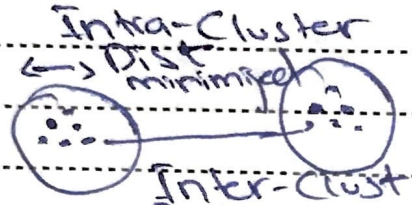
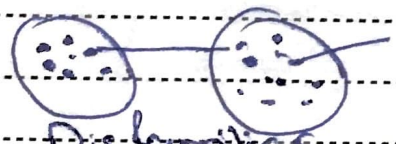
DATE & DAY

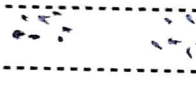
MON TUE WED THU FRI SAT SUN

Cluster: Collection of data Objects

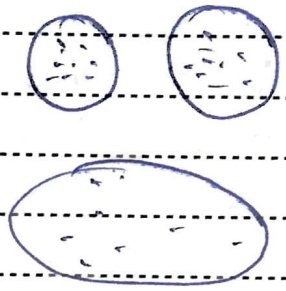




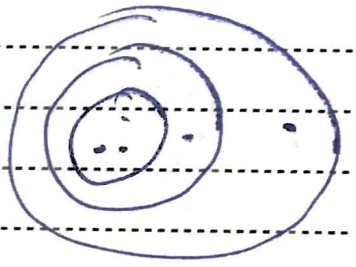
Original Points



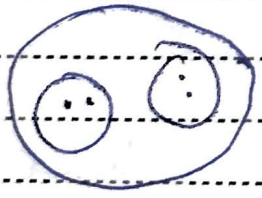
Partional Clustering



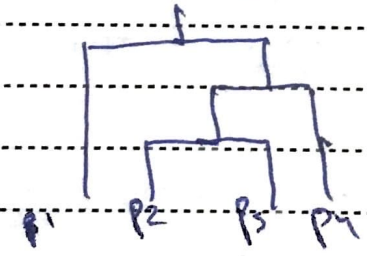
Traditional Hierarchical



Non-Traditional Hierarchical



Traditional Dendogram



Partioning Method: Partioning D database of n objects into a set of k clusters, such that the sum of sq dist is minimized where c_i is the centroid or medroid.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} (p - c)^2$$

stewart™

DATE DAY

MON TUE WED THU FRI SAT SUN

Heuristic Method: K-means, K-mediods

K-Means Clustering

$O(n*K*I*d)$ n = no. of points, K = no. of clusters, I = no. of iterations, d = no. of attributes

	Data Points	D: A1	D: B1	D: C1	C
1 A1	(2, 10)	0	3.61	8.06	1
	2, 5	5	4.24	3.16	3
2 B1	8, 4	8.5	5	7.28	2
	5, 8	3.61	0	7.21	2
	7, 5	7.07	3.61	6.71	2
3 C1	6, 4	7.21	4.12	5.39	2
	1, 2	8.06	7.21	0	3
	4, 9	2.24	1.41	7.62	2

Step 1 : Select Random Centroids

Step 2 Find Dist of each point for every cluster

Step 3 Assign the point to the closest cluster

Step 4 Update the centroids

Step 5 Do until converge

Euclidean Dist = $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

New Centroids

A1 = (2, 10)

$B1 = \left(\frac{8+5+7+6+4}{5}, \frac{4+8+5+4+9}{5}\right) = (5, 8)$

$C1 = \left(2+\frac{1}{3}, 5+\frac{2}{3}\right) = (1, 2)$

Data Points		D:A1	D:B2	D:C1	C
2	10	0	5.66	8.52	1
2	5	5	4.12	1.58	3
8	4	8.49	2.83	6.52	2
5	8	3.61	2.24	5.70	2
7	5	7.07	1.41	5.70	2
6	4	2.21	2	4.53	2
1	2	8.06	6.4	1.58	3
4	9	2.24	3.61	6.04	1

Do Untill all Clusters Converge

H- Node Clustering

For categorical data

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

we find the ~~dissss~~ dissimilarities b/w the point and the cluster

	Person	hair color	eye color	skin color	P:A1	O:B1	O:C1
A1	P1	blonde	amber	fair	0	2	2
	P2	brunette	gray	brown	3	3	3
	P3	red	green	brown	3	1	3
	P4	black	hazel	brown	3	3	1
	P5	brunette	amber	fair	1	2	2
2	P6	black	gray	brown	3	3	2
B1	P7	red	green	fair	2	0	2
C1 3	P8	black	hazel	fair	2	2	0

New Centroids

				1
A1	Blonde	amber	fair	1
	brunette	gray	brown	2
	brunette	amber	fair	3
				1
Since brunette occurs 2 time we will				3
select brunette. Same with amber				2
and fair. If all were diff then we				3
would've selected randomly				

