 1: clusters are clearly distinguished
- 0: insignificant distance between the clusters
- -1: sample assigned to wrong cluster

This can also help choose how many clusters to use!

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i,j)$$

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_I| > 1$$

```
from sklearn.metrics import silhouette_score
labels = clustering.labels_
silhouette_score(df1, labels, metric = 'euclidean')
```

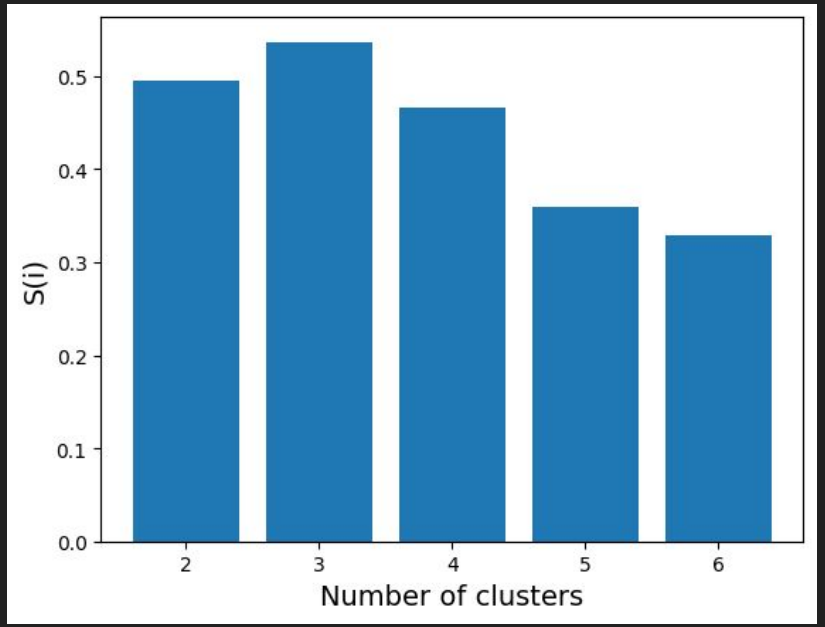
Choosing number of clusters

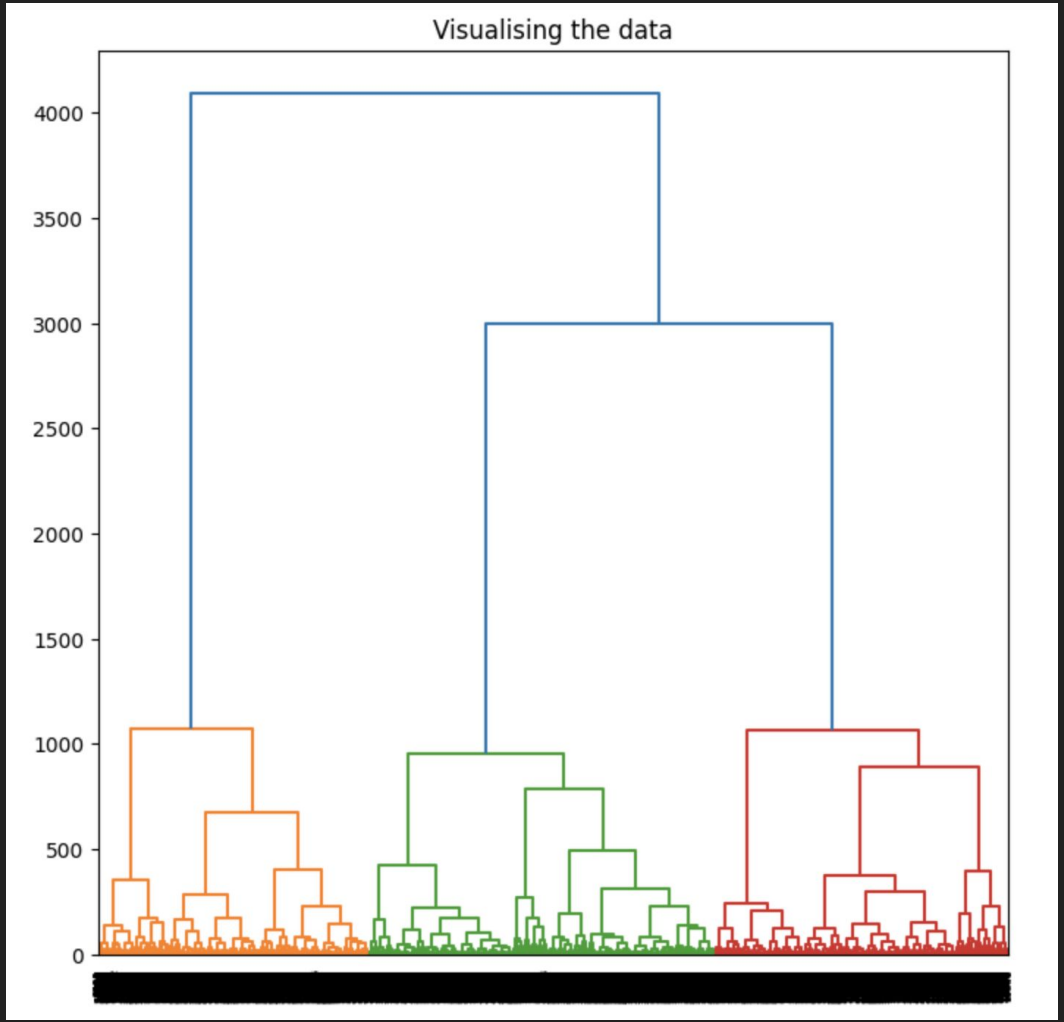
Use silhouette score

The higher score, the stronger the model

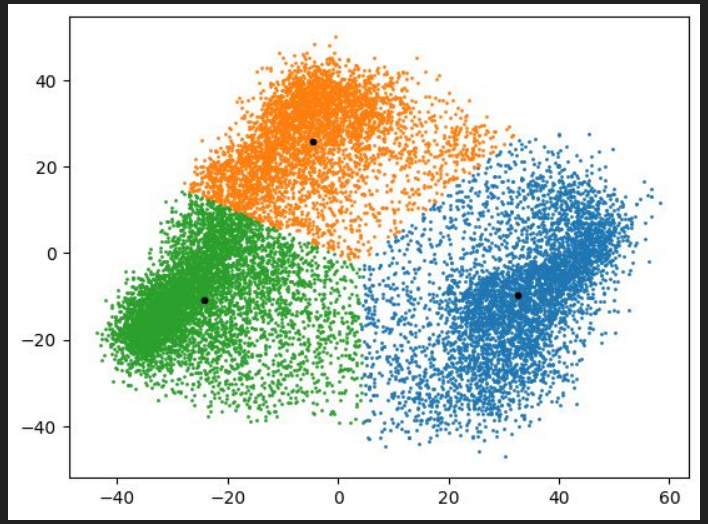
The chart to the right displays the silhouette scores for our hierarchical clustering model for each number of clusters from 2-6

