

Short HW2: Classification: Introduction

Submitted individually by Sunday, 18.02, 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.

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

2. We now wish to prove that the information gain, i.e., $IG(v, a) = H(v) - \frac{|v_{a=T}|}{|v|} H(v_{a=T}) - \frac{|v_{a=F}|}{|v|} H(v_{a=F})$, 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:

$$IG(A) = Entropy(D) - \sum (|D_v| / |D| * Entropy(D_v))$$

where $Entropy(D)$ is the entropy of the original dataset D, $|D_v|$ is the number of instances in the subset of D that have a specific value v for the attribute A, and $Entropy(D_v)$ is the entropy of that subset.

Entropy is a measure of the randomness or unpredictability of the target variable, and it is calculated as:

$$Entropy(D) = - \sum (p(c) * \log_2(p(c)))$$

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:

$$Entropy(D) - \sum (|D_v| / |D| * Entropy(D_v)) < 0$$

Rearranging the terms, we get:

$$Entropy(D) < \sum (|D_v| / |D| * Entropy(D_v))$$

Since both $Entropy(D)$ and $Entropy(D_v)$ are non-negative, the right-hand side of the inequality is also non-negative.

Therefore, we have:

$$Entropy(D) \geq 0.$$

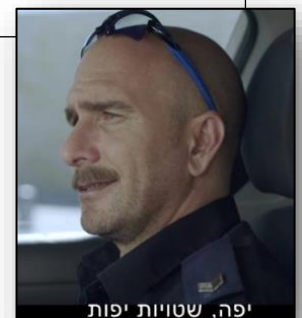
This contradicts the assumption that $IG(A) < 0$, because

$$Entropy(D) - \sum (|D_v| / |D| * Entropy(D_v)) \text{ is the value of } IG(A).$$

Thus, we can conclude that the information gain in the ID3 algorithm is always non-negative and cannot be negative.

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

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



- 2.2. 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: $IG(v, a) \geq 0$.

You can use the following property without proving it.

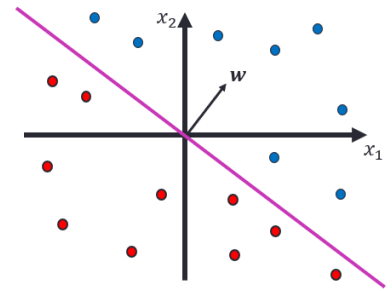
Property: It holds $\forall \beta_1, \beta_2, \alpha \in [0, 1]$ that: $-\alpha H(\beta_1) - (1 - \alpha) H(\beta_2) \geq -H(\alpha \beta_1 + (1 - \alpha) \beta_2)$.

Linear classification

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

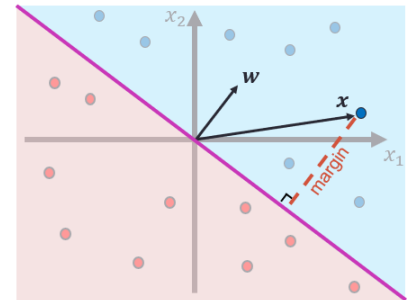
$$h(\mathbf{x}) = +1 \Leftrightarrow \mathbf{w}^T \mathbf{x} = \underbrace{\|\mathbf{w}\| \|\mathbf{x}\|}_{>0} \cos \angle(\mathbf{w}, \mathbf{x}) > 0 \Leftrightarrow \cos \angle(\mathbf{w}, \mathbf{x}) > 0.$$

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



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

$$\frac{\mathbf{w}^T \mathbf{x}}{\|\mathbf{w}\|}$$



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

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

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

- 3.1. In the homogeneous case, we had a degree of freedom while choosing \mathbf{w} (due to the scale invariance).

What similar degree of freedom do we have while choosing \mathbf{w}, b in the non-homogeneous case?

That is, how can \mathbf{w}, b be changed without changing the classifier itself? Briefly explain your answer.

