SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

01.112 Machine Learning, Spring 2018
Homework 2

Due 14 Feb 2018, 5pm

This homework will be graded by Allan Jie

Question 1. Assume a function $f : R^n \rightarrow R$ is continuous. As discussed in class, if it is a convex function, then it satisfies the following property:

$$\textit{Property A:} \qquad f\left(\frac{x_1 + x_2}{2}\right) \leq \frac{f(x_1) + f(x_2)}{2}$$

for any $x_1, x_2 \in R^n$.

Show that the following statements are true (**with formal proofs**):

(a) If two functions $f(x)$ and $g(x)$ both satisfy Property A, then the following function also satisfies Property A:

$$h(x) = f(x) + g(x)$$

(5 points)

(b) If two functions $f(x)$ and $g(x)$ both satisfy Property A, then the following function also satisfies Property A:

$$h(x) = \max(f(x), g(x))$$

(5 points)

Question 2. Anticoagulants are drugs that reduce blood clotting and are used to prevent a wide variety of medical conditions such as deep vein thrombosis, pulmonary embolism, myocardial infarction and ischemic stroke. Warfarin is the most widely used oral anticoagulant worldwide (with more than 30 million prescriptions in the United States alone in 2004). The correct dose of warfarin is hard to determine because it can vary by as much as a factor of 10 among patients, and the consequences of taking a wrong dose can be lethal. In this problem, you shall **implement stochastic gradient descent to learn a linear regression model** to predict the correct dose of warfarin. You are provided with three files: `train_warfarin.csv` (training data), `validation_warfarin.csv` (training data that we have withheld for you to tune your algorithm parameters, if necessary), and `test_warfarin.csv` (test data). The format of the csv files are given in Annex A.

(a) Train your linear regression model using stochastic gradient descent on `train_warfarin.csv`. Run 10000 iterations of stochastic gradient descent. Save the weights of your model after every 100 iterations of stochastic gradient descent. Plot the mean squared error (i.e., $\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \theta \cdot \mathbf{x}^{(i)} - \theta_0)^2$ where $(\mathbf{x}^{(i)}, y^{(i)})$ is an example and n is the number of examples) for each set of weights that are saved (error values on the vertical axis; iterations on the horizontal axis). **On the same graph**, draw one plot each for the mean squared errors on the training set, validation set, and test set (clearly label the three plots). We suggest you try a fixed learning rate of 0.1. If you can get better performance with another learning rate, please do so. Provide crystal clear instructions along with the source code on how to execute it. *(8 points)*

Hints: 1) the stochastic gradient descent algorithm for linear regression presented in the notes/class does not involve θ_0. You will need to figure out what should be the update equation for θ_0, 2) for this question we do not ask you to consider the regularization term. You are, however, free to investigate the effectiveness of the regularization on your own - no submission of such results is required.

(b) Explain in English how could you use the validation set to select the model (with the parameters θ, θ_0) to use on the test set? *(2 points)*

(Note that this is a real-world application published in *The New England Journal of Medicine*[1]. We are using the same attributes as the paper, but the data is artificially generated. The licensing of the paper's data makes it tricky to distribute in class, but if you want to experiment with it, you may download it for your own purposes from www.pharmgkb.org. Make sure to read the license!)

Question 3. In clustering, Euclidean distance is not the only way to measure the distance between two points/vectors. l_p norms is a family of distance measures that are parameterized by $p \geq 1$. The l_p norm of a vector is:

$$\|x\|_p = \left(\sum_j |x_j|^p\right)^{\frac{1}{p}}.$$

Euclidean distance is the l_2 norm of the vector difference between two points, i.e.,

$$\|x - y\|_2 = \left(\sum_j |x_j - y_j|^2\right)^{\frac{1}{2}}.$$

The Manhattan distance is the l_1 norm of the vector difference between two points, i.e.,

$$\|x - y\|_1 = \sum_j |x_j - y_j|.$$

[1]Estimation of the Warfarin Dose with Clinical and Pharmacogenetic Data (http://www.nejm.org/doi/full/10.1056/NEJMoa0809329#t=articleBackground)

The l_∞ distance is the maximum absolute element in the vector difference between two points, i.e.,

