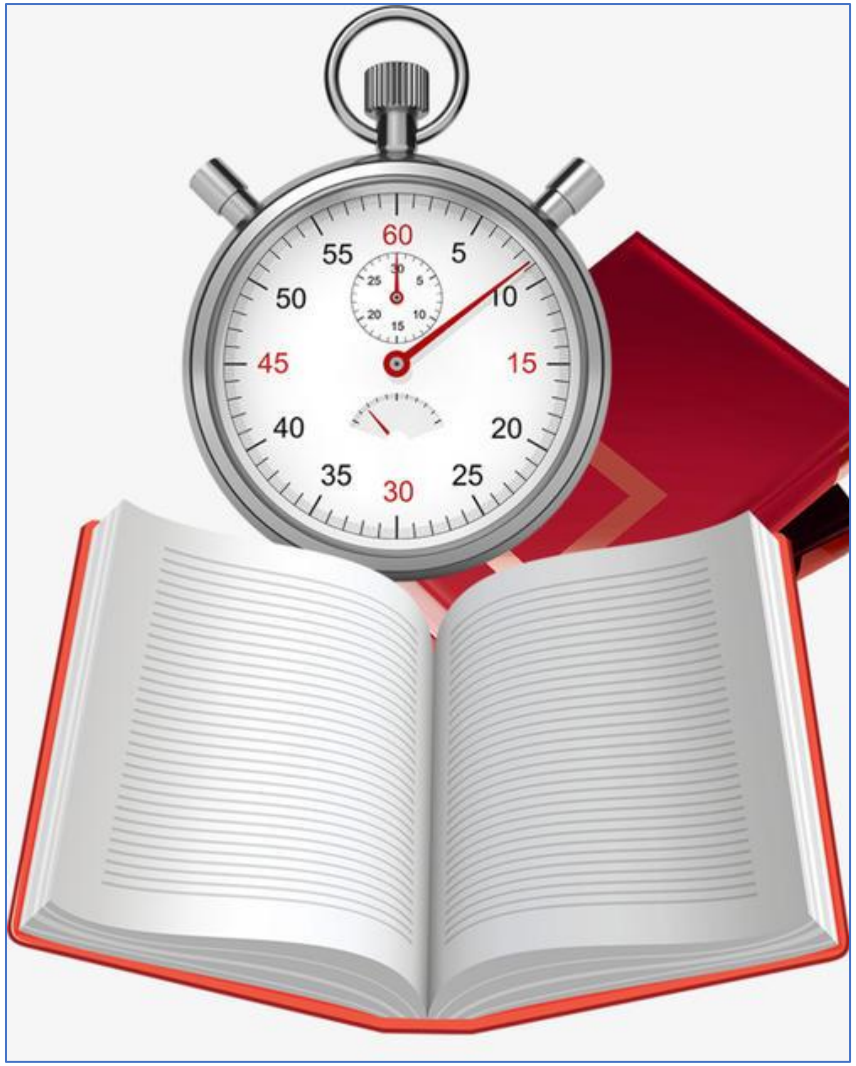
CS 103 -13
Machine Learning

Jimmy Liu 刘江

2020-12-11

Review



1. Homework

- In a document classification applications, 7 text document objects from 3 categories of articles are represented by the points, which are:

Point	**1**	**2**	**3**	**4**	**5**	**6**	**7**
X	1	8	7	9	1	9	5
Y	6	3	1	8	2	3	6

- please use K-means algorithm to cluster the 7 document into three categories of articles (K=3)
- please specify which points belong to which cluster. For easy calculation, the initial three starting cluster center points are point 1, point 2 and point 3, also, please use L2 Distance to calculate the distance between two points
- The L2 distance between two points a=(x1, y1) and b=(x2, y2) is defined as:

$$d_{a,b} = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

1. Hint

		(1, 6)	(8, 3)	(7, 1)	
Point		Distance to centroid of cluster 1	Distance to centroid of cluster 2	Distance to centroid of cluster 3	Cluster
1	(1, 6)	0	7.62	7.81	**1**
2	(8, 3)				
3	(7, 1)				
4	(9, 8)				
5	(1, 2)				
6	(9, 3)				
7	(5, 6)				

Iteration 1

		(1,6)	(8,3)	(7,1)	
Point		Distance to centroid of cluster of 1	Distance to centroid of cluster of 2	Distance to centroid of cluster of 3	Cluster
1	(1,6)	0.00	7.62	7.81	1
2	(8,3)	7.62	0.00	2.24	2
3	(7,1)	7.81	2.24	0.00	3
4	(9,8)	8.25	5.10	7.28	2
5	(1,2)	4.00	7.07	6.08	1
6	(9,3)	8.54	1.00	2.83	2
7	(5,6)	4.00	4.24	5.39	1

Cluster ID	1	2	3
New centroid	(2.33,4.67)	(8.67,4.67)	(7.00,1.00)

Iteration 2

		(2.33,4.67)	(8.67,4.67)	(7.00,1.00)	
Point		Distance to centroid of cluster of 1	Distance to centroid of cluster of 2	Distance to centroid of cluster of 3	Cluster
1	(1,6)	1.89	7.78	7.81	1
2	(8,3)	5.91	1.80	2.24	2
3	(7,1)	5.93	4.03	0.00	3
4	(9,8)	7.45	3.35	7.28	2
5	(1,2)	2.98	8.12	6.08	1
6	(9,3)	6.87	1.70	2.83	2
7	(5,6)	2.98	3.90	5.39	1

Since none of the inputs changed cluster in the last iteration, we exit.

Result

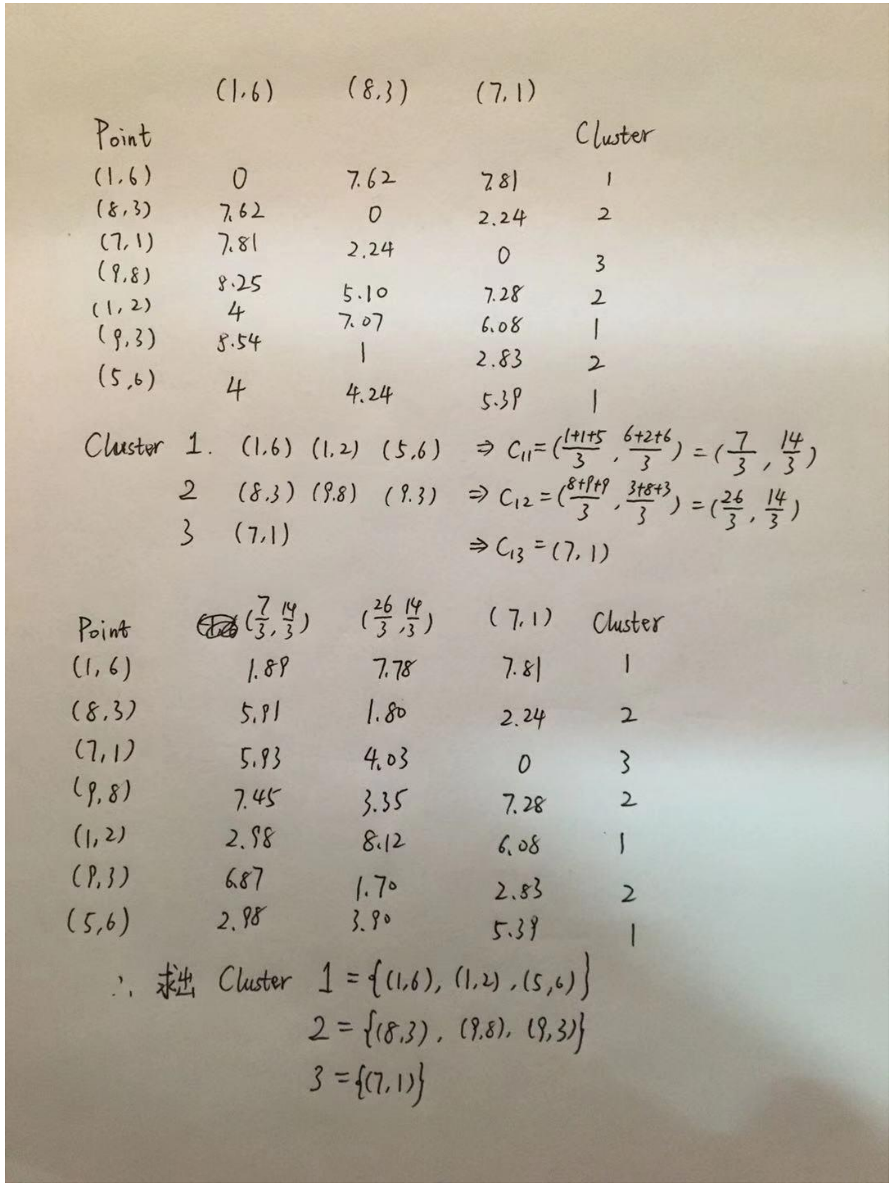
Cluster ID	1	2	3
Centroid of cluster	(2.33,4.67)	(8.67,4.67)	(7.00,1.00)

