Introduction to Machine Learning Course

Short HW2: Classification: Introduction

Submitted individually by Sunday, 07.07, at 23:59.

You may answer in Hebrew or English and write on a computer or by hand (but be clear).

Please submit a PDF file named as your ID number, e.g., 123456789.pdf.

Bonus (maximal grade is 100): Writing on a computer (using LyX/LaTeX, Word + Equation tool, etc.) = 2 pts.

Decision trees

1. This question demonstrates that ID3 does not always yield an optimal tree. Consider the following dataset:

ID	**Feature A**	**Feature B**	**Feature C**	**Target**
1	1	1	1	+1
2	1	0	0	+1
3	1	1	0	-1
4	0	0	1	-1

1.1. Run ID3 manually up to depth 2. In the case of a leaf with non-zero entropy predict the majority label in that leaf. In case of a tie, predict “-1”. The nodes and edges in the decision tree should be labeled appropriately. You can use your preferred graphics software to draw the tree, or you can hand draw it (neatly). Calculate the training error on the given dataset and state which tuples were misclassified.

Answer:

I will use the following formla: $\mathrm{IG}(v,a) = H(v) - \frac{|v_{a=1}|}{|v|}H(v_{a=1}) - \frac{|v_{a=0}|}{|v|}H(v_{a=0})$

First we need to find the feature with the largest information gain:

Attribute	$\frac{\lvert v_{a=1}\rvert}{\lvert v\rvert}$	$\frac{\lvert v_{a=0}\rvert}{\lvert v\rvert}$	$H(v_{a=1})$	$H(v_{a=0})$	$\mathrm{IG}(v,a) - H(v)$
Feature A	¾	¼	$H(\frac{2}{3})$	$H(1) = 0$	$-\frac{3}{4}H\left(\frac{2}{3}\right) \approx -0.477$
Feature B	½	½	$H(\frac{1}{2})$	$H(\frac{1}{2})$	$-H\left(\frac{1}{2}\right) = -1$
Feature C	½	½	$H(\frac{1}{2})$	$H(\frac{1}{2})$	$-H\left(\frac{1}{2}\right) = -1$

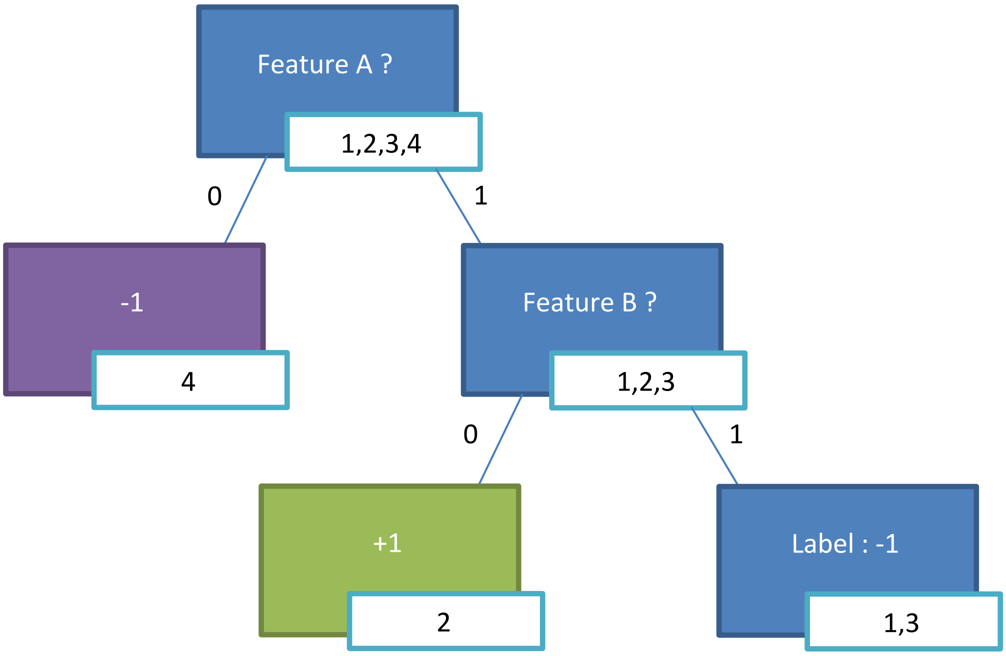
We have that the largest information gain is for feature A so we choose this feature for the first split.