N=3 clusters seems to be the best fit



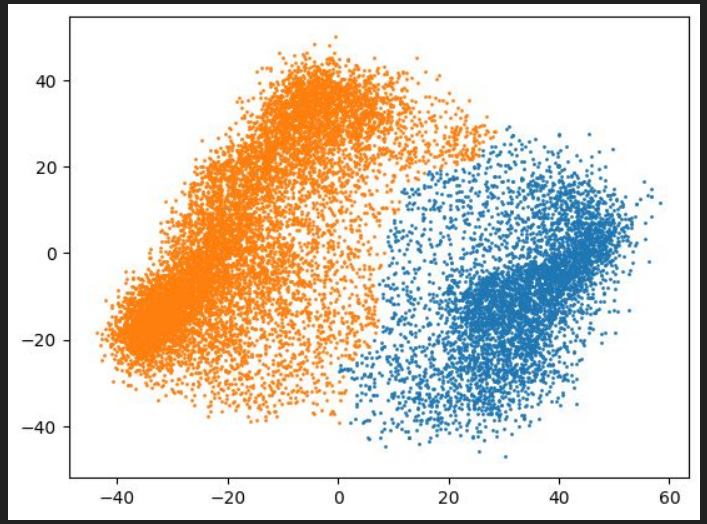

Visualising the data
4000
3500
3000
2500
2000
1500
1000
500
0

Results



K-means

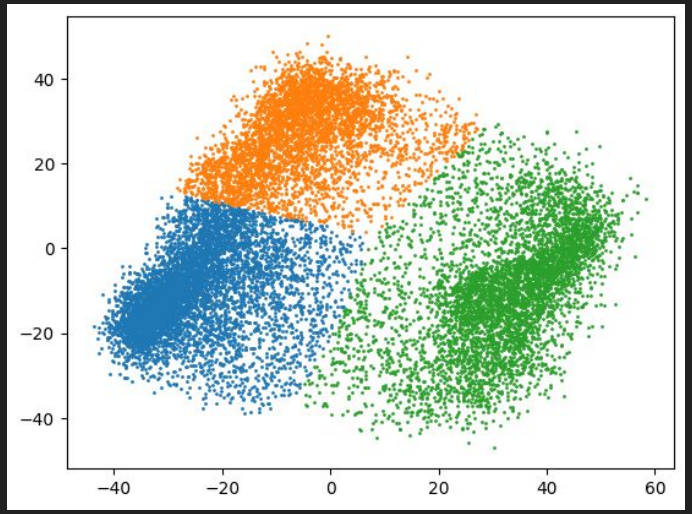
Silhouette score (n=3): 0.56529



Spectral Clustering

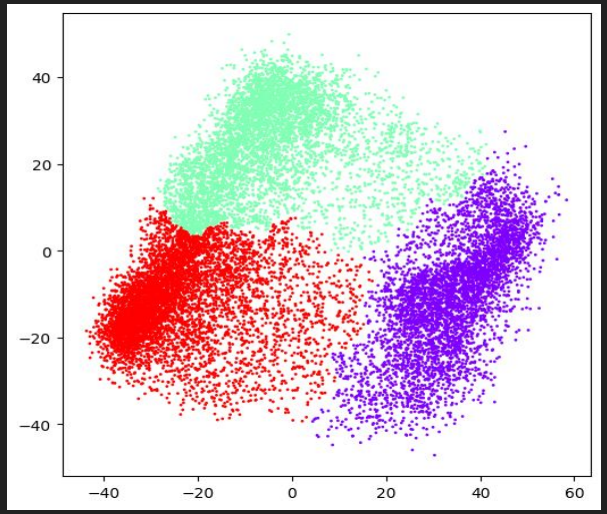
Silhouette score (n=2): 0.51286

Results (cont)



Gaussian Mixture Model

Silhouette score (n=3): 0.45129



Hierarchical Clustering

Silhouette score (n=3): 0.55215

Future Improvements: Dimensionality Reduction

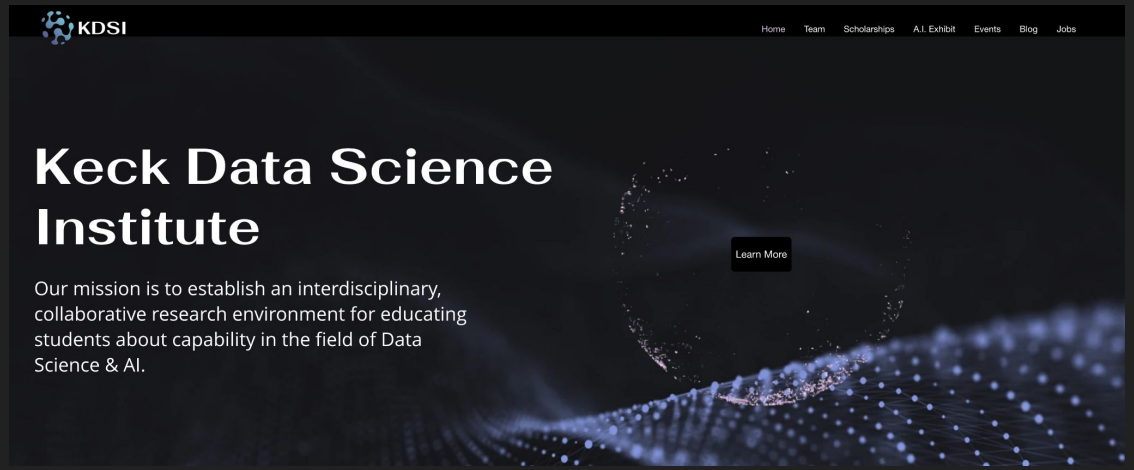
T-distributed Stochastic Neighbor Embedding (t-SNE):

- Better suited for nonlinear relationships
- Preserves the similarities between data points, pairwise distances
- Calculates pairwise similarity in high dimension with Gaussian kernel, then groups these points in lower dimension based on the similarities

```
from sklearn.manifold import TSNE

tsne = TSNE(n_components=2, random_state=42)
X_tsne = tsne.fit_transform(X)
tsne.kl_divergence_
```

Conclusion

- Showcases capability to group documents using semantic data
- Can be implemented in social networking site to automatically sort publications
- Can automatically group new publications
- Allows users to look for similar publications using cluster data



References

[1] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2270–2282, Online. Association for Computational Linguistics.

[2] Chowdhary, KR1442, and K. R. Chowdhary. ”Natural language processing.” Fundamentals of artificial intelligence (2020): 603-649.

[3] Hirschberg, Julia, and Christopher D. Manning. ”Advances in natural language processing.” Science 349.6245 (2015): 261-266.

[4] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, System Demonstrations (Association for Computational Linguistics, Stroudsburg, PA, 2014), pp. 55–60.

[5] P. Koehn, F. J. Och, D. Marcu, “Statistical phrase-based translation,” in Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (Association for Computational Linguistics, Stroudsburg, PA, 2003), pp. 48–54.

[6] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. ”Sequence to sequence learning with neural networks.” Advances in neural information processing systems 27 (2014). [7] Shahapure, Ketan Rajshekhar, and Charles Nicholas. ”Cluster quality analysis using silhouette score.” 2020 IEEE 7th international conference on data science and advanced analytics (DSAA). IEEE, 2020. [8] Shlens, Jonathon. ”A tutorial on principal component analysis.” arXiv preprint arXiv:1404.1100 (2014).