Point ID	1	2	3	4	5	6	7
Cluster ID	1	2	3	2	1	2	1

Source Code

```python
#! /usr/bin/env python3
# -*- coding: utf-8 -*-
import numpy as np
import math


def dist(p1, p2):
    x1, y1 = p1
    x2, y2 = p2
    dx = x1-x2
    dy = y1-y2
    return math.sqrt(dx*dx+dy*dy)


if __name__ == "__main__":
    data = [(1, 6), (8, 3), (7, 1), (9, 8), (1, 2), (9, 3), (5, 6)]
    siz = len(data)
    idx = [0, 0, 0, 0, 0, 0, 0]
    center = [(1, 6), (8, 3), (7, 1)]
    class_num = len(center)

    it = 0
    while True:
        it += 1
        print("Iteration {}".format(it))
        cidx = []
        for p in data:
            d = list(map(lambda c: dist(p, c), center))
            c = d.index(min(d))
            print("{} {} {:.2f} {:.2f} {:.2f} {}".format(
                len(cidx)+1, p, d[0], d[1], d[2], c+1))
            cidx.append(c)
        if idx == cidx:
            break
        idx = cidx
        for i in range(class_num):
            cnt = 0
            sx = 0
            sy = 0
            for j in range(siz):
                if idx[j] == i:
                    cnt += 1
                    sx += data[j][0]
                    sy += data[j][1]
            if cnt == 0:
                raise "Assertion cnt!=0 failed."
            center[i] = (sx/cnt, sy/cnt)
        for i in range(class_num):
            print("{} {:.2f} {:.2f}".format(i+1, center[i][0], center[i][1]))

    print("Result", end=" ")
    for i in range(siz):
        print(idx[i]+1, end=" ")

    print()
```

做对且过程详细



Point	(1,6)	(8,3)	(7,1)	Cluster
(1,6)	0	7.62	7.81	1
(8,3)	7.62	0	2.24	2
(7,1)	7.81	2.24	0	3
(9,8)	8.25	5.10	7.28	2
(1,2)	4	7.07	6.08	1
(9,3)	8.54	1	2.83	2
(5,6)	4	4.24	5.39	1

Cluster 1. (1,6) (1,2) (5,6) $\Rightarrow C_{11} = (\frac{1+1+5}{3}, \frac{6+2+6}{3}) = (\frac{7}{3}, \frac{14}{3})$

2 (8,3) (9,8) (9,3) $\Rightarrow C_{12} = (\frac{8+9+9}{3}, \frac{3+8+3}{3}) = (\frac{26}{3}, \frac{14}{3})$

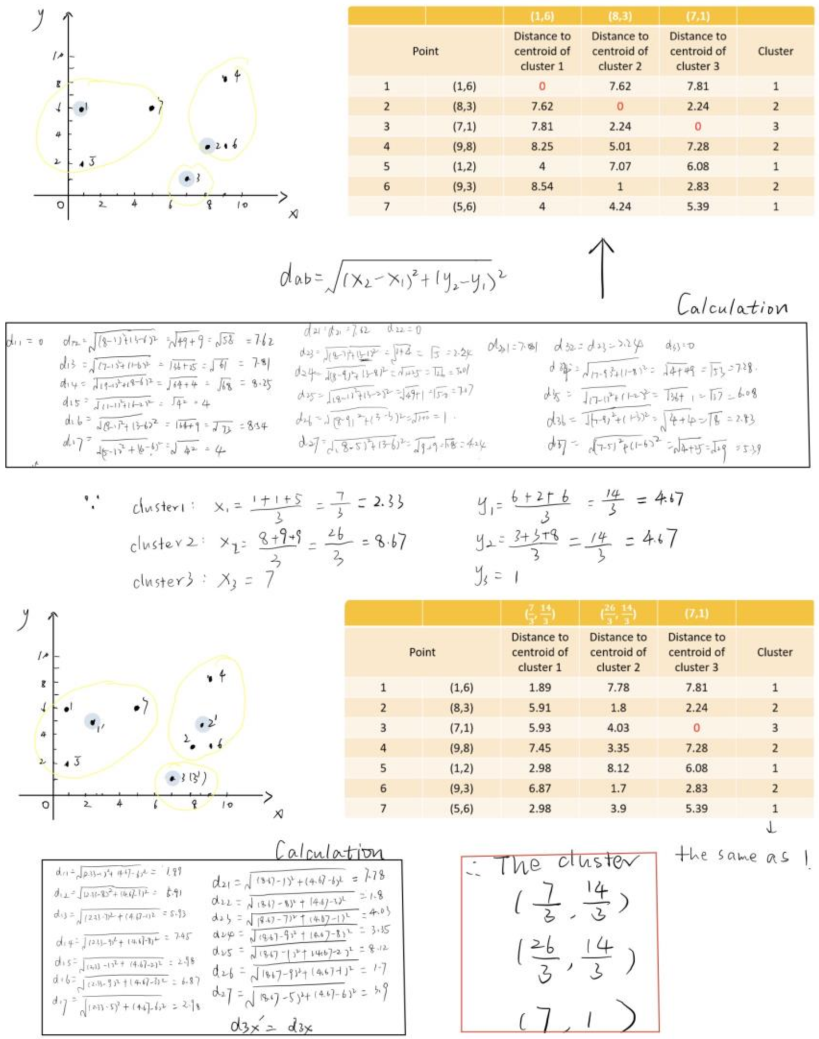
3 (7,1) $\Rightarrow C_{13} = (7,1)$

Point	$(\frac{7}{3},\frac{14}{3})$	$(\frac{26}{3},\frac{14}{3})$	(7,1)	Cluster
(1,6)	1.89	7.78	7.81	1
(8,3)	5.91	1.80	2.24	2
(7,1)	5.93	4.03	0	3
(9,8)	7.45	3.35	7.28	2
(1,2)	2.98	8.12	6.08	1
(9,3)	6.87	1.70	2.83	2
(5,6)	2.98	3.90	5.39	1