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

A1 : brunette amber fair

Similarly do with other points

B1 : red green fair

C1 : black hazel brown

Repeat the process untill Converge

K-Mediod

	X	Y	D:C1	D:C2	Cluster
	8	7	6	2	2
	3	7	3	7	1
	4	9	4	8	1
	9	6	6	2	2
C1	8	5	-	-	
	5	8	4	6	1
	7	3	5	3	2
	8	4	5	1	2
	7	5	3	1	2
C2	4	5	-	-	

Dist : $|X_1 - X_2| + |Y_1 - Y_2|$

DATE & DAY

MON TUE WED THU FRI SAT SUN

Cost = (3+4+4) + (3+1+1+2+2) = 20

	X	Y	D:C1	D:C2	Cluster
	8	7	6	3	2
	3	7	3	8	1
	4	9	4	9	1
	9	6	6	3	2
	8	5	4	1	2
	5	8	4	7	1
	7	3	5	2	2
C1	8	4	-	-	
	7	5	3	2	2
C2	4	5	-	-	

New

Cost = (3+4+4) + (2+2+1+3+3) = 22

Swap cost = New cost - Old cost

= 22 - 20 = 2 > 0

As the swap cost is not less than zero, we undo the swap

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

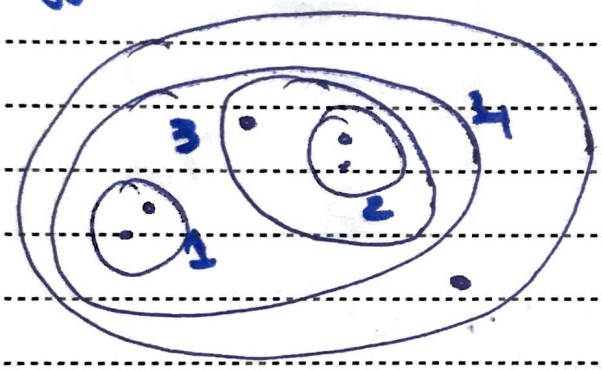
Hierarchical Clustering

uses dist matrix as a clustering criteria

Hierarchical Clustering
- Agglomerative
- Divisive

Agglomerative Clustering

bottom-up Approach



Consider the following points

18, 22, 25, 42, 27, 43

Step: 1

using Single Linkage

	18	22	25	27	42	43
18	0					
22	4	0				
25	7	3	0			
27	9	5	2	0		
42	24	20	17	15	0	
43	25	21	18	16	1	0

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

	18	22	25	27	42,43
18	0				
22	4	0			
25	7	3	0		
27	9	5	2	0	
42,43	24	20	17	15	0

	18	22	25,27	42,43
18	0			
22	4	0		
25,27	7	3	0	
42,43	24	20	15	0

	18	25,27,22	42,43
18	0		
25,27,22	9	0	
42,43	24	15	0

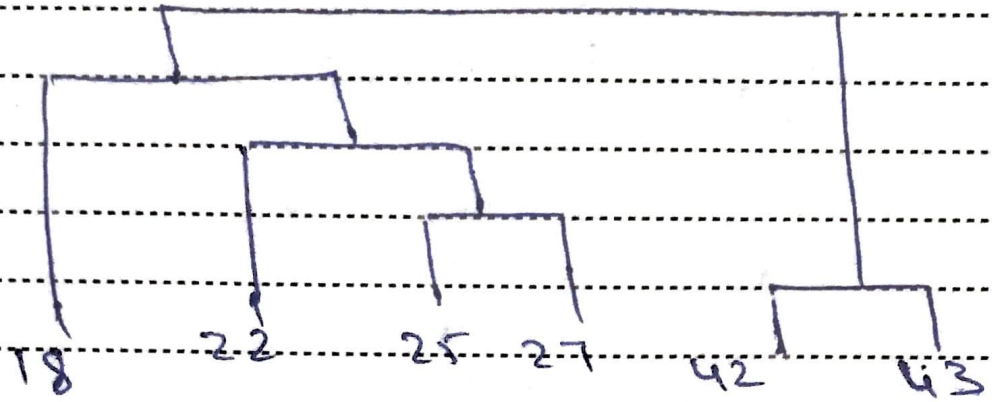
	18,25,27,22	42,43
18,25,27,22	0	
42,43	15	0

18, 25,27,22 ,42,43

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN



Single Linkage — Min

Complete Linkage — Max

Average Linkage — Avg of Dist

Divisive Clustering — Top down Approach

P	x	y
1	0.4	0.53
2	0.22	0.38
3	0.35	0.32
4	0.26	0.19
5	0.08	0.41
6	0.45	0.3

Step: 1

dist Matrix

	1	2	3	4	5	6
1	0					
2	0.23	0				
3	0.22	0.15	0			
4	0.37	0.20	0.15	0		
5	0.34	0.14	0.28	0.29	0	
6	0.23	0.25	0.11	0.22		0

