Agenda

Problems on:

- KNN
- Decision Tree
- Questions on HW1
Q1 - Decision Tree to KNN

- This decision tree classifies 2D vectors \( \{X_1, X_2\} \in R \setminus \{A, B\} \).
- In other words, \( X_1 = A \) and \( X_2 = B \) are never used as inputs.
- Can this decision tree be converted to a **1-NN**?
- If so, what is the minimum number of training points you need?
- Else, explain or give a counterexample.
The 4 minimum training points and their labels are:

<table>
<thead>
<tr>
<th>Training Point</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A+1, B+1}</td>
<td>1</td>
</tr>
<tr>
<td>{A+1, B-1}</td>
<td>0</td>
</tr>
<tr>
<td>{A-1, B+1}</td>
<td>0</td>
</tr>
<tr>
<td>{A-1, B-1}</td>
<td>1</td>
</tr>
</tbody>
</table>
Q2.1 - KNN to Decision Trees

- Let’s classify data points in 2D Euclidean space.
- You have $n$ points $P_1, P_2, \ldots, P_n$ and their labels.
- For 1-NN, the input space can be divided as shown in the Voronoi diagram.
Q 2.1

- Is it possible to build a decision tree (with decision boundaries at each node of the form “\(\text{is } x > a\)”, “\(\text{is } x < b\)”, “\(\text{is } y > c\)”, “\(\text{is } y < d\)” for any real constants \(a, b, c, d\)) which classifies according to the 1-NN scheme using the Euclidean distance measure?
A 2.1

- No.
- The decision boundaries for 1-NN correspond to the cell boundaries for each point.
- Decision tree boundaries would always be parallel to the coordinate axes.
- To approximate a gradient for a decision boundary could take arbitrary number of decisions.
Q2.2

- Assume the distance measure is not given to you.
- Instead you have a **black box**, where you input a set of training instances $P_1, P_2, \ldots, P_n$ and a new text example $Q$. The black box returns the NN of $Q$, say $P_i$ and its label $C_i$.
- Is it possible to build a kNN classification algorithm based on this black box alone?
Training set: $T = \{P_1, P_2, \ldots P_n\}$

Test point: $Q$

Nearest neighbour to $Q$ (say $P_i$)

Label of $P_i$
Updated Training set: $T' = \{P_1, P_2, \ldots, P_n\} - \{P_i\}$

Test point: $Q$

Next Nearest neighbour to $Q$ (say $P_j$)

Label of $P_j$
This process if repeated $k$ times returns the $k$ nearest neighbours of point $Q$. 
HW 1 doubts ?