∴ 故 Cluster 1 = {(1,6), (1,2), (5,6)}

2 = {(8,3), (9,8), (9,3)}

3 = {(7,1)}



	Point	(1,6) Distance to centroid of cluster 1	(8,3) Distance to centroid of cluster 2	(7,1) Distance to centroid of cluster 3	Cluster
1	(1,6)	0	7.62	7.81	1
2	(8,3)	7.62	0	2.24	2
3	(7,1)	7.81	2.24	0	3
4	(9,8)	8.25	5.01	7.28	2
5	(1,2)	4	7.07	6.08	1
6	(9,3)	8.54	1	2.83	2
7	(5,6)	4	4.24	5.39	1

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

Calculation

∵ cluster1: $x_1 = \frac{1+1+5}{3} = \frac{7}{3} = 2.33$ $y_1 = \frac{6+2+6}{3} = \frac{14}{3} = 4.67$

cluster2: $x_2 = \frac{8+9+9}{3} = \frac{26}{3} = 8.67$ $y_2 = \frac{3+3+8}{3} = \frac{14}{3} = 4.67$

cluster3: $x_3 = 7$ $y_3 = 1$

	Point	$(\frac{7}{3},\frac{14}{3})$ Distance to centroid of cluster 1	$(\frac{26}{3},\frac{14}{3})$ Distance to centroid of cluster 2	(7,1) Distance to centroid of cluster 3	Cluster
1	(1,6)	1.89	7.78	7.81	1
2	(8,3)	5.91	1.8	2.24	2
3	(7,1)	5.93	4.03	0	3
4	(9,8)	7.45	3.35	7.28	2
5	(1,2)	2.98	8.12	6.08	1
6	(9,3)	6.87	1.7	2.83	2
7	(5,6)	2.98	3.9	5.39	1

Calculation

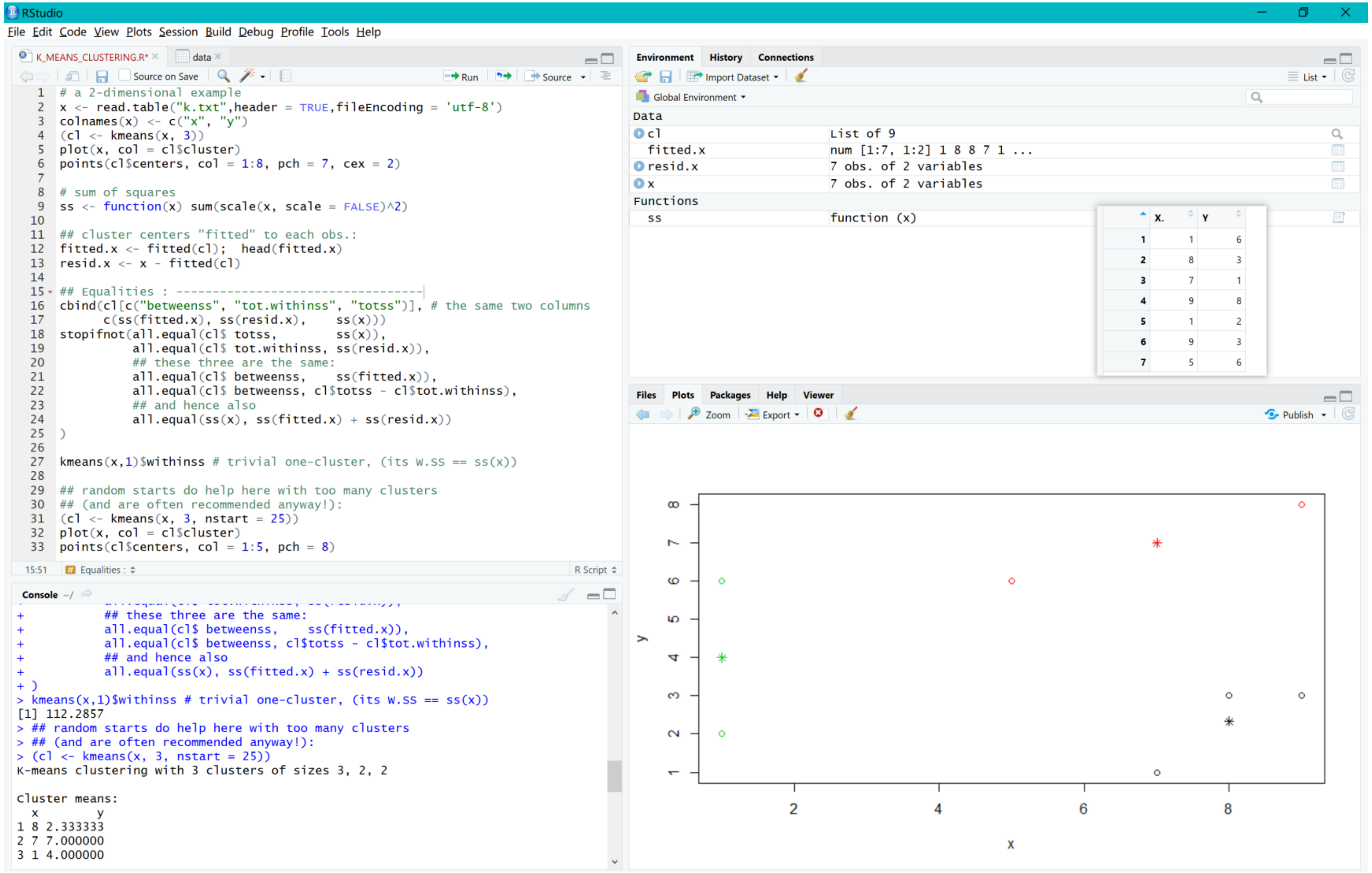
$d_{3x}' = d_{3x}$

∴ The cluster $(\frac{7}{3}, \frac{14}{3})$ $(\frac{26}{3}, \frac{14}{3})$ $(7, 1)$

the same as 1

	Point	(1,6) distance	(8,3) distance	(7,1) distance	cluster
1	(1,6)	0	7.62	7.81	1
2	(8,3)	7.62	0	2.24	2
3	(7,1)	7.81	2.24	0	3
4	(9,8)	8.25	5.10	7.28	2
5	(1,2)	4	7.07	6.08	1
6	(9,3)	8.54	1	2.83	2
7	(5,6)	4	4.24	5.39	1