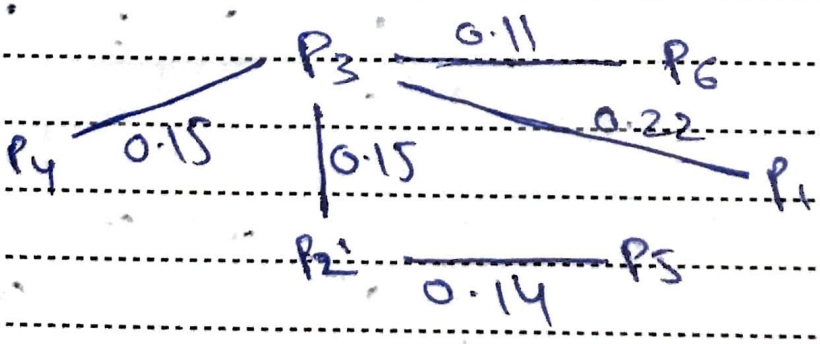
DATE & DAY

MON TUE WED THU FRI SAT SUN

Step:2 Compute Minimum Spanning Tree

Edge	Cost	← Prim's Algo
P3, P6	0.11	
P2, P5	0.14	
P2, P3	0.15	
P3, P4	0.15	
P2, P4	0.20	
P1, P3	0.22	
P1, P2	0.23	
P1, P6	0.23	
P2, P6	0.25	
P3, P5	0.28	
P4, P5	0.29	
P1, P5	0.34	
P1, P4	0.37	
P5, P6	0.39	

Step:3



stewart™

DATE

MON TUE WED THU FRI SAT SUN

Applying Complete Linkage to break the edges

0.11

0.15 P5 P6

P4 0.15 P1

P2 0.14 P3

P5 0.11 P6

P4 0.15 P1

P2 0.14 P3

P5 0.11 P6

P4 P1

P2 0.14 P5

P3 0.11 P6

P4 P1

P2 P5

P3 P6

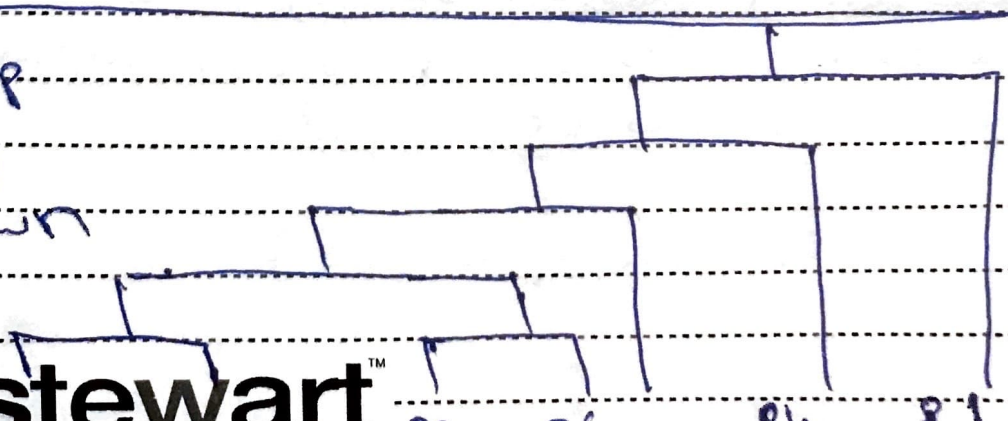
P4 P1

P2 P5

Top

⇓

Down



stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

BIRCH

Branching factor B=2

max no. of sub-cluster at each leaf node L=5

threshold T=1.5

Data

x	y	x^2	y^2
3	4	9	16
2	6	4	36
4	5	16	25
4	7	16	49
3	8	9	64
6	2	36	4
7	2	49	4
7	4	49	16
8	4	64	16
8	5	64	25

→ Uses CF (Clustering feature) to summarize a cluster. CF is a 3D vector summarizing info about cluster objects

CF = (n, LS, SS)

DATE & DAY

MON TUE WED THU FRI SAT SUN

where n is the no. of objects in the cluster, LS is the linear sum of the objects and SS is the sq sum of obj

$$LS = \sum_{i=1}^{n} x_i \qquad SS = \sum_{i=1}^{n} x_i^2$$

Cluster centroid $x_0 = LS/n$

Cluster Radius $R = \sqrt{\frac{\sum_{i=1}^{n} (x_i - x_0)^2}{n}}$

$$\sqrt{\frac{SS}{n} - \frac{LS^2}{n}} \quad \text{OR} \quad \sqrt{\frac{n(SS) - 2LS^2 - n(LS)}{n^2}}$$

Cluster Diameter $D = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)^2}{n(n-1)}}$

$$= \sqrt{\frac{2n(SS) - 2(LS)^2}{n(n-1)}}$$

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

Consider data point $u_1 = (3,4)$

Radius = 0

LS = 3,4

SS = 9,16

CF = 1, (3,4), (9,14)

CF[1,(3,4),(9,14)]

leaf nodes

$u_1 = (3,4)$

For data point

$u_2 = (2,6)$

LS = (3,4) + (2,6)

= (5, 10)

SS = (9,16) + (4,36)