Next, we’ll find the next best information gain feature on the left un-classified data.

Attribute	$\frac{\lvert v_{a=1}\rvert}{\lvert v\rvert}$	$\frac{\lvert v_{a=0}\rvert}{\lvert v\rvert}$	$H(v_{a=1})$	$H(v_{a=0})$	$\mathrm{IG}(v,a) - H(v)$
Feature B	2/3	1/3	$H(\frac{1}{2})$	$H(1) = 0$	$-\frac{2}{3}H(\frac{1}{2})$
Feature C	1/3	2/3	$H(1) = 0$	$H(\frac{1}{2})$	$-\frac{2}{3}H(\frac{1}{2})$

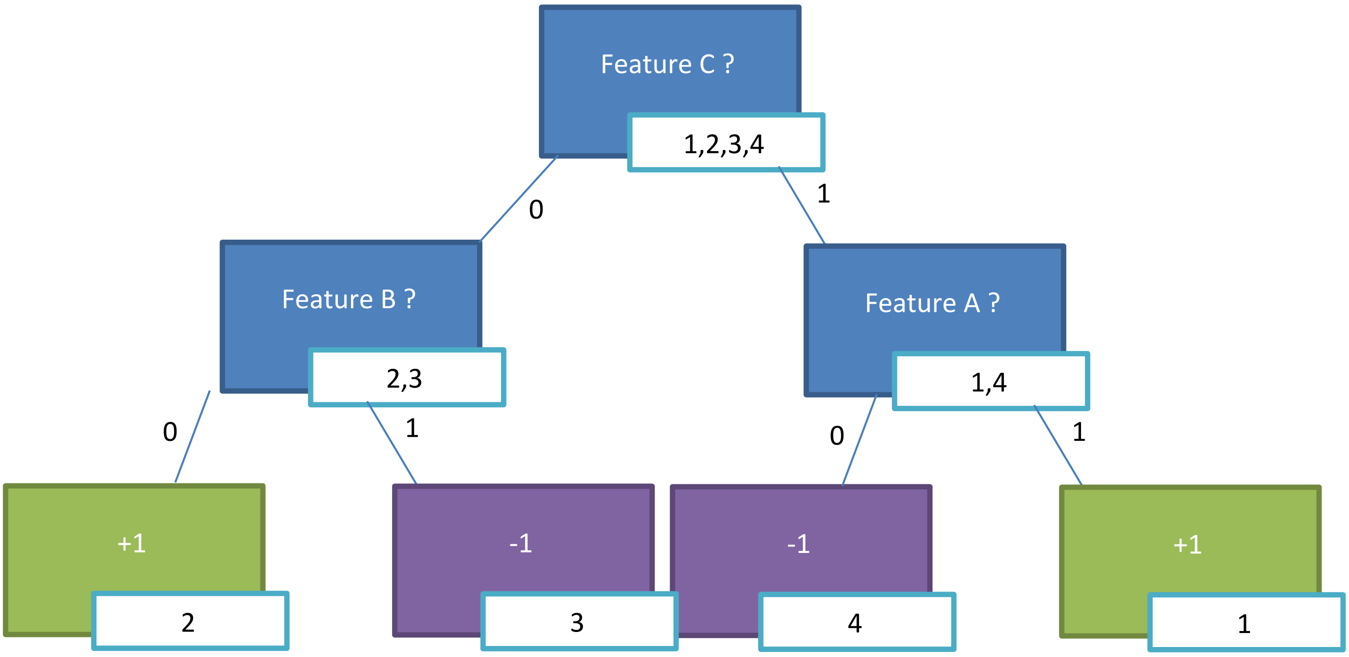
Since feature B and feature C has the same information gain rate, choosing either of them will lead us to the same result, hence consinder ramdomely choosing feature B as our 2nd feature.