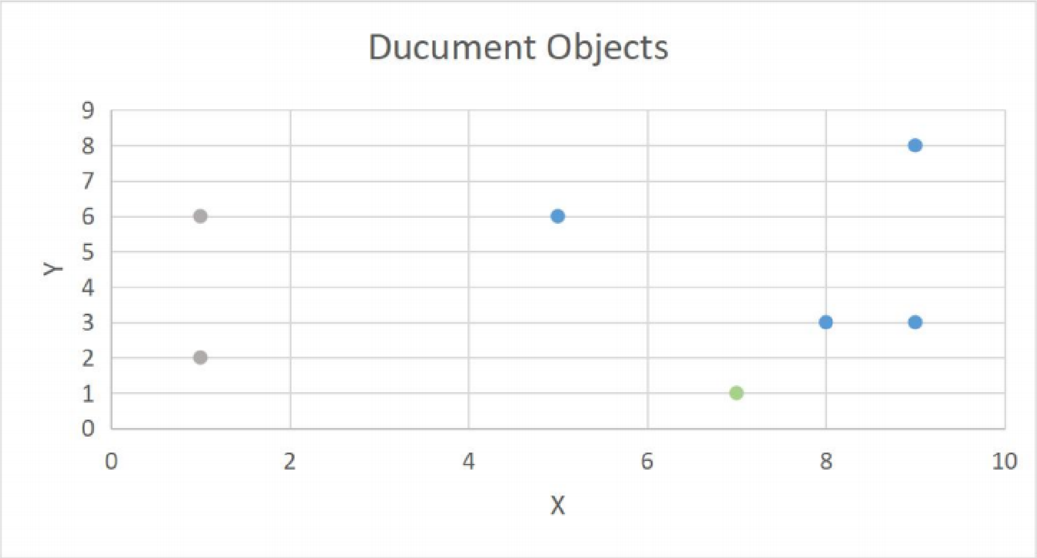
只有一次迭代，没有计算更新后的centers



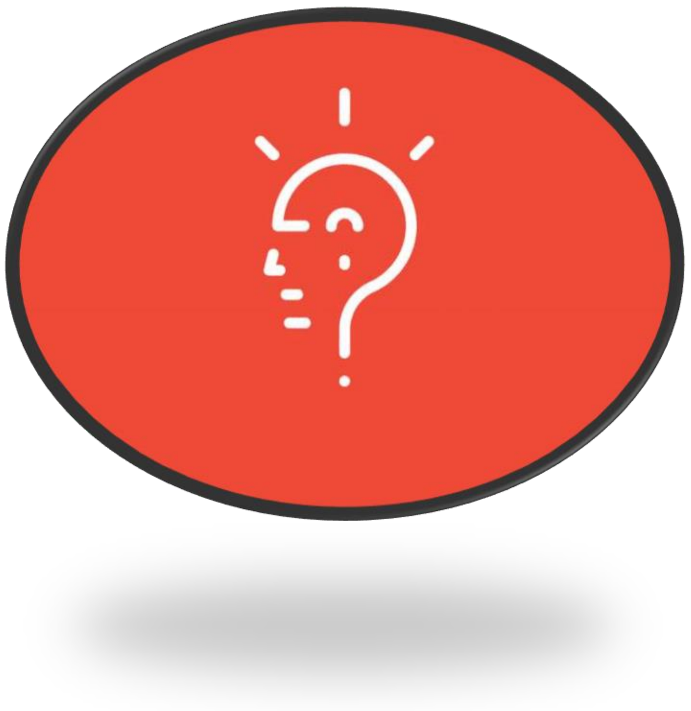
初始点和题目要求不一致导致最终聚类结果不一致

小错误，（5,6）点归错类了

Point	1	2	3	4	5	6	7
X	1	8	7	9	1	9	5
Y	6	3	1	8	2	3	6
D1	0	7.6158	7.8102	8.2462	4	8.544	4
D2	7.6158	0	2.2361	5.099	7.0711	1	4.2426
D3	7.8102	2.2361	0	7.2801	6.0828	2.8284	5.3852
Cluster	1	2	3	2	1	2	2



Any Question?



投票打分群


中国联通
下午1:20
群二维码名片
人工智能导论下午班-presentation
该二维码7天内(12月18日前)有效，重新进入将更新

Machine Learning

3 5

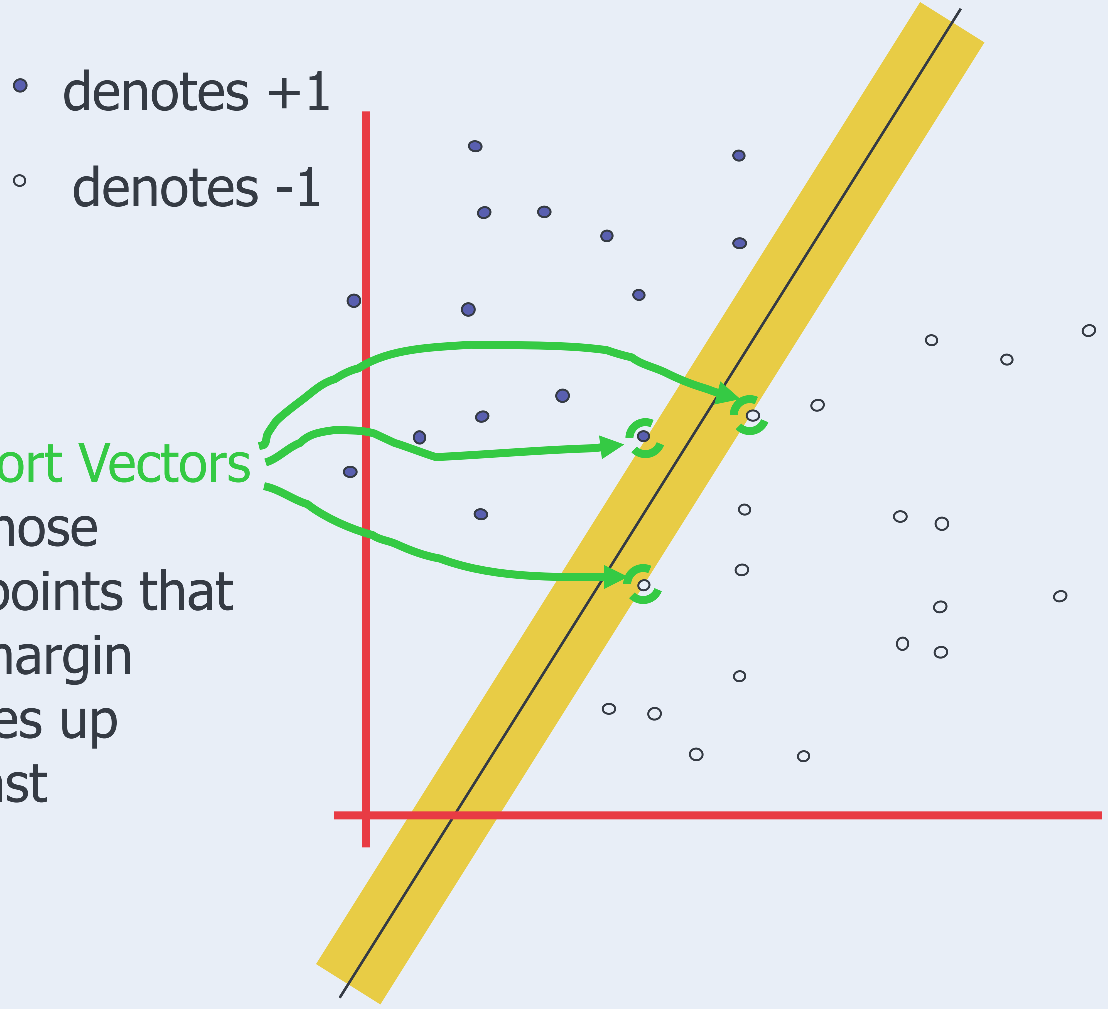
1 Back Propagation

2 Supporter Vector Machine

3 Machine Learning

4 Knowledge

$f(x,w,b) = \text{sign}(w.\ x + b)$



Support Vectors are those datapoints that the margin pushes up against

Margin of a linear classifier is the width that the boundary could be increased by before hitting a datapoint.