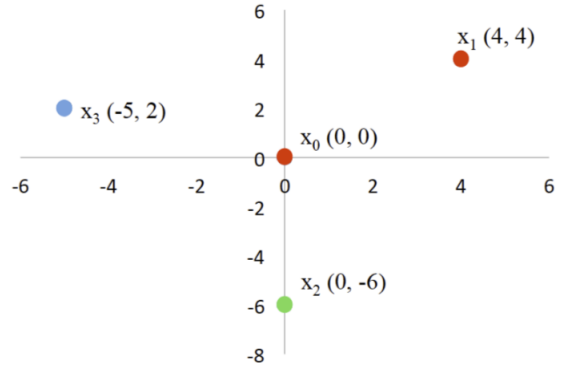
$$\|x - y\|_\infty = \max_j |x_j - y_j|.$$

(Think about why this is so.)

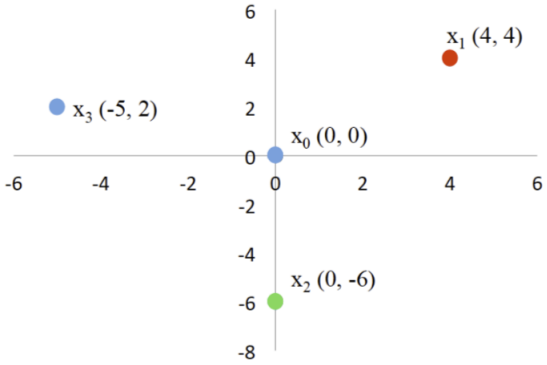
(a) Consider a set of points $X = (0.6, 0.8), (0.8, 0.6), (-0.8, 0.6)$. Compute the value of z that minimizes $\sum_{x \in X} d(x, z)$ when $d(x, z)$ is defined as follows respectively: 1) the Euclidean distance between x and z, and 2) the Manhattan distance between x and z. *(5 points)*

(b) The following figures (points in the same cluster have the same color) are produced by the k-medoids algorithm for $k = 3$ clusters using l_1 , l_2 , and l_∞ distance measures. Indicate which distance measure is used for each figure. *(5 points)*

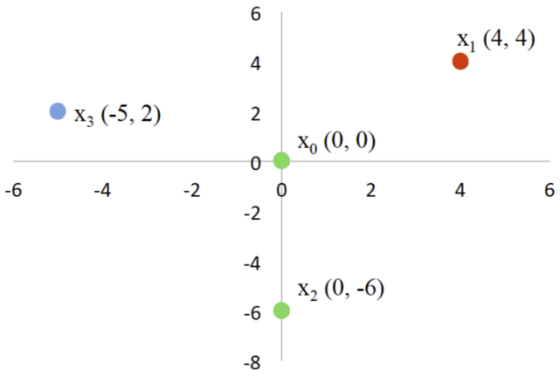
A.



B.



C.



Some of you asked how to approach Q3(a) when the distance is Euclidean distance. Some hints: the point that minimizes such a total distance to the three points is known as the Fermat point associated with the triangle. So one way is to construct some functions associated with some circles and find the intersection points analytically. You may even use a computer program to help you - we only need a solution so any (creative) method is Okay here.

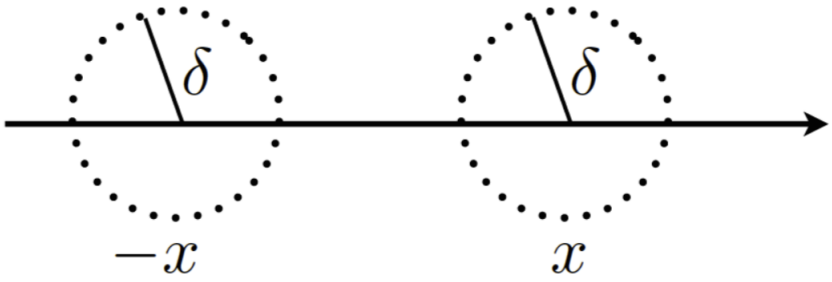
We are Okay to accept your solution even if you use squared Euclidean distance for this question this time, but it's strongly encouraged to think through the solution for the Euclidean distance case - it's interesting to see how the solutions would be different.