All together the final ID3 tree would look as the following:



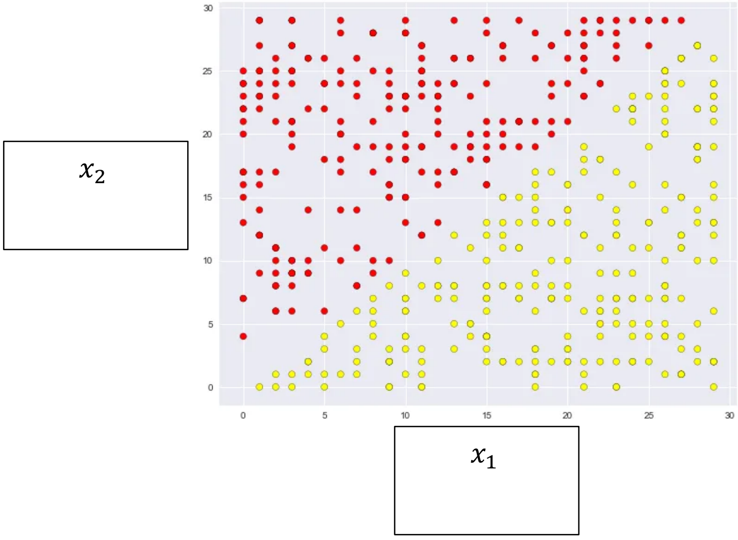
Training error: out of 4 samples, the misclassified tupple is sample number 1 with target +1 and classification -1. Other then that all tupples were classified correctly. So training error is 25%.

1.2. Find (manually) a decision tree of depth 2 that fits the training data perfectly.



Decision trees strike again

2. Given before you a dataset that consists of samples with two features $x_1, x_2 \in \mathbb{R}$ and a label $y \in \{-1,1\}$

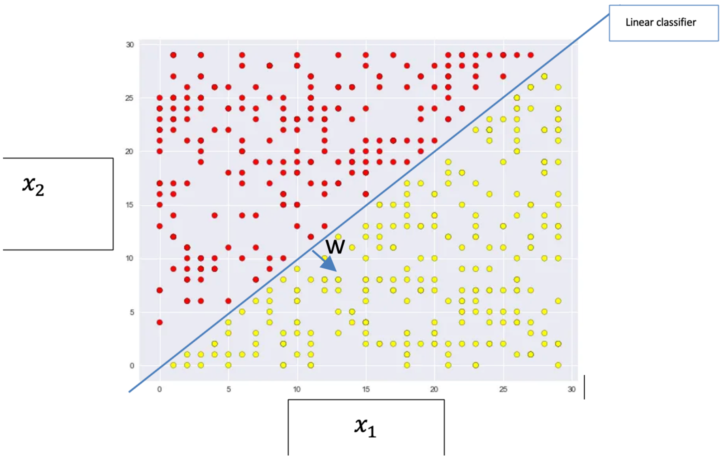


2.1. Can you identify a linear classifier $h(x_1, x_2) = sign(w^T x + b)$ that achieves a perfect accuracy on the dataset? You may provide a drawing to emphasize your point.

Answer:

The linear classifier is $h(x_1, x_2) = sign(w^T x)$ s.t. $h\ is\ defined\ by\ w = (1, -1)$. Following that:

$$h(x_1, x_2) = sign((1,-1)^T x) = sign(x_1 - x_2) = \begin{cases} 1(yellow), x_1 - x_2 > 0 \\ -1\ (red), x_1 - x_2 < \ 0 \end{cases} = \begin{cases} 1(yellow), x_1 > x_2 \\ -1\ (red), x_2 > x_1 \end{cases}$$



2.2. Can you identify a decision tree of depth 1 that achieves a perfect accuracy on the dataset? You may provide a drawing of the decision boundary to emphasize your point.