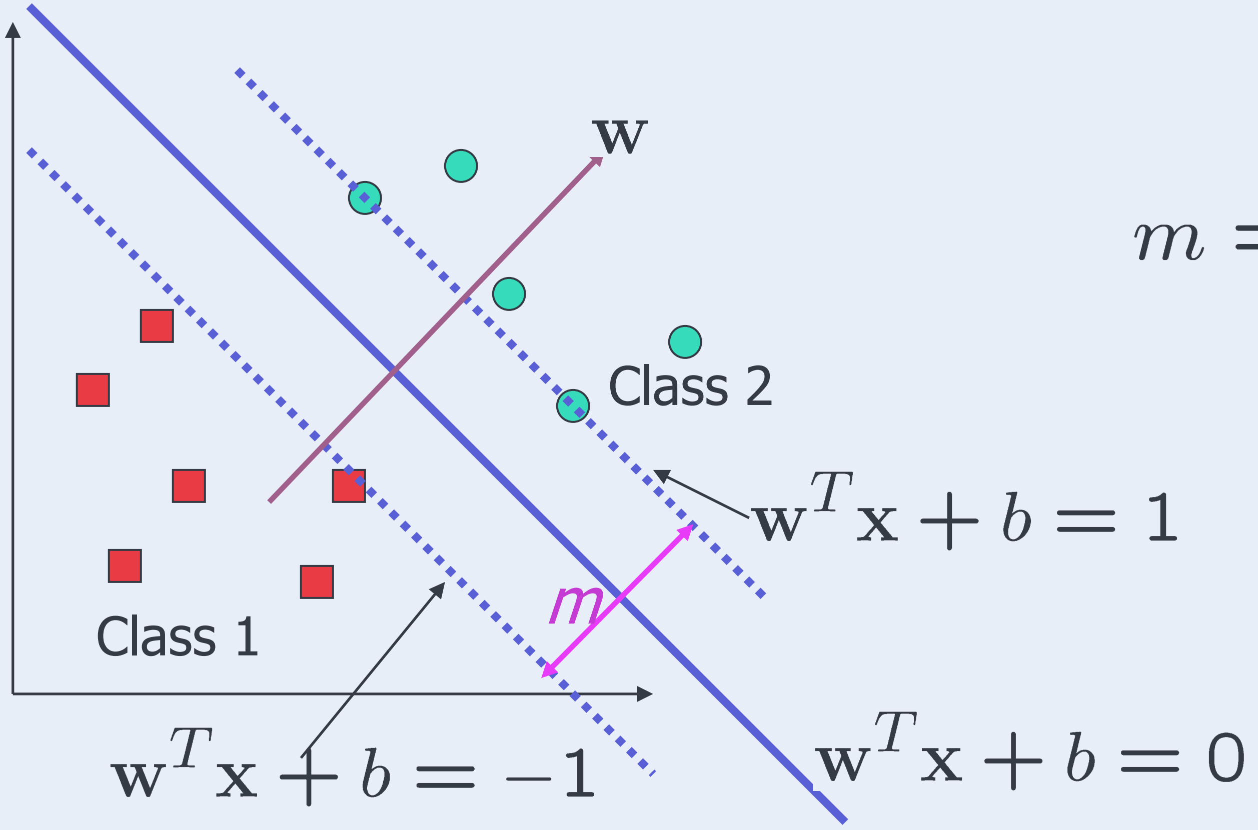
The maximum margin linear classifier is the Linear SVM (LSVM)

Margin m

The decision boundary should be as far away from the data of both classes as possible

We should maximize the margin *m: smallest distance from observations to hyperplane*

Distance between the origin and the line $\mathbf{w}^t\mathbf{x}=-b$ is $b/||\mathbf{w}||$



$$m = \frac{2}{||\mathbf{w}||}$$

Solve SVM by Decision Boundary (Max Margin)

- Let $\{x_1, \ldots, x_n\}$ be our data set and let $y_i \in \{1,-1\}$ be the class label of x_i
- The decision boundary should classify all points correctly

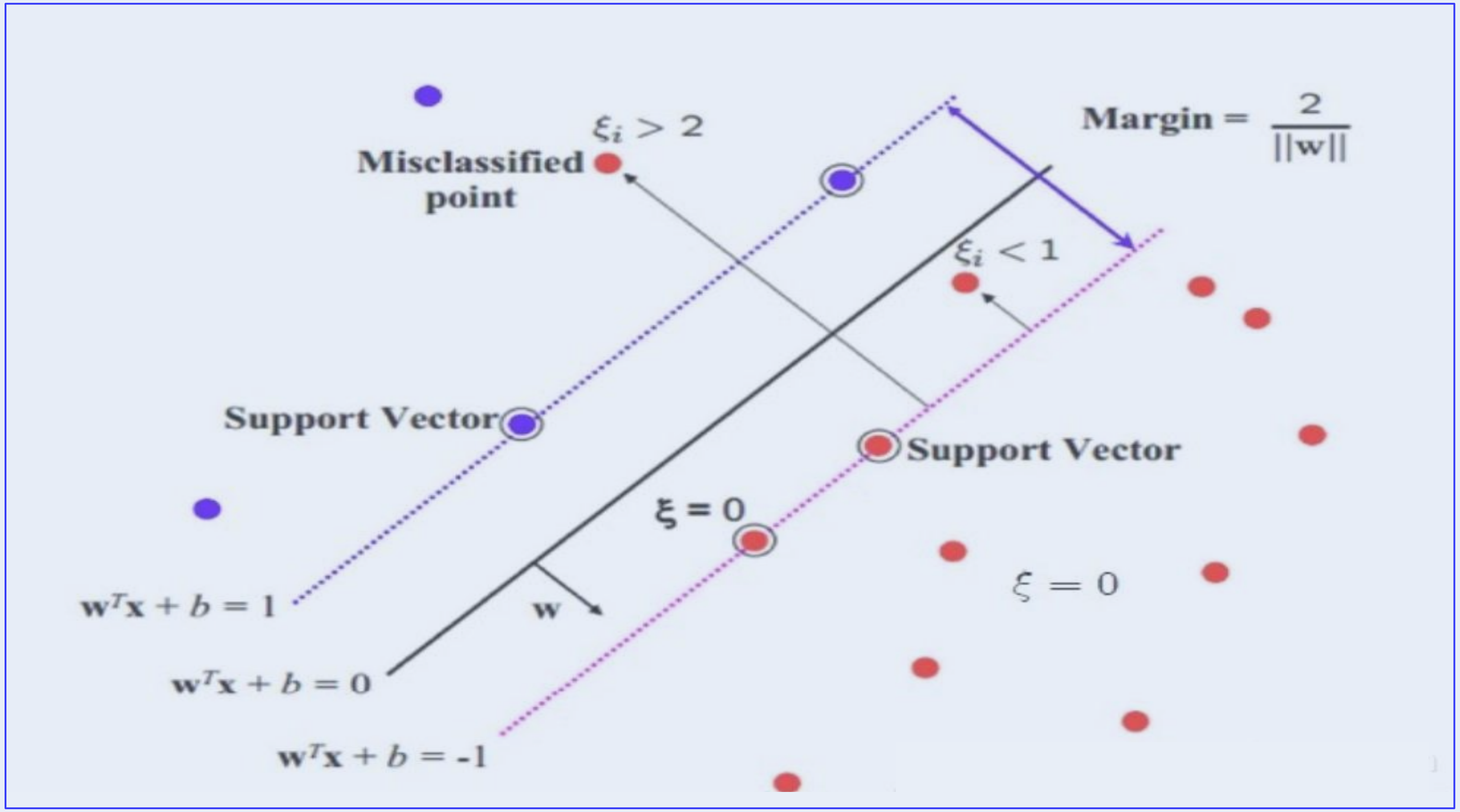
$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \qquad \forall i$$