Question 4. Each iteration of the k-means algorithm consists of two steps: assigning points to centroids, and updating the centroids based on the points assigned to them. Assume that the number of clusters $k = 2$.

(a) If the centroids are initialized to be the means of two *well-separated* clusters, will the centroids change after the first iteration? (A yes/no answer suffices.) *(1 point)*

(b) If the centroids are initialized by setting each to a random point from each of the two well-separated clusters, how many iterations does it take for k-means to converge? Explain your answer. *(4 point)*

(c) In the figure below, we have two spherical (circle-like) clusters of radius δ that are centered at locations $-x$ and x. For what values of x would the k-means algorithm fail to find the centers of the two clusters regardless of the initialization? Explain your answer. *(5 points)*



Annex A A row in each csv file for the warfarin problem contains the output and attributes for a patient in the following order.

1. Warfarin Dose (mg/week; output being predicted)
2. Normalized age in years
3. Normalized height in cm
4. Normalized weight in kg
5. VKORC1 genotype A/A (1: present; 0: absent)
6. VKORC1 genotype A/G (1: present; 0: absent)
7. VKORC1 genotype G/G (1: present; 0: absent)
8. VKORC1 genotype unknown (1: unknown; 0: known)
9. CYP2C9 genotype *1/*1 (1: present; 0: absent)

10. CYP2C9 genotype *1/*2 (1: present; 0: absent)
11. CYP2C9 genotype *1/*3 (1: present; 0: absent)
12. CYP2C9 genotype *2/*2 (1: present; 0: absent)
13. CYP2C9 genotype *2/*3 (1: present; 0: absent)
14. CYP2C9 genotype *3/*3 (1: present; 0: absent)
15. CYP2C9 genotype unknown (1: unknown; 0: known)
16. Race Asian (1: true; 0: false)
17. Race Black (1: true; 0: false)
18. Race White (1: true; 0: false)
19. Race Unknown (1: true; 0: false)
20. Taking Enzyme Inducer (1: Yes; 0: No)
21. Taking Amiodarone (1: Yes; 0: No)

Exactly one of the VKORC1 genotypes attributes is 1, all others are 0. Likewise for the CYP2C9 genotype attributes, and race attributes.

affine function : $y = Ax + c$
(does not need to fix ē origin)
(unlike linear functions)

Question 1. Assume a function $f : R^n \to R$ is continuous. As discussed in class, if it is a convex function, then it satisfies the following property:

$$\textit{Property A:} \quad f\left(\frac{x_1 + x_2}{2}\right) \leq \frac{f(x_1) + f(x_2)}{2}$$

for any $x_1, x_2 \in R^n$.

Show that the following statements are true (**with formal proofs**):

(a) If two functions $f(x)$ and $g(x)$ both satisfy Property A, then the following function also satisfies Property A:

$$h(x) = f(x) + g(x)$$

(5 points)

(b) If two functions $f(x)$ and $g(x)$ both satisfy Property A, then the following function also satisfies Property A:

$$h(x) = \max(f(x), g(x))$$

(5 points)

(a)

$d = v - u$ (1)

$[u, v] = \{ u + \lambda d : \lambda \in [0,1] \}$ (2)

Sub (1) into (2):

$= \{ u + \lambda(v-u) : \lambda \in [0,1] \}$

$= \{ (1-\lambda)u + \lambda v : \lambda \in [0,1] \}$

$C \in \mathbb{R}^n$ is convex if $\forall x, y \in \mathbb{R}$, $[x, y] \subseteq C$

$d = x_2 - x_1$ (1)

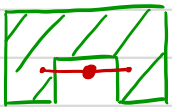
$[x_1, x_2] = \{ x_1 + \lambda x_2 : \lambda \in [0,1] \}$ (2)

Sub (1) into (2)

$= \{ x_1 + \lambda(x_2 - x_1) : \lambda \in [0,1] \}$

$= \{ (1-\lambda)x_1 + \lambda x_2 : \lambda \in [0,1] \}$

$C \in \mathbb{R}^n$ is convex if $\forall x, y \in \mathbb{R}$, $[x, y] \subseteq C$



not convex

convex

Take $x_1, x_2 \in C \quad \exists \lambda_x, \lambda_y \in [0,1]$ st $\quad f(x) = (1-\lambda_x)x_1 + \lambda_x x_2$