= (13, 52)

n = 2

Radius = $\sqrt{\frac{13,52}{2} - \frac{(5,10)^2}{2}}$

$= \sqrt{(6.5, 26) - (6.25, 25)}$

$= \sqrt{(0.25, 1)} = (0.5, 1)$

DATE & DAY

MON TUE WED THU FRI SAT SUN

R=(0.5, 1) < (T, T) : T=1.5

Therefore u_2 will cluster with leaf u_1

CF1 < 2, (5,10), (13,52)>

For $u_3 = (4,7)$

CF1 < 2, (5,10), (13,52

LS = (4,5) + (5,10)
= (9,15)

SS = (16,25) + (13,52)
= (29,77

R = (0.47, 0.47) < (1.5, 1.5) T $u_4 = 4,7$

CF1 < 3, (9,15), (29,77)>

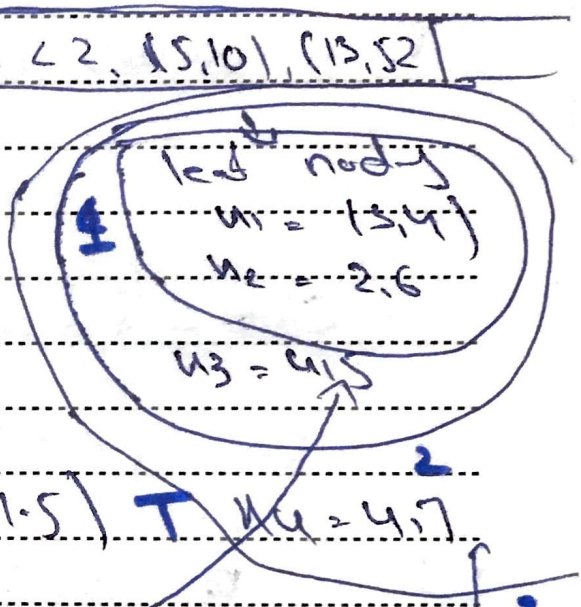
For $u_4 = (4,7)$

LS = (4,7) + (9,15) = (13,22)

SS = (16,49) + (29,77) = (45,126)

R = (0.47, 0.47) < (1.5, 1.5) T

CF1 < 4, (13,22), (45,126)>



stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

For $u_5 = (3, 8)$

LS = (13, 22) + (3, 8) = (16, 30)

SS = (9+45, 64+126) = (54, 190)

R = (0.33, 0.63) < (1.5, 1.5) T

Add the point to the leaf node

CF1 < 5, (16, 30), (54, 190) >

For $u_6 = (6, 2)$

LS = (16, 30) + (6, 2) = (22, 32)

SS = (54, 190) + (36, 4) = (90, 194)

R = (1.24, 1.97) < (1.5, 1.5) F

Create a New leaf

CF2 < 1, (6, 2), (36, 4)