- To see this: when y=-1, we wish (wx+b)<1, when y=1, we wish (wx+b)>1. For support vectors, we wish y(wx+b)=1.
- The decision boundary can be found by solving the following constrained optimization problem

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

SVM with Slacks for Red Class



Hinge Loss

In machine learning, the hinge loss is a loss function used for training classifiers. The hinge loss is used for "maximum-margin" classification, most notably for support vector machines (SVMs). For an intended output $t = \pm 1$ and a classifier score y, the hinge loss of the prediction y is defined as

$$\ell(y)=\max(0,1-t\cdot y)$$

Note that y should be the "raw" output of the classifier's decision function, not the predicted class label.

Wikipedia

SVM Unique Solution

Optimization Problem

(Cortes and Vapnik, 1995)

Constrained optimization:

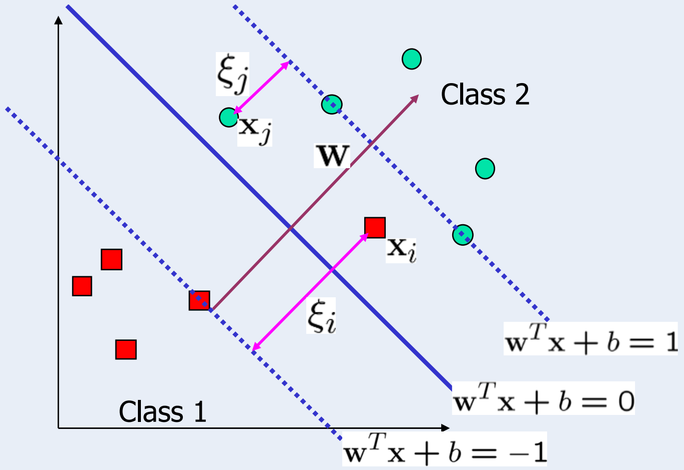
$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

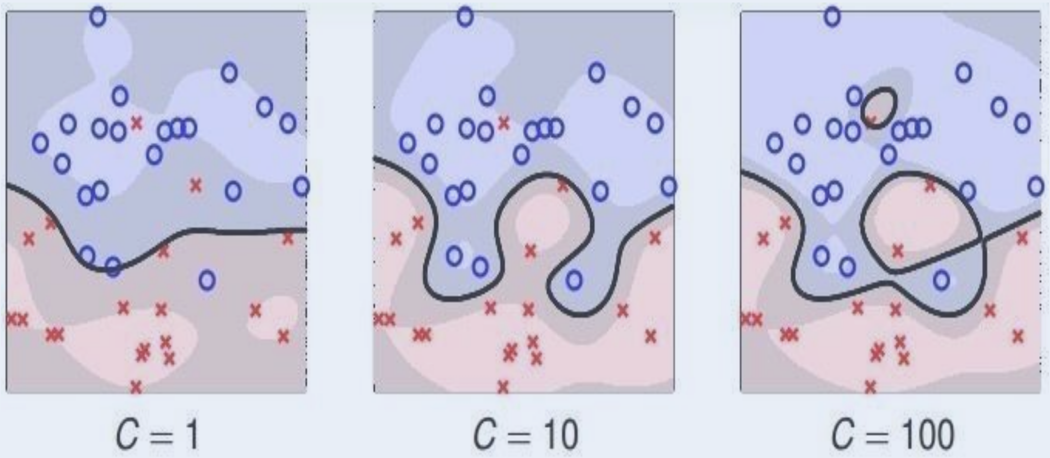
$$\text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \;\wedge\; \xi_i \geq 0, i \in [1, m].$$

Properties:

- $C \geq 0$ trade-off parameter.
- Convex optimization.
- Unique solution.