$g(x) = (1-\lambda_y)x_1 + \lambda_y x_2$

Proof

Take $h(x) \in [f(x), g(x)]$. Then $\exists \lambda \in [0,1]$ st $\quad h(x) = (1-\lambda)x_1 + \lambda x_2$

$h(x) = \alpha[f(x)] + \beta[g(x)]$, $\alpha, \beta \geq 0$

$= \alpha(1-\lambda_x)x_1 + \alpha\lambda_x x_2 + \beta(1-\lambda_y)x_1 + \beta\lambda_y x_2$, $\alpha + \beta = 1$

$= (\alpha+\beta)x_1 - (\alpha\lambda_x + \beta\lambda_y)x_1 + (\alpha\lambda_x + \beta\lambda_y)x_2$, $\gamma = \alpha\lambda_x + \beta\lambda_y$

$= 1 \cdot x_1 - \gamma x_1 + \gamma x_2$

$= \underbrace{(1-\gamma)}_{\text{non-ve}} x_1 + \underbrace{\gamma}_{\text{non-ve}} x_2 \Rightarrow h(x) \in C \rightarrow [x_1, x_2] \subseteq C$

→ $h(x)$ is also convex if $f(x)$ and $g(x)$ are both convex.

$h(x)$ in line segment b/w x_1 and x_2 and hence in C where C is convex.

↓ Aka intersection of 2 convex sets.

(b) $h((1-\lambda)x_1 + \lambda x_2) = \max\left(f((1-\lambda_x)x_1 + \lambda_x x_2), g((1-\lambda_y)x_1 + \lambda_y x_2)\right)$

$\leq \max\left((1-\lambda_x)f(x_1), \lambda_x f(x_2)\right) + \max\left((1-\lambda_y)f(x_1), \lambda_y f(x_2)\right)$

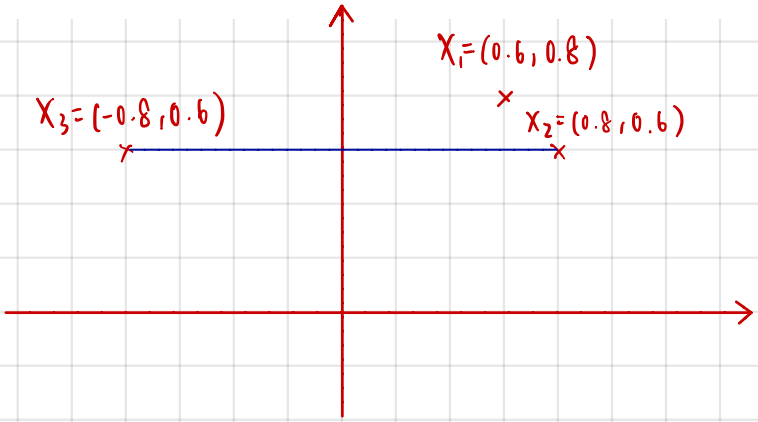
$\leq (1-\lambda)x_1 + \lambda x_2$

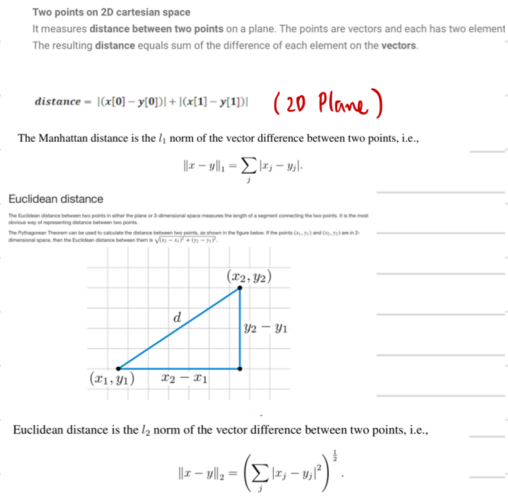
2(a) Code, Input & Instructions Attached

2(b) (b) Explain in English how could you use the validation set to select the model (with the parameters θ, θ_0) to use on the test set? *(2 points)*

As shown in 2(a), mean squared error would decrease until ≈ 5500 iteration and start to increase thereafter. This means that the error increases after that number of iterations. This means that is an ideal number of iterations for tuning the parameters when setting the number of iterations to tune the weights based on the training set.