stewart™

Since B=2, only 2 branches will exist

DATE & DAY

MON TUE WED THU FRI SAT SUN

For $u_7 = (7,2)$

CF1 < 5, (16,30), (54,190) >	CF2 ···

leaf nodes

$u_1 = (3,4)$
$u_2 = (2,6)$
$u_3 = (4,5)$
$u_4 = (4,7)$
$u_5 = (3,8)$

leaf nodes

$u_6 = (6,2)$

Firstly we will check u_7 is close to which branch

$CF1 = LS/N = (16,30)/5$

$= (3.2, 6)$

$CF2 = (6,2)/1$

$= 6, 2$

u_7 is close to CF2

$LS = (7,2) + (6,2) = (13,4)$

$SS = (49,4) + (36,4) = (85,8)$

$R = (0.5, 0) < (1.5, 1.5)$ T

$CF2 = <2, (13,4), (85,8)>$

Add point to CF2 leaf node

stewart™

For $u_8 = (7,4)$

$CF1 = (16,30)/5$ $\quad CF2 = \frac{(13,4)}{2}$

$= (3.2, 6)$ $\quad = (6.5, 2)$

Closer to CF2

$LS = (13,4) + (7,4) = (20,8)$

$SS = (49,16) + (85,8) = (134,24)$

$R = (0.47, 0.94) < (1.5, 1.5)$ T

$CF2 < 3, (20,8), (134,24) >$

For $u_9 = (8,4)$

$CF1 = (16,30/5)$ $\quad CF2 = (20,8)/3$

$= (3.2, 6)$ $\quad = (6.6, 2.6)$

Closer to CF2

$LS = (20,8) + (8,4) = (28,12)$

$SS = (64 * 134) + (16 * 24) = (198, 40)$

$R = (0.7, 1) < (1.5, 1.5)$

T

DATE & DAY

MON TUE WED THU FRI SAT SUN

$u_{10} = (7,9)$

$CF1 = (16,30)/5$ $\quad CF2 = (28,12)/4$

$= (3.2, 6)$ $\quad = (7, 3)$

Closer to CF2

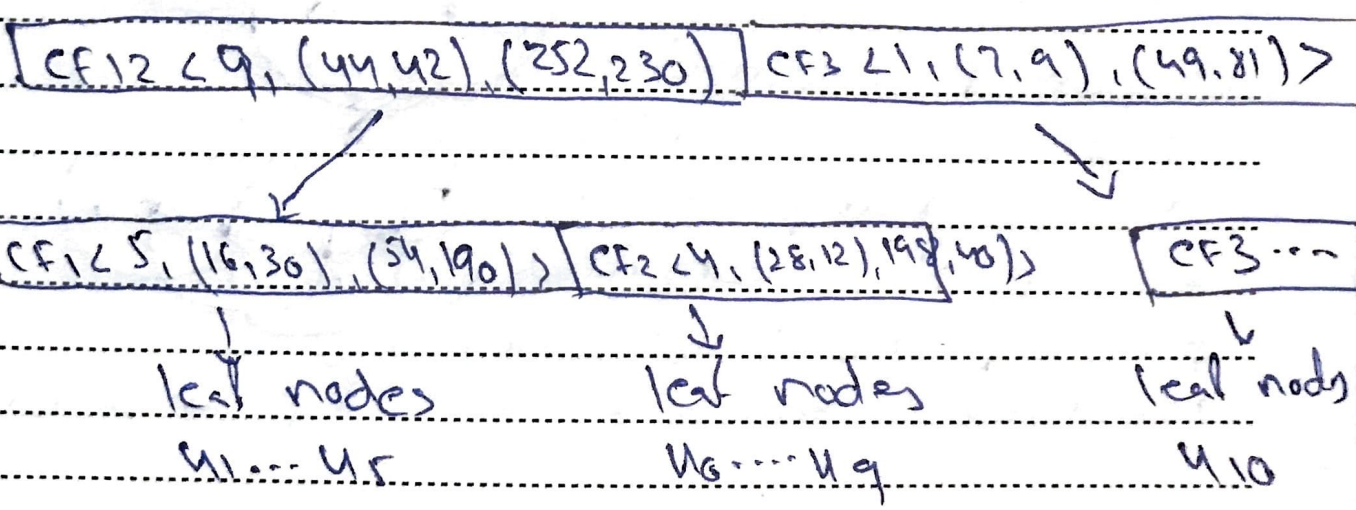
$LS = (16,30) + (7,9) = (23, 39)$

$SS = (54, 190) + (49, 81) = (103, 271)$

$R = (1.57, 1.7) < (1.5, 1.5)$ **F**

create a new CF3

$CF3 < 1, (7,9), (49,81) >$



stewart™

DATE & DAY

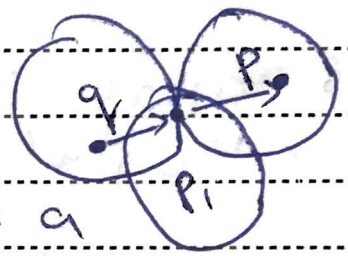
MON TUE WED THU FRI SAT SUN

Density Based Clustering

Eps: Max radius of neighbourhood

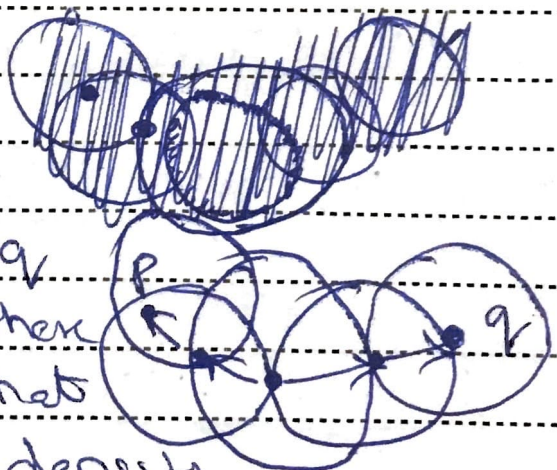
Minpts: Min no. of points in an Eps-neighbour