SVM with Soft Margins

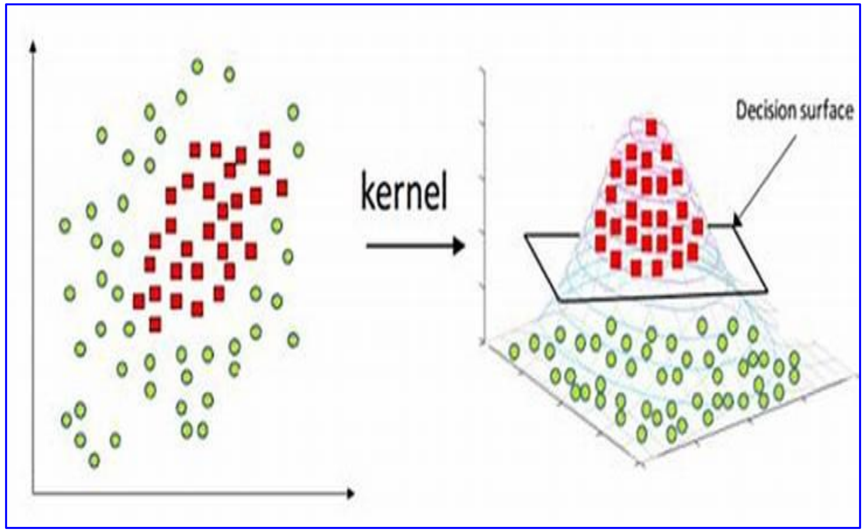


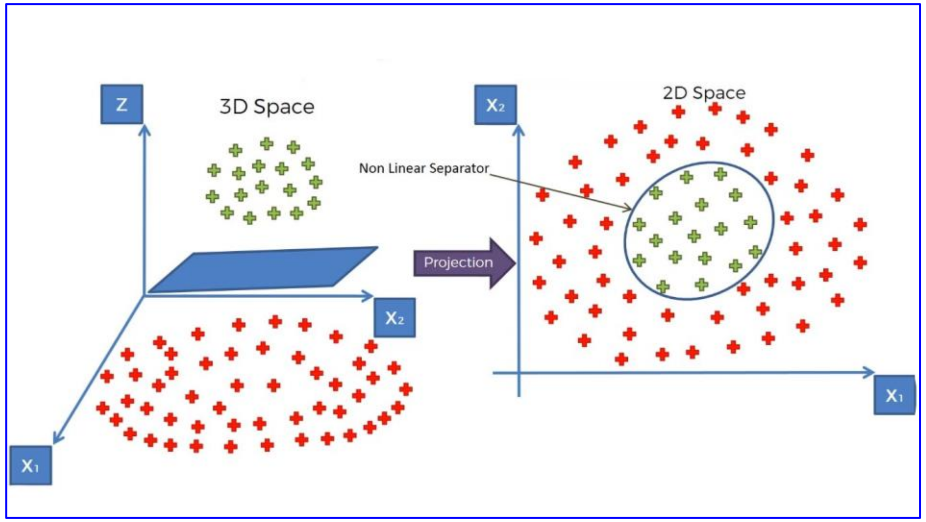


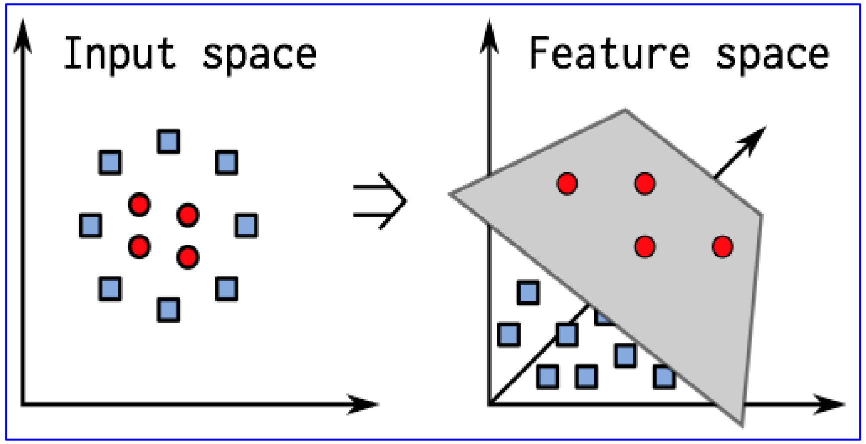
soft-margin SVM: $\min_{b,\mathbf{w},\xi} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \cdot \sum_{n=1}^{N} \xi_n$

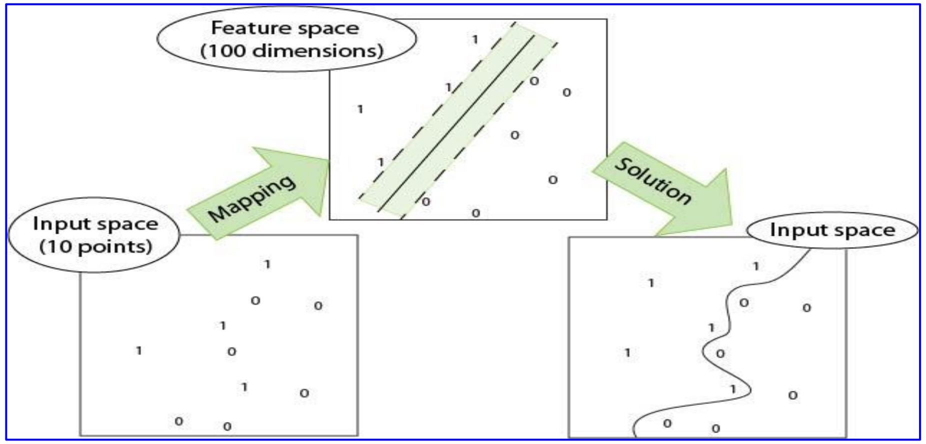
s.t. $\quad y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1 - \xi_n$ and $\xi_n \geq 0$ for all n

Non-Linear SVM Classifier with Kernel Mapping









Kernel Trick

Kernel Trick is an approach consisting in the use of **kernel functions**, operating in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply **computing the inner products between the images of all pairs of data in the feature space**.

Kernel Trick

Directly computing $K(x, z)$ can be faster than "feature transformation + inner product" sometimes.

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ \vdots \\ z_k \end{bmatrix}$$

$$K(x, z) = (x \cdot z)^2$$

$$= (x_1 z_1 + x_2 z_2 + \cdots + x_k z_k)^2$$

$$= {x_1}^2 {z_1}^2 + {x_2}^2 {z_2}^2 + \cdots + {x_k}^2 {z_k}^2$$

$$+ 2x_1 x_2 z_1 z_2 + 2x_1 x_3 z_1 z_3 + \cdots$$

$$+ 2x_2 x_3 z_2 z_3 + 2x_2 x_4 z_2 z_4 + \cdots$$

$$= \phi(x) \cdot \phi(z)$$

$$\phi(x) = \begin{bmatrix} {x_1}^2 \\ \vdots \\ {x_k}^2 \\ \sqrt{2} x_1 x_2 \\ \sqrt{2} x_1 x_3 \\ \vdots \\ \sqrt{2} x_2 x_3 \\ \vdots \end{bmatrix}$$

Similarities of SVM and NN – Non-linear and Parametric

Both SVM and NN can **map the input data to a higher dimensional space to assign a decision boundary**. Both classes of algorithms can approximate non-linear decision functions, with different approaches.

For SVM, it is done by using kernel tricks whereas
NN via non-linear activation functions.

Both parametric, though for different reasons.

In the case of the SVM, the typical parameters are:

- **the soft-margin parameter** C
- **the parameter of the kernel function** γ

Neural networks also use parameters, though they require significantly more of them.

- **The most important parameters concern the number of layers and their size, but also the number of training epochs and the learning rate**.

Both models are similar insofar as they are both parametric, but dissimilar with regards to the type and number of parameters that they require.

Machine Learning

3 5

1 Back Propagation

2 Supporter Vector Machine

3 Machine Learning

4 Knowledge

TOP 10 Machine Learning Algorithms

1. **K-MEANS CLUSTERING**

   Aims to find groups in given data set. The number of groups is represented by a variable called K.

2. **NAIVE BAYES CLASSIFIER**

   A family of algorithms which assume that values of the features used in the classification are independent.

3. **K-NEAREST NEIGHBOR (KNN)**

   A simple algorithm that stores all existing data objects and classifies the new data objects based on a similarity measure.

TOP 10 Machine Learning Algorithms

4

SUPPORT VECTOR MACHINES (SVM)

Used to sort two data sets by similar classification. Draw lines (hyperplanes) that separate the groups according to some patterns.

5

DECISION TREE

A machine learning technique for data mining that creates classification or regression models in the shape of a tree structure.

6

GENERALIZED LINEAR MODELS (GLM)

Combines a number of models including linear regression models, logistic regression, Poisson regression, ANOVA, log-linear models and etc.

7

NEURAL NETWORKS

Nonlinear models which represent a metaphor for the functioning of the human brain.

TOP 10 Machine Learning Algorithms

8

ASSOCIATION RULES

If/then statements that aim to uncover some relationships between unrelated data in a given database.

9

GENETIC ALGORITHMS

A family of stochastic search algorithms witch mechanism is inspired by the process of neo-Darwinian evolution.

10

LATENT DIRICHLET ALLOCATION (LDA)

A generative probabilistic model designed for collections of discrete data.

1. K-means Clustering

1. Based on the value k,
2. Initialize the k cluster centroids (many ways).
3. Cluster the n inputs by assigning them to the nearest cluster centroids.
4. Re-calculate the new k cluster centroids based on the inputs.
5. Comparing with the previous clustering centroids, if none of the n inputs changed cluster in the last iteration, exit. Otherwise go to 3.