Answer:

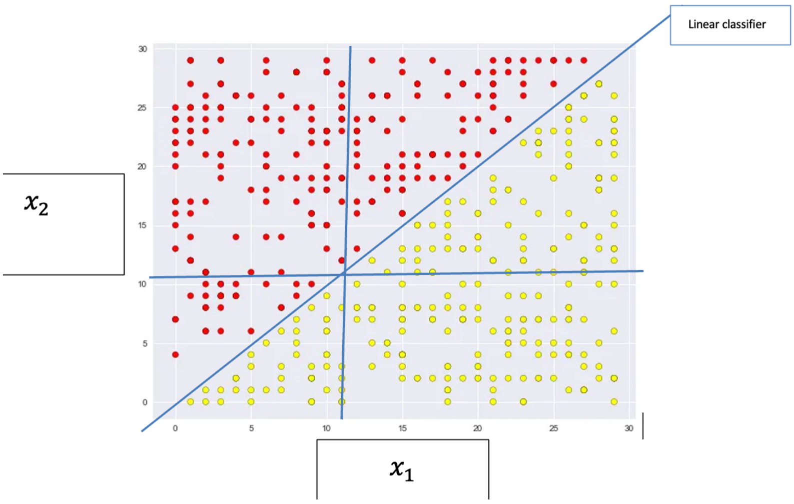
No, there isn't a decision tree of depth 1 that achives perfect accuracy on this dataset. In our course the decisions on the nodes are threshholds for one spacific feature. For any threshold niegher for x1 nor x2, the labels are not perfactly sepetrable (because any decision boundry is parallel to the axises although diaganal isn't).

2.3. Can you identify a decision tree of depth 2 that achieves a perfect accuracy on the dataset? You may provide a drawing of the decision boundary to emphasize your point.

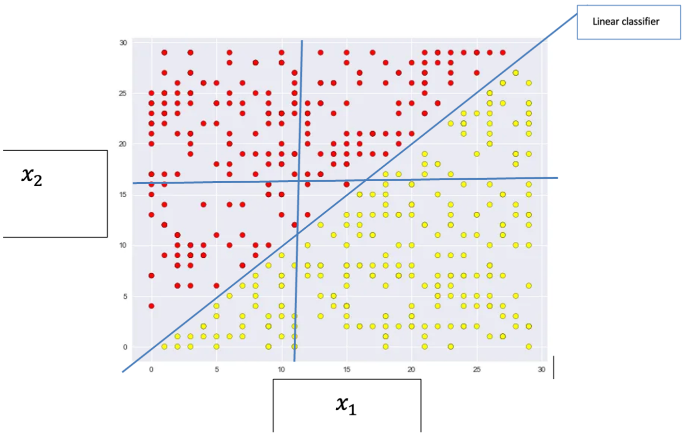
Answer:

No, there isn't a decision tree of depth 2 that achives perfect accuracy on this dataset. Notice that two decision yield 4 ractangles as decision areas. Since the accurate seperator is a diaganal, the geometric seperation is shaped by a 3-corner although it is not possible to make that fit using 4 ractangles. Assume that there were 4 ractangles to classifies the data perfectly. Then, spliting into cases:

1- **The 4 vertexes meet on the diaganal:** then, there are two ractangles with mixed labals which will mistake on one of the halfs, in each.



2- **The 4 vertexes meet above (simirly below) the diaganal:** in this case the model will mistake over some area in 3 ractangles.



Hence there isn't a decision tree of depth 2 to make accurate labelling to this model.

2.4. Will your answers change if we rotate the dataset by 45 degrees clockwise? Specifically, if each (x_1, x_2) in the dataset is multiplied by a 45 degrees rotation matrix $R \in \mathbb{R}^{2\times 2}$, while the labels remain unchanged?

Answer:

In case of a CW 45 degrees rotation, even one decision is enough to make accurate seperation. This way, asking the question $x_2 \overbrace{<}^{?} 0$ will make the following decision regions:

$$x_2 \overset{?}{<} 0 = \begin{cases} yellow, & if\ yes \\ red, & if\ no \end{cases}$$

Then of course 2 decision tree will be accurate and fit well.

3. We now wish to prove that the information gain, i.e., $\mathrm{IG}(v,a) = H(v) - \frac{|v_{a=T}|}{|v|}H(v_{a=T}) - \frac{|v_{a=F}|}{|v|}H(v_{a=F})$, is non-negative for any set of examples v and a binary attribute a.