Density-Reachable



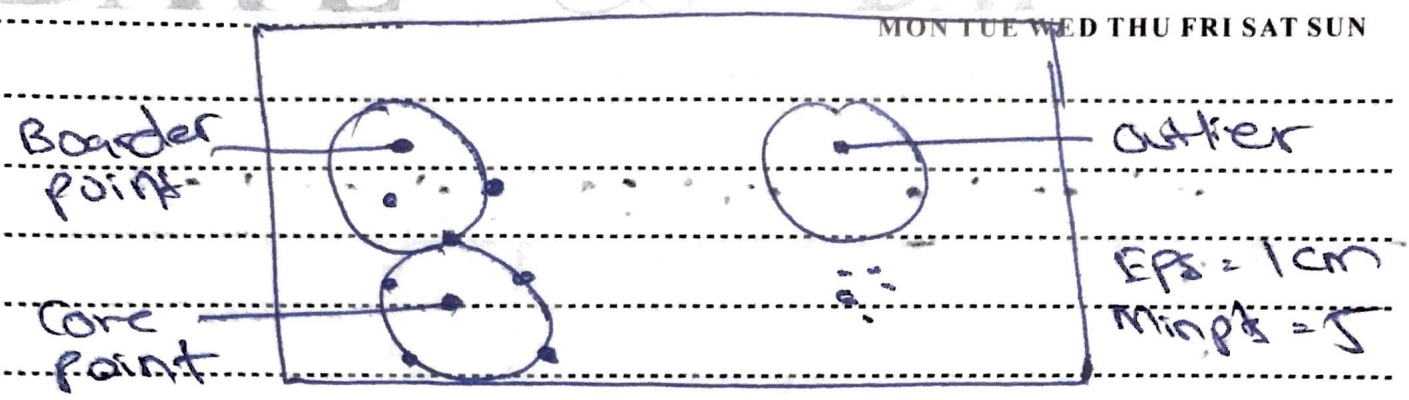
A point p is density-Reachable from q wrt Eps, Minpts if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly-density-reachable from p_i

Density-Connected



A point p is density-connected to a point q wrt Eps, Minpts if there is a point o such that both p and q are density reachable from o wrt Eps and Minpts

stewart™



Core Point: Obj or point which has atleast Minpts obj within the radius of ϵ

Boundary Point: Direct Density Reachable. An Obj or point which is not a core point but it is in the neighbourhood of the core point with the radius of ϵ

Noise Point: An Obj/Point which is neither a core point or boundary point

S1	5	7
S2	8	4
S3	3	3
S4	4	4
S5	3	7
S6	6	7
S7	6	1
S8	5	5

$\epsilon = 3.5$

Minpts = 3

DATE & DAY

MON TUE WED THU FRI SAT SUN

Step 1 Distance Matrix

	S1	S2	S3	S4	S5	S6	S7	S8
S1	0							
S2	4.24	0						
S3	4.47	5.1	0					
S4	3.16	4	1.41	0				
S5	2	5.83	4	3.16	0			
S6	1	3.61	5	3.61	3	0		
S7	6.08	3.61	3.61	3.61	6.71	6	0	
S8	2	3.16	2.83	1.41	2.83	2.24	4.12	0

Step 2 Identify the neighbours of each point for ϵ 3.5

S1: S4, S5, S6, S8

S2: S8

S3: S4, S8

S4: S5, S8, S1, S3

S5: S1, S4, S6, S8

S6: S1, S5, S8

S7: None

S8: S1, S2, S3, S4, S5, S6

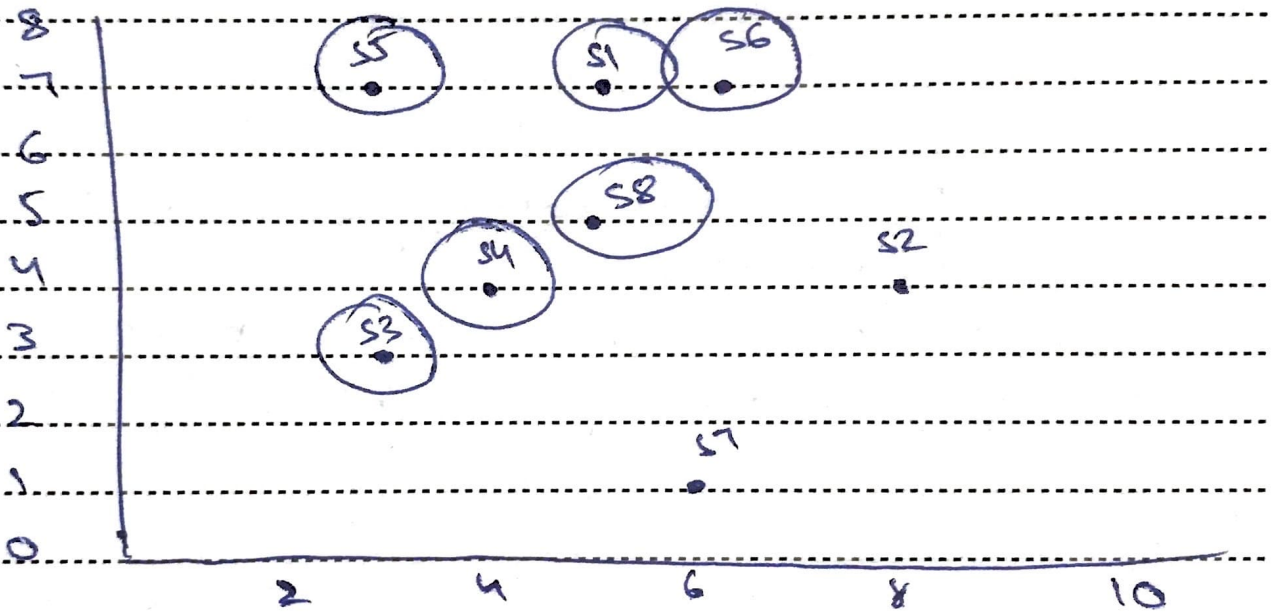
stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