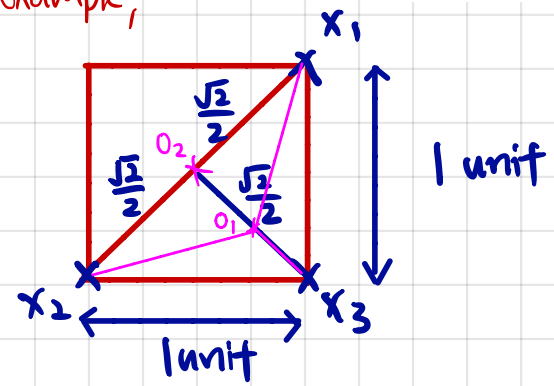
(a) Consider a set of points $X = (0.6, 0.8), (0.8, 0.6), (-0.8, 0.6)$. Compute the value of z that minimizes $\sum_{x \in X} d(x, z)$ when $d(x, z)$ is defined as follows respectively: 1) the Euclidean distance between x and z, and 2) the Manhattan distance between x and z. *(5 points)*





3(a)

Using a 1 x 1 unit as an example,



Using K-Means Algo,

From O_1 to X_2, Euclidean distance = $\sqrt{0.75^2 + 0.25^2} \approx 0.7906$

O_1 to X_1, Euclidean distance = " ≈ 0.7906

O_1 to X_3, Euclidean distance = $\sqrt{0.25^2 + 0.25^2} \approx 0.3536$

Total Euclidean distance = 1.9348

From O_1 to X_2, Manhattan distance = $0.75 + 0.25 = 1$

O_1 to X_1, Manhattan distance = $0.75 + 0.25 = 1$

O_1 to X_3, Manhattan distance = $0.25 + 0.25 = 0.5$

Total Manhattan distance = $1 + 1 + 0.5 = 2.5$

From O_2 to X_1/X_2/X_3, Euclidean distance = $\sqrt{2}/2 \approx 0.707$

Total Euclidean distance $\approx$ 2.121

From O_2 to X_2, Manhattan distance = $0.707 + 0.5 = 1.207$

O_2 to X_1, Manhattan distance = " $= 1.207$

O_2 to X_3, Manhattan distance = " $= 1.207$

Total Manhattan distance = 3.621

Using K-Mediods Algo, X_3 as representative

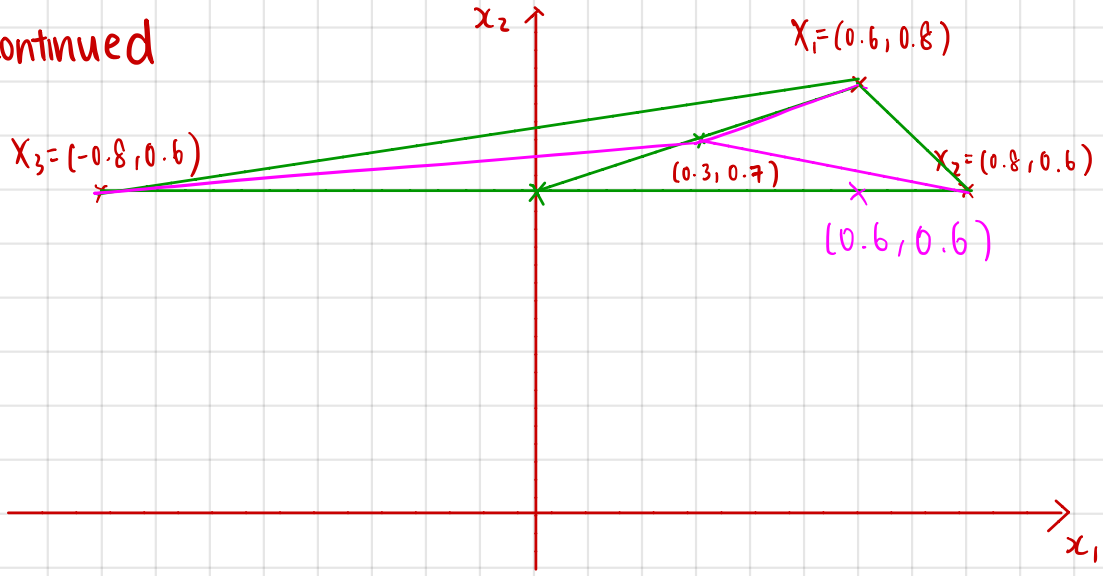
→ Total Euclidean distance = $1 + 1 = 2$