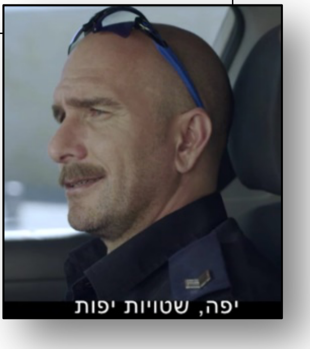
Recall our definition of $v_{a=T} \triangleq \{(x_i, y_i) \in v \mid x_i[a] = T\}$ and $v_{a=F} \triangleq \{(x_i, y_i) \in v \mid x_i[a] = F\}$.

See the following dialogue, where we (really) asked ChatGPT to prove this (we shortened it a little):

> **Question**: could you prove that the information gain in id3 cannot be negative?
>
> **Answer**: Yes, I can prove this using mathematical reasoning.
>
> In the ID3 algorithm, the information gain (IG) of an attribute A with respect to a dataset `D` is defined as: `IG(A) = Entropy(D) - Sum([|Dv| / |D| * Entropy(Dv)])`
> where `Entropy(D)` is the entropy of the original dataset `D`, `|Dv|` is the number of instances in the subset of `D` that have a specific value `v` for the attribute `A`, and `Entropy(Dv)` is the entropy of that subset.
>
> Entropy is a measure of the randomness or unpredictability of the target variable, and it is calculated as: `Entropy(D) = - Sum([p(c) * log2(p(c))])`
> where `p(c)` is the proportion of instances in `D` that belong to class `c`.
>
> Now, let's assume that the IG for a given attribute `A` is negative, i.e., `IG(A) < 0`.
>
> Then we have: `Entropy(D) - Sum([|Dv| / |D| * Entropy(Dv)]) < 0`
> Rearranging the terms, we get: `Entropy(D) < Sum([|Dv| / |D| * Entropy(Dv)])`
>
> Since both `Entropy(D)` and `Entropy(Dv)` are non-negative, the right-hand side of the inequality is also non-negative.
> Therefore, we have: `Entropy(D) >= 0.`
>
> This contradicts the assumption that `IG(A) < 0`, because
> `Entropy(D) - Sum([|Dv| / |D| * Entropy(Dv)])` is the value of `IG(A)`.
>
> Thus, we can conclude that the information gain in the ID3 algorithm is always non-negative and cannot be negative.

3.1. Attached is a quote from “ארץ נהדרת” that best describes the answer above.

Briefly explain what the problem with the above “proof” is.



Answer:

The contradiction is incorrect. The only thing it got is lower boundry of zero for This inequallity 0<= `Entropy(D) < Sum([|Dv| / |D| * Entropy(Dv)]).`

But no-where chatGPT proved the proportion between `Entropy(D)` and `Sum([|Dv| / |D| * Entropy(Dv)]).` In order to make the contradiction chatGPT aimed to do, it needed to show that `Entropy(D) >= Sum([|Dv| / |D| * Entropy(Dv)])` , that way IG(A) >= 0, in contradiction.

3.2. Let us really prove this.

Let v be an arbitrary set of examples and a be a binary attribute (feature).

Assume the classes are also binary.

The entropy is thus: $H(v) = H\left(\frac{1}{|v|}\{(x,y) \in v \mid y = 1\}\right) \triangleq H(p_v) = -p_v \log_2 p_v - (1 - p_v)\log_2(1 - p_v)$.

(Notice that like in the tutorial, we use H with a set v or with a probability p_v interchangeably.)

Prove: $\mathrm{IG}(v,a) \geq 0$.

You can use the following property without proving it.

Property: It holds $\forall \beta_1, \beta_2, \alpha \in [0,1]$ that: $\quad -\alpha H(\beta_1) - (1-\alpha)H(\beta_2) \geq -H(\alpha\beta_1 + (1-\alpha)\beta_2)$.

