Tutorial 8 (Wk9): kNN and SVM

k-Nearest Neighbours (kNN) Algorithm

To compute kNN classification for a new data point, follow these steps:

1. **Compute distances**: Calculate the distance between the new data point and all training samples. For this task, we will use both *Euclidean distance* and *Manhattan distance*:
   - **Euclidean distance formula:**
   $$d = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$
   - **Manhattan distance formula:**
   $$d = \sum_{i=1}^{n} |a_i - b_i|$$
2. **Identify nearest neighbours**: Determine the k-th nearest neighbours based on their distance values.
3. **Predict the class**: Gather the classes of these neighbours and use a simple majority vote to predict the class for the new instance.

kNN Example

Consider the following quantitative data and corresponding classifications:

$$\text{(strength, thickness)} \rightarrow \text{classification}$$
$$(0, 1) \rightarrow \text{Bad}$$
$$(13, 1) \rightarrow \text{Bad}$$
$$(2, 5) \rightarrow \text{Good}$$
$$(10, 6) \rightarrow \text{Good}$$

Predict the class for the new data point $(6, 1)$ using:

1. $k = 3$ (number of neighbours) and *Euclidean distance*
2. $k = 3$ (number of neighbours) and *Manhattan distance.*

Does the choice of distance metric change the prediction? Consider why this might happen.

Exercise: k-Nearest Neighbours

Given the training data below, predict the class of the following new example using k-Nearest Neighbours for $k = 5$: **age**=≤ 30, **income**=medium, **student**=yes, **credit_rating**=fair. For distance measure between neighbours use a simple match of attribute values:

$$\text{distance}(A, B) = \frac{\sum_{i=1}^{4} w_i \cdot \delta(a_i, b_i)}{4}$$

where $\delta(a_i, b_i)$ is 1 if a_i equals b_i and 0 otherwise. a_i and b_i are either **age**, **income**, **student**, or **credit_rating**. Weights are all 1 except for **income**, where it is 2.

RID	**age**	**income**	**student**	**credit_rating**	**Class: buys_computer**
1	≤ 30	high	no	fair	no
2	≤ 30	high	no	excellent	no
3	$31 \ldots 40$	high	no	fair	yes
4	> 40	medium	no	fair	yes
5	> 40	low	yes	fair	yes
6	> 40	low	yes	excellent	no
7	$31 \ldots 40$	low	yes	excellent	yes
8	≤ 30	medium	no	fair	no
9	≤ 30	low	yes	fair	yes
10	> 40	medium	yes	fair	yes
11	≤ 30	medium	yes	excellent	yes
12	$31 \ldots 40$	medium	no	excellent	yes
13	$31 \ldots 40$	high	yes	fair	yes
14	> 40	medium	no	excellent	no

Support Vector Machines

Go over the following SVM examples (skipping Section 3 if necessary):

SVM example.pdf