→ Total Manhattan distance = $1 + 1 = 2$

Conclusion: K-means Algo produce shortest Euclidean Distance compared to K-Mediods Algo

cont. → 3a.

3a. Continued



1. Centre Coordinate for Shortest Euclidean Distance

$$= \left(\frac{\left(\frac{0.8+(-0.8)}{2}\right) + 0.6}{2} , \frac{\frac{0.6+(0.6)}{2} + 0.8}{2} \right) = (0.3, 0.7)$$

Euclidean Distance

$$= \sqrt{(0.6-0.3)^2 + (0.8-0.7)^2} + \sqrt{(0.8-0.3)^2 + (0.6-0.7)^2} + \sqrt{(-0.8-0.3)^2 + (0.6-0.7)^2}$$

$$= 1.931$$

2. For Manhattan Distance, if use X_1 as representative

Distance from X_3 to X_1 $= |-0.8-0.6| + |0.6-0.8| = 1.6$

X_2 to X_1 $= |0.8-0.6| + |0.6-0.8| = 0.4$

Total Manhattan Distance $= 2.0$

For Manhattan Distance, if use X_2 as representative

Distance from X_3 to X_2 $= |-0.8-0.8| + |0.6-0.6| = 1.6$

X_1 to X_2 $= |0.8-0.6| + |0.6-0.8| = 0.4$

Total Manhattan Distance $= 2.0$

For Manhattan Distance, choose medium x_1, x_2 coordinates set $C_2(0.6, 0.6)$ as representative

x_1	x_2
-0.8	0.6
0.6	0.6
0.8	0.8

Distance from X_3 to C_2 $= |-0.8-0.6| + |0.6-0.6| = 1.4$

X_2 to C_2 $= |0.8-0.6| + |0.6-0.6| = 0.2$

X_1 to C_2 $= |0.6-0.6| + |0.8-0.6| = 0.2$

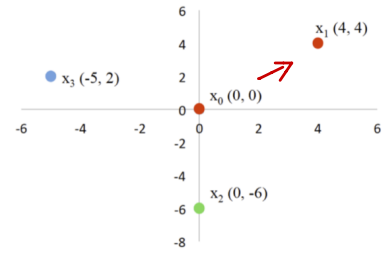
Total Manhattan Distance $= 1.8$

In Conclusion, $d(x,z) = (0.3, 0.4)$ for shortest Euclidean Distance

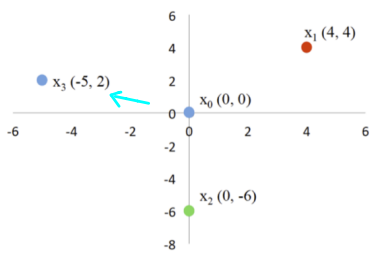
$d(x,z) = (0.6, 0.6)$ for shortest Manhattan Distance.

(b) The following figures (points in the same cluster have the same color) are produced by the k-medoids algorithm for $k = 3$ clusters using l_1 , l_2 , and l_∞ distance measures. Indicate which distance measure is used for each figure. *(5 points)*

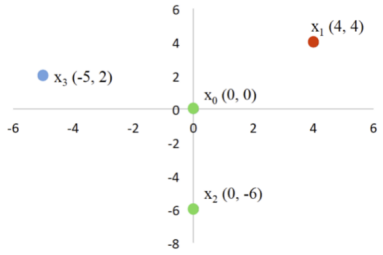
A.



B.



C.