Step: 3 Ascertain whether the point is the core point for Minpts = 3

S1, S3, S4, S5, S6 and S8 are core points



S2 and S7 are noise points. Moreover S2 can be classified as a boundary point since it is direct density reachable from S8

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

Text Classification

Bag of words

→ Represents text as a collection of words
→ Counting occurences in a document / corpus

Cons

- Ignore content and meaning of words which causes ambiguity
- cannot capture rls b/w word such as synonym etc
- creates sparse feature space which causes overfitting

Example

S1 : The cat in the hat
S2 : The dog chased the cat
S3 : The hat was lost

unique words: The, cat, in, hat, dog, chased, was, lost

Vector S1 : 1 1 1 1 0 0 0 0
" S2 : 1 1 0 0 1 1 0 0
" S3 : 1 0 0 1 0 0 1 1

DATE & DAY

MON TUE WED THU FRI SAT SUN

Vector Space Model

-> Represents texts as vectors of features, where each feature represents a term or word in the text and each vector represents a document

Cons

- Sparce Representation of the data
- doesn't capture the content and meaning of word
- Suffer from curse of dimensionality

Example

D1 : The cat in the hat vector D1 : 11110000

D2 : The dog chased the cat vector D2 : 11001100

D3 : The hat was lost vector D3 : 10010011

unique words : The, cat, in, hat, dog, chased, was, lost

Similarity b/w D1 and D2

$$\frac{1*1 + 1*1 + 1*0 + 1*0 + 0*1 + 0*1 + 0*0 + 0*0}{\sqrt{1^2+1^2+1^2+1^2}\sqrt{1^2+1^2+1^2+1^2}}$$

$$= \frac{2}{\sqrt{4}\sqrt{4}} =$$

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

Term Frequency

→ How often a terms appears in a doc. TF weighting is the process of normalizing the TF by taking into account the length of doc, so that longer docs do not have an unfair advantage over shorter docs

→ Give more imp to terms that are relevant

1+log(TF) OR log(TF)

Example

	disease	symptom	osteoporosis	
Doc 1	20	15	1	36
Doc 2	3	6	13	26

1+log(TF)

	Disease	symptom	osteoporosis	
Doc1	2.3	2.2	1	5.5
Doc2	1.47	2	2.1	5.57

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

Inverse Document Frequency

→ Measure how imp a term is in the corpus or collection of documents

→ IDF weighting gives more weight to terms that are rare in the corpus and less weight to the terms that are common

$$IDF = \log(N/df(t)) \text{ OR } 1 + \log(N/df(t))$$

where N is the total no. of docs in collect and df(t) is the no. of docs containing the term t

Example

Suppose N = 100,000

	df(t)	IDF
disease	2000	$1+\log(100,000/2000) = 2.7$
symptom	300	$1+\log(100,000/300) = 3.5$
osteoporosis	10	$1+\log(100,000/10) = 5$

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

TF-IDF

→ Measures the imp of a term in a doc within a corpus

→ Gives higher weigh to terms that are frequent in the doc but rare in corpus and lower weight to terms that are frequent in both doc and corpus

→ Biased towards longer doc and give more weights to words like "the" or "and"

TF-IDF = TF * IDF

Example

1+log(TF)

	Disease	symptom	Osteoporosis
Doc 1	2.3	2.2	1
Doc 2	1.47	2	2.1

	df(t)	IDF
Disease	2000	2.7
Symptom	300	3.5
Osteoporosis	10	5

TF-IDF Doc1 : 2.3*2.7+2.2*3.5+1*5 = 18.9

TF-IDF Doc2 : 1.47*2.7+2*3.5+2.1*5 = 21.5

stewart™

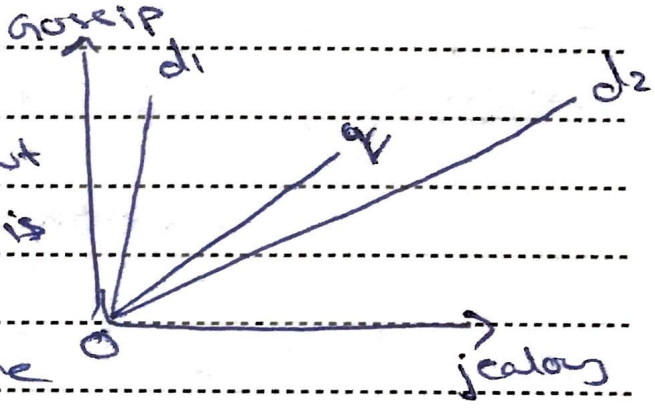
DATE & DAY

MON TUE WED THU FRI SAT SUN

Cosine Similarity

→ Unaffected by the length of doc
→ Robust measure of similarity that can handle high-dimensional vector spaces

Dist of terms in doc 2 is similar to the query q, but if dist was used then d_2 is very different from q. Therefore we normalize the length