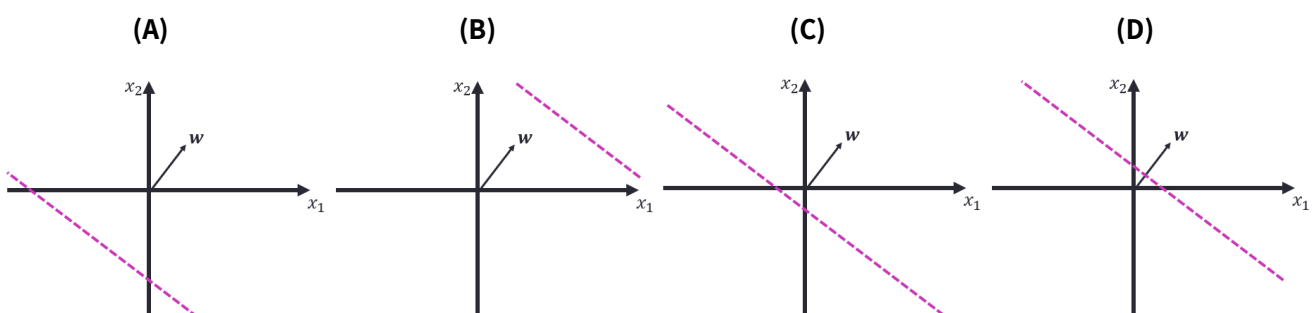
- 3.2. Following are 4 figures of decision boundaries induced by the same \mathbf{w} with different biases $b \in \mathbb{R}$.

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 .

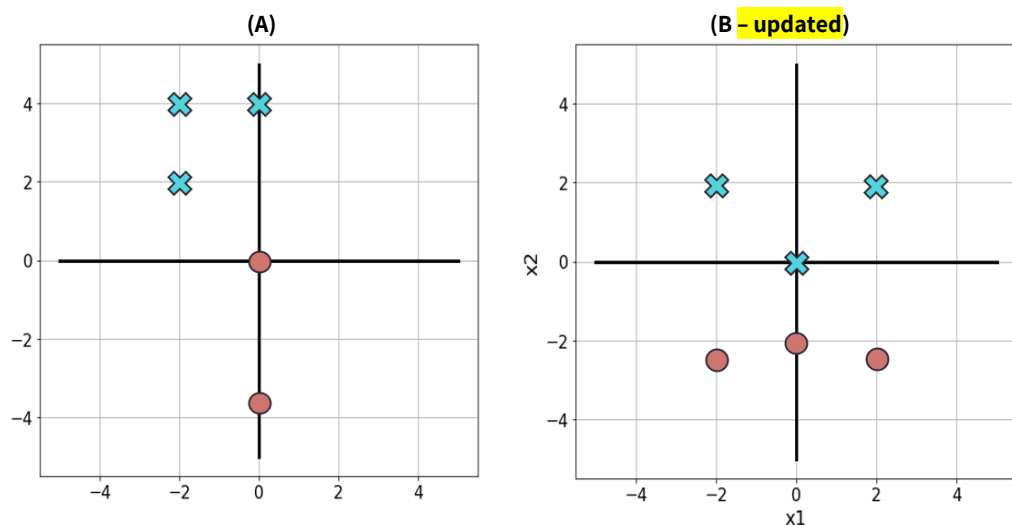
No explanations are required.



k Nearest Neighbors

4. 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.

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

4.1. 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 table, like in the example below. When you say a model cannot 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.	Yes	Yes
ii.	No. For unknown reasons.	Yes.
iii.	Yes	No. Since k is positive.

4.2. 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 $\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

Without knowing the exact angle θ , answer for each of the models above:

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

Answer for example:

- Answers unchanged because this is the best model ever.
- Answers unchanged since this was once revealed to me in a dream.
- Answers on dataset (A) might change because of this and that.