Two points on 2D cartesian space

It measures **distance between two points** on a plane. The points are vectors and each has two elements. The resulting **distance** equals sum of the difference of each element on the **vectors**.

$$distance = |(x[0] - y[0])| + |(x[1] - y[1])|$$ (2D Plane)

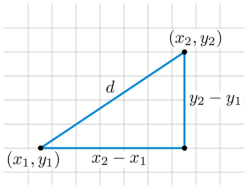
The Manhattan distance is the l_1 norm of the vector difference between two points, i.e.,

$$\|x - y\|_1 = \sum_j |x_j - y_j|.$$

Euclidean distance

The Euclidean distance between two points in either the plane or 3-dimensional space measures the length of a segment connecting the two points. It is the most obvious way of representing distance between two points.

The Pythagorean Theorem can be used to calculate the distance between two points, as shown in the figure below. If the points (x_1, y_1) and (x_2, y_2) are in 2-dimensional space, then the Euclidean distance between them is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.



Euclidean distance is the l_2 norm of the vector difference between two points, i.e.,

$$\|x - y\|_2 = \left(\sum_j |x_j - y_j|^2\right)^{\frac{1}{2}}.$$

The l_∞ distance is the maximum absolute element in the vector difference between two points, i.e.,

$$\|x - y\|_\infty = \max_j |x_j - y_j|.$$

3(b)

Distance Measure

For x_3 , Manhattan Dist = 5 + 2 = 7

Euclidean Dist = $\sqrt{5^2 + 2^2} = \sqrt{25+4} = \sqrt{29} \approx 5.39$

l_∞ Dist = max (x_1 , x_2) = max (5, 2) = 5

B.

For x_2 , Manhattan Dist = 0 + 6 = 6

Euclidean Dist = $\sqrt{0^2 + 6^2} = \sqrt{6^2} = \sqrt{36} = 6$

l_∞ Dist = max (x_1 , x_2) = max (0, 6) = 6

A.

For x_1 , Manhattan Dist = 4 + 4 = 8

Euclidean Dist = $\sqrt{4^2 + 4^2} = \sqrt{16+16} = \sqrt{32} \approx 5.66$

l_∞ Dist = max (x_1 , x_2) = max (4, 4) = 4

C.

Rationale: Find the shortest distance from point to origin using the distance method. ie Euclidean Distance, $x_1 = 5.66$, $x_2 = 6$, $x_3 = 5.39$

↑ Shortest Distance Hence B is labelled blue with origin.

In conclusion Fig A uses l_1

Fig B uses l_2

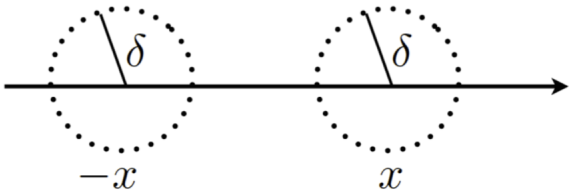
Fig C uses l_∞

Question 4. Each iteration of the k-means algorithm consists of two steps: assigning points to centroids, and updating the centroids based on the points assigned to them. Assume that the number of clusters $k = 2$.

(a) If the centroids are initialized to be the means of two *well-separated* clusters, will the centroids change after the first iteration? (A yes/no answer suffices.) *(1 point)*

(b) If the centroids are initialized by setting each to a random point from each of the two well-separated clusters, how many iterations does it take for k-means to converge? Explain your answer. *(4 point)*

(c) In the figure below, we have two spherical (circle-like) clusters of radius δ that are centered at locations $-x$ and x. For what values of x would the k-means algorithm fail to find the centers of the two clusters regardless of the initialization? Explain your answer. *(5 points)*



(A) No

(B) Since the clusters are well-separated, the initialization will not change assignment of points to centroids. The centroids become cluster means after the first iteration.

(C) Given radius to be labelled delta units, if |x| < delta and there lies points between both intersected circles, then the k-means algorithm would fail to find the centers of the two clusters regardless of the initialization. This is because the two circle-like clusters that overlap causes the points within the overlap to be unable to be assigned just one membership. If x = delta, then the point that lie on both circumference would also face the same problem causing the k-means algorithm to fail.