Lecture 06 –
Introduction to machine learning

✓ Example: Lenses dataset
✓ Machine learning terminology
✓ Majority class classifier
✓ Decision tree learning:
  ✓ Decision stump
  ✓ Decision tree
  ✓ Learning with fixed feature order
  ✓ Measuring decision tree node purity
  ✓ ID3

• **KNN**
• Example: hand-written digit recognition
• Curse of dimensionality
How to predict? (an alternative)

<table>
<thead>
<tr>
<th>P</th>
<th>Y</th>
<th>H</th>
<th>A</th>
<th>R</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
How to predict? (an alternative)

<table>
<thead>
<tr>
<th>P</th>
<th>Y</th>
<th>H</th>
<th>A</th>
<th>R</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
| 0 | 1 | 1 | 1 | 0 | ???

Let’s find the training instance that is the most similar to the test instance and predict the same value for label
How to predict? (an alternative)

<table>
<thead>
<tr>
<th>P</th>
<th>Y</th>
<th>H</th>
<th>A</th>
<th>R</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>???</td>
</tr>
</tbody>
</table>

Let’s find the training instance that is the most similar to the test instance and predict the same value for label
Let's find the training instance that is the most similar to the test instance and predict the same value for label.
Algorithm: 1-Nearest-neighbour

<table>
<thead>
<tr>
<th>P</th>
<th>Y</th>
<th>H</th>
<th>A</th>
<th>R</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Actually, on this dataset there is another training instance at the same distance, so result could also be \( L = 1 \) (depends on the implementation which is chosen, for example the first one in the table)
Classification algorithm: KNN

• Requires the choice of a distance measure, for example:
  – Hamming distance (number of features with a different value between instances)
  – Euclidean distance

• KNN – K nearest neighbours
  1. Find the K training instances which are the nearest to the given test instance (these are called neighbours)
  2. Predict the class with the highest frequency among the K neighbours
Evaluating classification algorithms

• Learn the models on the training data
• Apply the models to the test data to get label predictions
• Accuracy = the proportion of correct predictions
  – Ranges from [0,1] or [0%,100%]
• Error rate = the proportion of wrong predictions (the complement of accuracy)
• Many more evaluation measures
  – Some of these (precision, recall, F-measure) we will discuss in the course
Example: Lenses dataset

Machine learning terminology

Majority class classifier

Decision tree learning:
- Decision stump
- Decision tree
- Learning with fixed feature order
- Measuring decision tree node purity
- ID3

KNN

Example: hand-written digit recognition

Curse of dimensionality
MNIST dataset
(10000 images)

could be downloaded from: http://yann.lecun.com/exdb/mnist/
MNIST dataset (10000 images)

could be downloaded from: http://yann.lecun.com/exdb/mnist/
MNIST dataset
(10000 images)

could be downloaded from: http://yann.lecun.com/exdb/mnist/
MNIST dataset (10000 images)

In total 784 pixel values

3

Adapted from slides by Dmytro Fishman
The MNIST dataset contains 10,000 images, each of size 28x28 pixels. In total, there are 784 pixel values per image. Each image is labeled with a digit from 0 to 9.

Adapted from slides by Dmytro Fishman
### MNIST dataset

(10000 images)

The MNIST dataset consists of 10,000 digit images, each 28x28 pixels in size. The images are labeled with one of ten classes, representing the digit they depict.

#### Feature

Pixel values:

- **3**:
  - Pixel values: 0 0 0 0 0 0 0 31 132 254 253 254 213 82 0 0 0 0 0 0 0
  - Label: 3

- **6**:
  - Pixel values: 0 0 0 0 0 0 0 25 142 254 254 193 30 0 0 0 0 0 0 0
  - Label: 6

- **1**:
  - Pixel values: 0 0 0 0 0 0 0 123 254 87 0 0 0 0 0 0 0 0 0
  - Label: 1

- **8**:
  - Pixel values: 0 0 0 0 0 0 0 59 163 254 254 254 194 112 18 0 0 0 0
  - Label: 8

- **1**: (Continued)
  - Pixel values: 0 0 0 0 0 0 0 19 227 254 84 0 0 0 0 0 0 0 0
  - Label: 1

#### Note

Adapted from slides by Dmytro Fishman

Meelis Kull - Autumn 2020 - LTAT.02.002 – Intro to Data Science - Lecture 06
The images are **784-dimensional** (have 784 features)

<table>
<thead>
<tr>
<th>Instances</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Images](0 0 0 0 0 0 0 31 132 254 253 254 213 82 0 0 0 0 0 0 ...)</td>
<td>3</td>
</tr>
<tr>
<td>![Images](0 0 0 0 0 0 0 25 142 254 254 193 30 0 0 0 0 0 0 0 ...)</td>
<td>6</td>
</tr>
<tr>
<td>![Images](0 0 0 0 0 0 0 123 254 87 0 0 0 0 0 0 0 0 0 ...)</td>
<td>1</td>
</tr>
<tr>
<td>![Images](0 0 0 0 0 0 0 59 163 254 254 254 194 112 18 0 0 0 0 ...)</td>
<td>8</td>
</tr>
<tr>
<td>![Images](0 0 0 0 0 0 0 19 227 254 84 0 0 0 0 0 0 0 0 ...)</td>
<td>1</td>
</tr>
</tbody>
</table>