Angel to cousine

Text is converted into vectors and vector contains count of words. word count cannot be neg therefore angle to cousine would always be positive and would range from 0-1

$\cos(v_{q_1}, v_d) = Y_q \times Y_d$

$\cos(\vec{q}_1, \vec{q}_2) = \frac{q \cdot d}{|\vec{q}_1| |\vec{d}|} = \frac{\vec{q}}{|\vec{q}_1|} \cdot \frac{\vec{d}}{|\vec{d}|}$

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

Example 1 Doc to Doc

term	Doc1	Doc2	Doc 3
affection	115	58	20
jealous	10	7	11
Gossip	2	0	6
wuthering	0	0	38

1+log(TF)

term	Doc 1	Doc 2	Doc 3
affection	3.06	2.76	2.30
jealous	2.00	1.85	2.04
gossip	1.30	0	1.78
wuthering	0	0	2.58

length of Doc1 = $\sqrt{(3.06)^2 + (2)^2 + (1.30)^2 + 0^2} = 3.89$

length of Doc2 = $\sqrt{(2.76)^2 + (1.85)^2 + 0^2 + 0^2} = 3.32$

length of Doc3 = $\sqrt{(2.30)^2 + (2.04)^2 + (1.78)^2 + (2.58)^2} = 4.40$

length Normalization

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

Term	Doc1	Doc2	Doc 3
affection	3.06/3.89=0.78	2.76/3.32 =0.832	2.30/4.40 = 0.524
jealous	2/3.89 = 0.51	0.555	0.465
gossip	1.30/3.89 = 0.35	0	0.405
wuthering	0	0	0.588

cos(Doc1, Doc2) = 0.78 * 0.832 + 0.51 * 0.555 + 0.35 * 0 + 0 * 0

= 0.94

cos(Doc1, Doc3) = 0.79

cos(Doc2, Doc3) = 0.69

Example 2 Doc and query

Doc: car insurance auto insurance

q: best car insurance

Term	Query		Doc
	tf-raw	df	tf-raw
Auto	0	5000	1
best	1	50000	0
car	1	10000	1
insurance	1	1000	2

N = 1 million

stewart™

Product
= Query TF-IDF * Doc TF-weight

tf-weight = 1 + log(TF)

IDF = log(N / df(t))

Term	Query tf-raw	Query tf-weight	df	Query IDF	Document tf-raw	Document tf-weight	Product
Auto	0	0	5000	2.3	1	1	0
best	1	1	50000	1.3	0	0	0
car	1	1	10000	2.0	1	1	2
Insurance	1	1	1000	3.0	2	1.3	3.9

Doc length = $\sqrt{1^2 + 0^2 + 1^2 + 1.3^2}$

= 1.92

Query TF-IDF

0 * 2.3 = 0

1 * 1.3 = 1.3

1 * 2.0 = 2.0

1 * 3.0 = 3.0

Score = (0 + 0 + 2 + 3.9) / 1.92

= 3.07

Centroid Rocchio Classifier

→ Text classification that involves representing each class by its centroid vector and then classify new docs based on their proximi to the class centroids

→ Useful for large-scale classification with large no. of classes

→ Relatively simple to implement and

→ computationally efficient

Cons

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

→ Assumes that each class is linearly seperable from the others

→ each class is represented by a singk centroid vector

→ May not perform well as more complex classification algos on some tasks

Large margin Classifier

→ Finds a hyperplan that maximizes the margin b/w different classes

→ Mostly used in binary classification problems

Pros	Cons
• High Accuracy and Robustness	• Computationally expensive for large datasets
• Can handle noisy or overlapping data	• Careful selection of hyper-parameters
• Applicable to wide range of classification problems	• Not perform well on imbalanced datasets
• Effective with high-dimensional data	• limited to linearly seperable classes and may require complex transformations to handle non-linear decision boundaries

stewart™

DATE & DAY

MON TUE WED THU FRI SAT SUN

Passive Aggressive Algo

→ Margin-based algo used for binary classification
→ Similar to SVM but with some key differences in the way it updates its model parameters
→ makes an aggressive update to the model if a data point is misclassified and passive update if correctly classified

Pros	Cons
• well-suited for online learning scenerios	• Sensitive to choice of hyperparameters
• Fast and efficient updates to model param	• Require more iterations to converge
• Robust to noisy and mis- labeled data	• limited to binary classification

Support Vector Machine

→ Supervised ML Algo that works by identifying a hyperplan in a high-dimensional space that can best separate the data points into different classes
→ Handle non-linear decision boundaries through the use of kernel functions

DATE & DAY

MON TUE WED THU FRI SAT SUN

Pros	Cons
• High accuracy	• computationally expensive
• Robust to overfitting	• May not perform well with imbalanced data
• Works well with high dimensional data	• Careful selection of hyperparameters
• Can handle both binary and multi-class data	• Can be sensitive to outliers
• Effective in handling non-linear decision boundaries	