Answer:

Given v an arbitrary set of examples and a be a binary attribute. Notice that $|v_{a=T}| = |v| - |v_{a=F}|$ hence $\frac{|v_{a=T}|}{|v|} = \frac{||v| - |v_{a=F}||}{|v|} = 1 - \frac{|v_{a=F}|}{|v|}$. Also notice that $\frac{|v_{a=T}|}{|v|} \in [0,1]$.

All together:

$$-\frac{|v_{a=T}|}{|v|}H(v_{a=T}) - \frac{|v_{a=F}|}{|v|}H(v_{a=F})$$

$$= -\left(1 - \frac{|v_{a=F}|}{|v|}\right)H(v_{a=T}) - \frac{|v_{a=F}|}{|v|}H(v_{a=F}) \underbrace{\geq}_{\mathit{using\ the\ given\ propety}}$$

$$-H\left(\frac{|v_{a=F}|}{|v|}H(v_{a=F}) + \left(1 - \frac{|v_{a=F}|}{|v|}\right)H(v_{a=T})\right)$$

$$= -H\left(\frac{|v_{a=F}|}{|v|}H(v_{a=F}) + \frac{|v_{a=T}|}{|v|}H(v_{a=T})\right) = -H(v)$$

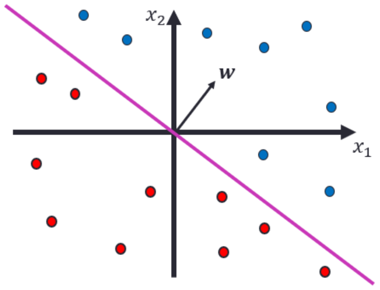
This gives us :

$$\frac{|v_{a=T}|}{|v|}H(v_{a=T}) + \frac{|v_{a=F}|}{|v|}H(v_{a=F}) \leq H(v)$$

Thus,

$$\mathrm{IG}(v,a) = H(v) - \left(\frac{|v_{a=T}|}{|v|}H(v_{a=T}) + \frac{|v_{a=F}|}{|v|}H(v_{a=F})\right) \geq H(v) - H(v) = 0$$

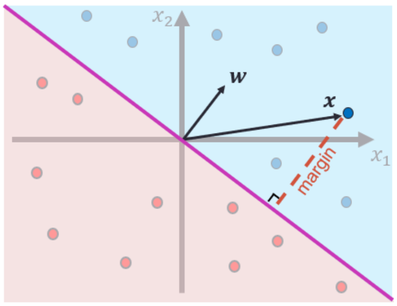
Linear classification



In tutorial 02, we illustrated algebraic and geometric properties of linear classifiers. Specifically, we explained that a homogeneous linear classifier $h: \mathbb{R}^d \to \mathbb{R}$ is parameterized by a normal vector $\boldsymbol{w} \in \mathbb{R}^d$, such that:

$$h(\boldsymbol{x}) = +1 \iff \boldsymbol{w}^\top \boldsymbol{x} = \underbrace{\|\boldsymbol{w}\|\|\boldsymbol{x}\|}_{>0} \cos\angle(\boldsymbol{w},\boldsymbol{x}) > 0 \iff \cos\angle(\boldsymbol{w},\boldsymbol{x}) > 0.$$

Notice how the decision rule is invariant to the scale of $\boldsymbol{w}$, and only considers its direction.



Furthermore, we showed that the geometric (signed) margin of $\boldsymbol{x} \in \mathbb{R}^d$ is:

$$\frac{\boldsymbol{w}^\top \boldsymbol{x}}{\|\boldsymbol{w}\|}$$

4. A nonhomogeneous linear classifier $h: \mathbb{R}^d \to \mathbb{R}$ is parameterized by a vector $\boldsymbol{w} \in \mathbb{R}^d$ and a bias $b \in \mathbb{R}$, such that:

$$h(\boldsymbol{x}) = +1 \Leftrightarrow \boldsymbol{w}^\top \boldsymbol{x} > -b.$$

Notice how b can almost be interpreted as the minimal margin required for a positive prediction (we say "almost" since $\boldsymbol{w}^\top \boldsymbol{x}$ is not the margin, but rather $\frac{\boldsymbol{w}^\top \boldsymbol{x}}{\|\boldsymbol{w}\|}$ is).