Adapted from slides by Dmytro Fishman

Meelis Kull - Autumn 2020 - LTAT.02.002 – Intro to Data Science - Lecture 06
These images are 784-dimensional (have 784 features)

Adapted from slides by Dmytro Fishman
Let’s use KNN with \( K=1 \)

Test instance A

Training instances B, C

Adapted from slides by Dmytro Fishman
Let’s use 1-NN: how to find the nearest?

Test instance A & Training instance B OR Test instance A & Training instance C

Adapted from slides by Dmytro Fishman
Let’s use 1-NN: how to find the nearest?

How about computing their **pixel-wise difference**?
Let’s use 1-NN: how to find the nearest?

Test instance A & Training instance B

\[ \sum_{i}^{784} | A_i - B_i | \]

OR

Test instance A & Training instance C

\[ \sum_{i}^{784} | A_i - C_i | \]

Adapted from slides by Dmytro Fishman
Let’s use 1-NN: how to find the nearest?

\[ \sum_{i} |A_i - B_i| = 36456 \]

\[ \sum_{i} |A_i - C_i| = 24026 \]

Adapted from slides by Dmytro Fishman
Let’s use 1-NN: how to find the nearest?

\[ \sum_{i} |A_i - B_i| = 36456 \]

\[ \sum_{i} |A_i - C_i| = 24026 \]

A is more similar (closer) to C than B

Adapted from slides by Dmytro Fishman
For each new test instance:

<table>
<thead>
<tr>
<th>Instance</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>?</td>
</tr>
</tbody>
</table>

Adapted from slides by Dmytro Fishman
For each new test instance:

Instance | Label
--- | ---
3 | ?

Training data:

```
2 3 8 9 4 4 5 0 7 6
0 3 6 1 1 6 1 0 8 6
4 3 7 3 3 5 9 1 9
6 1 2 8 1 4 8 2 9 6
5 5 4 0 8 7 2 2 5 2
0 8 2 8 2 8 0 3 8 2
1 3 3 2 0 2 1 4 1 5
6 2 4 9 6 9 4 1 1 8
6 7 9 3 2 1 6 6 6 0
1 4 3 3 0 9 2 8 0 3
6 0 6 9 5 8 0 5 7 2
5 8 9 1 1 0 7 3 7 1
7 6 8 3 6 6 7 1 9 0
```
For each new test instance:

1. Compute **pixel-wise distance** to all training examples

Adapted from slides by Dmytro Fishman
For each new test instance:

1. Compute **pixel-wise distance** to all training examples
For each new test instance:

1. Compute **pixel-wise distance** to all training examples
For each new test instance:

1. Compute **pixel-wise distance** to all training examples
For each new test instance:

1. Compute **pixel-wise distance** to all training examples

2. Find the **closest** training example

---

Adapted from slides by Dmytro Fishman

---
For each new test instance:

1. Compute **pixel-wise distance** to all training examples

2. Find the **closest** training example
For each new test instance:

1. Compute **pixel-wise distance** to all training examples
2. Find the **closest** training example
3. Report it’s label

Training data:

Adapted from slides by Dmytro Fishman

Meelis Kull - Autumn 2020 - LTAT.02.002 - Intro to Data Science - Lecture 06
For each new test instance:

1. Compute **pixel-wise distance** to all training examples
2. Find the **closest** training example
3. Report it's label

Training data:

1-NN classifier

Adapted from slides by Dmytro Fishman
KNN

• Advantages
  – Very easy to implement
  – A good choice for low-dimensional problems

• Disadvantages
  – Very slow in classification time
  – Suffers from the curse of dimensionality
Example: Lenses dataset

Machine learning terminology

Majority class classifier

Decision tree learning:
  - Decision stump
  - Decision tree
  - Learning with fixed feature order
  - Measuring decision tree node purity
  - ID3

KNN

Example: hand-written digit recognition

- Curse of dimensionality
Curse of dimensionality

• In a high-dimensional unit hyper-cube almost all data are ”in the corners”:
  – Hyper-ball with unit diameter has vanishingly small volume compared to the unit hyper-cube

• In a high-dimensional unit hyper-cube distribution of pairwise distances is very peaked:
  – For randomly chosen 2 points the distance is almost surely very close to the average distance
Lecture 06 –
Introduction to machine learning

✓ Example: Lenses dataset
✓ Machine learning terminology
✓ Majority class classifier
✓ Decision tree learning:
  ✓ Decision stump
  ✓ Decision tree
  ✓ Learning with fixed feature order
  ✓ Measuring decision tree node purity
  ✓ ID3
✓ KNN
✓ Example: hand-written digit recognition
✓ Curse of dimensionality