1. P-Norm Distances

For a point (x_1, x_2, ...,x_n) and a point (y_1, y_2, ...,y_n), the **Minkowski distance** of order p (**p-norm distance**) is defined as:

1-norm distance $= \sum_{i=1}^{n} |x_i - y_i|$

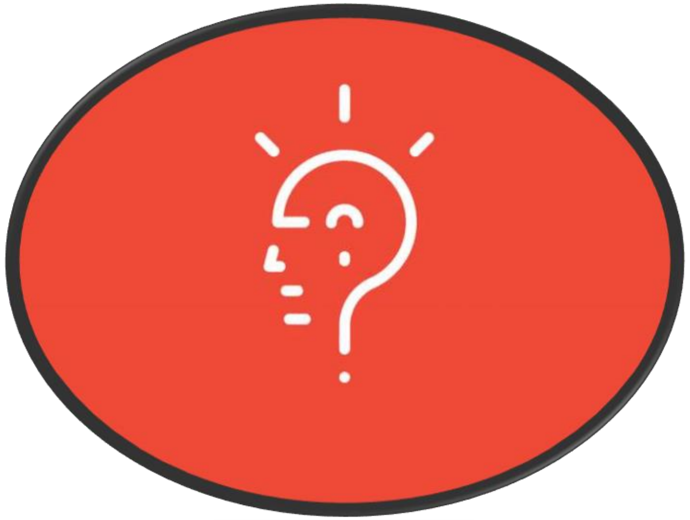
2-norm distance $= \left(\sum_{i=1}^{n} |x_i - y_i|^2\right)^{1/2}$

p-norm distance $= \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$

infinity norm distance $= \lim_{p\to\infty} \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$

$$= \max\left(|x_1 - y_1|, |x_2 - y_2|, \ldots, |x_n - y_n|\right).$$

Any Question?



Group Project Update

小组项目-调研综述进展汇报

序号	题目	成员
1	AI+斗地主	孙永康、李怀武、胡鸿飞、吴一凡、杨光、张习之、金肇轩（组长）、于佳宁
2	AI+五子棋	周贤玮、韩梓辰（组长）、赵云龙、张坤龙、夏星晨
3	High Score Gamer	易辰朗、许天淇、黄北辰（组长）、赵思源、朱佳伟、宛清源
4	AI application on diabetes	周钰奇、李仪轩、董叔文、湛掌、胡钧淇（组长）、裴鸿婧
5	AI in lung cancer	夏瑞浩、李悦明、龚颖璇、吴云潇潇（组长）、姜欣瑜、王英豪
6	基于MRI图像的阿尔茨海默症分类	董廷臻、郑英炜（组长）、李博翱、朱嘉楠、李杨燊
7	AI Applications in Breast Cancer Imaging	林文心、翟靖蕾（组长）、孙瀛、林宝月、陈帅名、冀鹏宇
8	Applications of artificial intelligence in covid-19 patients	罗岁岁（组长）、周雅雯、肖雨馨、程旸、尹子宜
9	基于OCT图像的眼部多种疾病诊断和分析的调研	何忧、郭煜煊、朱寒旭、赵子璇（组长）、王子杰、张晓新
10	人工智能对白内障分级的算法综述	赵宇航、徐格蕾、陈星宇、祖博瀛、黄弋骞（组长）
11	句子图片的文本情感分	唐云龙、刘叶充、刘旭坤、马卓远、陈子蔚（组

序号	题目	成员
12	gesture recognition	车文心、张静远、张骥霄（组长）、杜鹏辉
13	AI in Lab	孙含曦、于松琦、罗西（组长）、唐家豪、孙杰欣
14	人脸识别算法的发展与应用	易翔（组长）、陈俊滔、罗景南、胡泰玮、文颖潼、吴杰翰
15	人工智能在无障碍设施领域中的使用调查	马子晗（组长）、陈沐尧、林小璐、任艺伟、王增义
16	identification of handwriting elements	刘通、谈思序、赵伯航、张皓淇
17	AI虚拟主播制作计划	王标、张倚凡（组长）、李康欣、何泽安、曾宇祺、 Zhang Kenneth
18	人工智能技术在个性化推荐系统上的应用与研究	谭雅静、刘思岑、Ooi Yee Jing、孟宇阳、杨锦涛（组长）
19	校园巴士路线优化	王祥辰、何鸿杰、吴子彧、樊青远（组长）、方琪涵、袁通
20	给线稿上色的强大AI的算法研究	韩晗（组长）、刘思语、赵晓蕾、陈松斌
21	人工智能应用于病理分析的前景与挑战	刘宇欣、李修治（组长）、沈睿琦
22	深度学习在自动驾驶中的应用	王晓轩

规则

1. 每个小组presentation时间10分钟左右，3分钟提问时间；
2. presentation分数为20分，学生打分为15分，老师打分为5分；
3. 采用雨课堂投票打分；

课程项目汇报时间安排

小组编号	时间	主题	成员
3	十三周（上午）	High Score Gamer	易辰朗、许天淇、黄北辰（组长）、赵思源、朱佳伟、宛清源
4	十三周（上午）	AI application on diabetes	周钰奇、李仪轩、董叔文、湛掌、胡钧淇（组长）、裴鸿婧
10	十三周（上午）	人工智能对白内障分级的算法综述	赵宇航、徐格蕾、陈星宇、祖博瀛、黄弋骞（组长）
22	十三周（上午）	深度学习在自动驾驶中的应用	王晓轩
13	十四周（上午）	AI in Lab	孙含曦、于松琦、**罗西（组长）**
14	十四周（上午）	人脸识别算法的发展与应用	**易翔（组长）**、陈俊滔、罗景南、胡泰玮、文颖潼、吴杰翰
17	十四周（上午）	AI虚拟主播制作计划	王标、**张倚凡（组长）**、李康欣、何泽安、曾宇祺、Zhang Kenneth
19	十四周（上午）	校园巴士路线优化	王祥辰、何鸿杰、吴子彧、**樊青远（组长）**、方琪涵、袁通

课程项目汇报时间安排

小组编号	时间	主题	成员
1	十三周（下午）	AI+斗地主	孙永康、李怀武、胡鸿飞、吴一凡、杨光、张刁之、金肇轩（组长）、于佳宁
5	十三周（下午）	AI in lung cancer	夏瑞浩、李悦明、龚颖璇、吴云潇潇（组长）、姜欣瑜、王英豪
6	十三周（下午）	基于MRI图像的阿尔茨海默症分类	董廷臻、郑英炜（组长）、李博翱、朱嘉楠、李杨燊
7	十三周（下午）	AI Applications in Breast Cancer Imaging	林文心、翟靖蕾（组长）、孙瀛、林宝月、陈帅名、冀鹏宇
8	十三周（下午）	Applications of artificial intelligence in covid-19 patients	罗岁岁（组长）、周雅雯、肖雨馨、程旸、尹子宜
9	十三周（下午）	基于OCT图像的眼部多种疾病诊断和分析的调研	何忱、郭煜煊、朱寒旭、赵子璇（组长）、王子杰、张晓新
2	十三周（下午）	AI+五子棋	周贤玮、韩梓辰（组长）、赵云龙、张坤龙、夏星晨
11	十四周（下午）	句子图片的文本情感分析	唐云龙、刘叶充、**刘旭坤（组长）**、马卓远、陈子蔚、江欣乐、陈浩然
12	十四周（上午）	gesture recognition	车文心、张静远、**张骥霄（组长）**、杜鹏辉
15	十四周（下午）	人工智能在无障碍设施领域中的使用调查	**马子晗（组长）**、陈沐尧、林小璐、任艺伟、王增义
16	十四周（下午）	identification of handwriting elements	刘通、**谈思序（组长）**、赵伯航、张皓淇
18	十四周（下午）	人工智能技术在个性化推荐系统上的应用与研究	谭雅静、刘思岑、Ooi Yee Jing、孟宇阳、**杨锦涛（组长）**
20	十四周（下午）	给线稿上色的强大AI的算法研究	**韩晗（组长）**、刘思语、赵晓蕾、陈松斌
21	十四周（下午）	人工智能在皮肤癌诊断领域的可能性探索	刘宇欣、**李修治（组长）**、沈睿琦

Any Question?