4.1. In the homogeneous case, we had a degree of freedom while choosing $\boldsymbol{w}$ (due to the scale invariance). What similar degree of freedom do we have while choosing $\boldsymbol{w}, b$ in the non-homogeneous case? That is, how can $\boldsymbol{w}, b$ be changed without changing the classifier itself? Briefly explain your answer.

Answer:

W,b has to be scalled by the same positive factor in order to preserve the classifier function. This is because:

$$h(x) = +1 \Leftrightarrow w^\top x > -b \ \Leftrightarrow w^\top x + b > 0 \ \Leftrightarrow a(w^\top x + b) > 0 \ \ \Leftrightarrow \ 0 < a \in \mathbb{R}$$

4.2. Following are 4 figures of decision boundaries induced by the same $\boldsymbol{w}$ with different biases $b \in \mathbb{R}$. The biases are ordered as: $b_1 > b_2 > 0 > b_3 > b_4$.

Which bias corresponds to which figure?

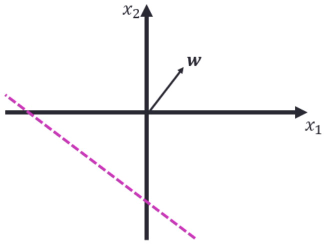
Write your answer in the following format (answers are made up): A is b_1, B is b_2, C is b_3, D is b_4.

No explanations are required.

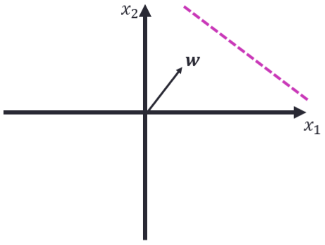
Answer:

A is b_1, B is b_4, C is b_2, D is b_3.

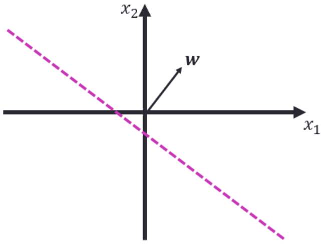
(A)



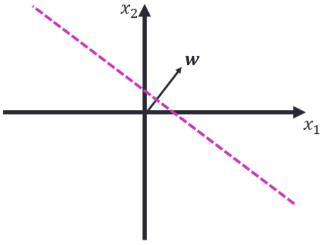
(B)



(C)



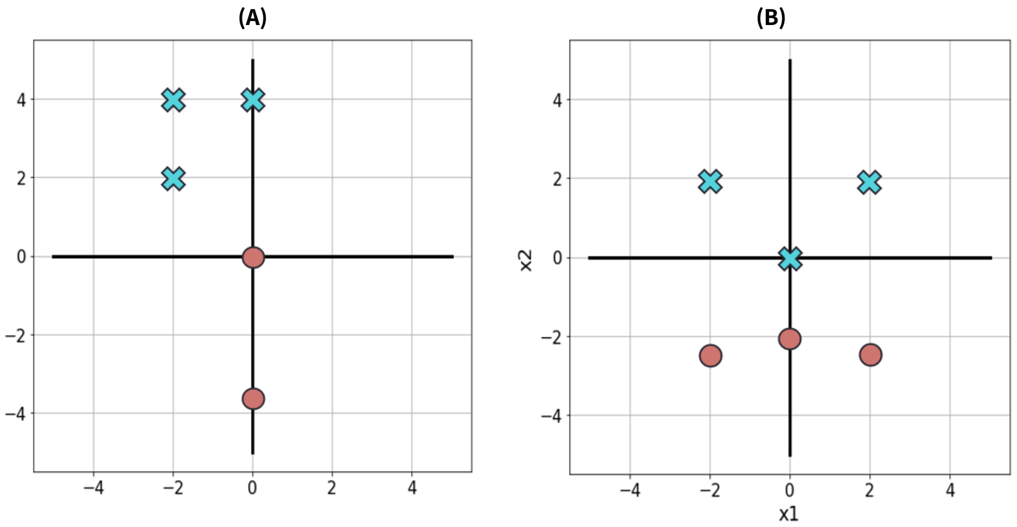
(D)



k Nearest Neighbors

5. Following are 2 training sets in the $\mathbb{R}^2$ feature space with 2 classes (blue/red).

Assume no dataset has two points in the exact same coordinates.



Following are 3 models.

i. kNN with $k = 1$ (a training point is not considered a neighbor of itself) w.r.t. $d(\mathbf{u},\mathbf{v}) = \|\mathbf{u}-\mathbf{v}\|_2$
ii. kNN with $k = 3$ (a training point is not considered a neighbor of itself) w.r.t. $d(\mathbf{u},\mathbf{v}) = \|\mathbf{u}-\mathbf{v}\|_2$
iii. kNN with $k = 1$ (a training point is not considered a neighbor of itself) w.r.t. $d(\mathbf{u},\mathbf{v}) = \|\mathbf{u}-\mathbf{v}\|_1$

5.1. For each model above, write which datasets this model can perfectly fit (i.e., with 0 training error) and which datasets it cannot. Write your answers in a <u>table</u>, like in the example below. When you say a model <u>cannot</u> perfectly fit a certain dataset, explain why in 1-2 sentences (without drawings).

Example for a table (answers are random):

Model / dataset	**(A)**	**(B)**
i.	No. (0,0) will be labeled blue, because $d(blue,red) = d\big((0,0),(-2,2)\big) = 2\cdot\sqrt{2} < d\big((0,0),(0,-3)\big) < d(red,red)$	No. (0,0) will be labeled red, because $d(blue,red) = d\big((0,0),(0,-2)\big) = 2 < 2\cdot\sqrt{2} = d\big((0,0),(2,2)\big) = d(blue,blue)$
ii.	No. Any red point has only one red neighbour and two other blue nearest neighbours. Hence, it will be labeled blue.	Yes.
iii.	Yes.	No. The above statement still hold, (0,0) will be labeled red although the reason now is: $d(blue,red) = d\big((0,0),(0,-2)\big) = 2 < 4 = d\big((0,0),(2,2)\big) = d(blue,blue)$

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$$

לכל $p \geq 1$ ממשי קבוע.

5.2. Now assume that the data points from all datasets above are rotated by the same unknown angle θ (around the origin). That is, each 2-dimensional data point $\begin{bmatrix}x_1\\x_2\end{bmatrix}$ is mapped into $[c]\begin{bmatrix}x_1\\x_2\end{bmatrix}$.

Without knowing the exact angle θ, answer <u>for each of the models above</u>:

- Might your answers for that <u>model</u> change?

- If not, briefly explain why.
- Otherwise, the answers for which datasets might change? Briefly explain why.

Answer:

i. Answers unchanged because the distance between thw points won't change hence the uclidian norm will remain the same between all points and nearest neighbour will be as before rotation.

ii. Answers unchanged due to the same argument above, rotation keep uclidian distances between any two points. Also in dataset (A) it wasn't a matter of distance but it had a defect because number of red points.

iii. Answers for might change, depenading on the rotation angle.

Dataset A:

Perhaps for a clockwise 45 degree rotation answers will change. The red dot at the origin before rotation is classified red because in $||\mathbf{u}-\mathbf{v}||_1$ norm, it's the closest. Although after rotation the closest point is the blue point at (-2,2) before rotation, which is placed at $(2\sqrt{2},0)$ after rotation and the red dot before rotation is at around (0,-4) and after rotation is around $(2\sqrt{2},-2\sqrt{2})$. Hence

$$d(blue,red) = d\left((0,0),(2\sqrt{2},0)\right) = 2\sqrt{2} < 4\sqrt{2} = d\left((0,0),(2\sqrt{2},-2\sqrt{2})\right) = d(red,red)$$

Dataset B:

The model could be accurate, perhaps for a 45 degree rotation the red dot at (0,-2) before rotation, and after will be at $(\sqrt{2},\sqrt{2})$ while the blue dot at $(2,2)$ before rotation will be at $(0,2\sqrt{2})$. Thus they both will be at a distance of $2\sqrt{2}$ from the blue dot at the origin so depending on the way the model decides, the blue dot has possibility to be labeled correctly.