Course. Introduction to Machine Learning

Work 3. Lazy Learning exercise

Session 1

Dr. Maria Salamó Llorente

Dept. Mathematics and Informatics,

Faculty of Mathematics and Informatics,
University of Barcelona

Work 2 team evaluation

- Mandatory form to evaluate group work in the lab (IML_W2 22-23). The same form will be filled by every member of the group (individually)

- *Please use this form to evaluate the contributions of each team member to the group effort. Consider attendance and participation in team meetings, individual contributions to idea generation and research, communication within the group, etc.*

***These evaluations are COMPLETELY CONFIDENTIAL and will never be shown to your team members**. Please respond as honestly as possible.*

WORK1.
https://forms.gle/qZLwTUDLnf9c1KsW6

WORK2.
https://forms.gle/Gy6XkSKaqea5YUp46

Contents

1. Introduction (Session 1)
2. Parameters in k-IBL (Session 1)
   1. Distance metric
   2. K parameter
   3. Voting schemes
   4. Retention policies
3. Feature Selection techniques (Session 2)
4. Instance Selection techniques (Session 2)

Introduction

Introduction

- Many lazy learning techniques exist
  - This course we will concentrate on IBL
- In lazy learning, storing and using specific instances may improve its performance
- IBL algorithms usually store all the training set but this causes:
  - A large storage is needed
  - The generalization process is slow
  - The data may contain inconsistencies and noise
- To deal with these problems, feature selection and instance selection techniques are used

Introduction

The **goal** of Work 3 is to…

1. Implement a k-IBL algorithm, based on IB1 (Week 1)
2. Implement k-IBL parameters
   1. Euclidean distance metric (Week 1)
   2. Voting schemes (Week 1)
   3. Retention approaches (Week 1)
3. Analyze Best k-IBL algorithm (Week 2)
4. Implement feature selection techniques (Week 3)
5. Implement instance selection techniques (Week 3)
6. Compare K-IBL with and without feature selection and instance selection techniques using different metrics: accuracy, efficiency and storage (Week 4)
7. Perform statistical analysis and write report (Week 4)

Introduction

- Four things make a lazy learner:
  - A distance metric
  - How many nearby neighbors to look at?
  - A weighting function (optional)
  - How to fit with the local points?

Introduction

- Instance-based learning, IB1
  - A distance metric: **Euclidean distance**
  - How many nearby neighbors to look at? **One**
  - A weighting function (optional): **Unused**
  - How to fit with the local points?: **Just predict the same output as the nearest neighbor**

IB1

Table 1. The IB1 algorithm (*CD* = Concept Description).

$CD \leftarrow \emptyset$

for each $x \in$ Training Set **do**

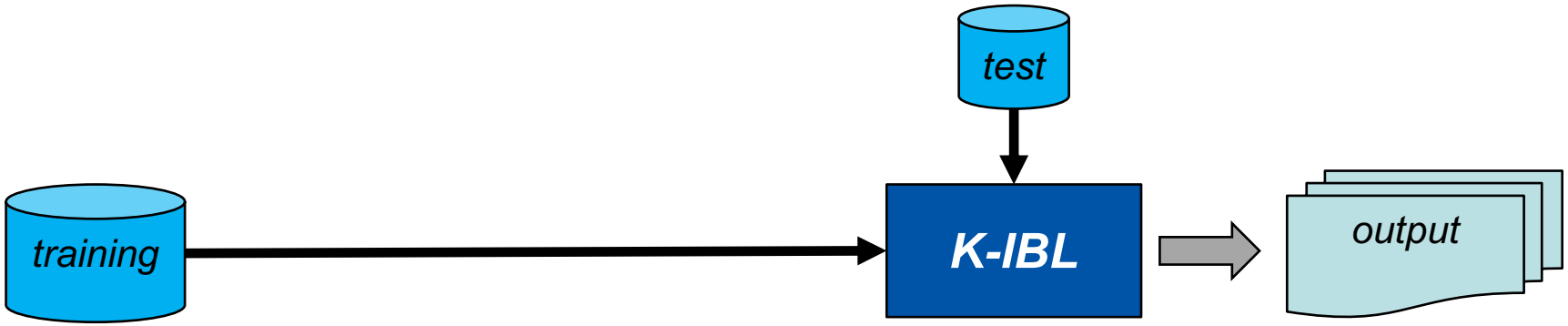
1. **for each** $y \in CD$ **do**
   Sim[y] $\leftarrow$ Similarity(x, y)
2. $y_{max} \leftarrow$ some $y \in CD$ with maximal Sim[y]
3. **if** class(x) = class(y_{max})
   then classification $\leftarrow$ **correct**
   else classification $\leftarrow$ **incorrect**
4. $CD \leftarrow CD \cup \{x\}$

Similarity based on the NN

Retention

Introduction

- In this work you implement and analyze **k-Instance Based Learning** algorithm
- **Special attention to:**
  - Distance metrics: **Euclidean**
  - How many nearby neighbors to look at? **One or K**
  - A weighting function (optional): **Assume equal weight**
  - How to fit with the local points?: **Evaluate 2 voting rules**
  - How to learn the new solved cases? **Evaluate 4 policies**

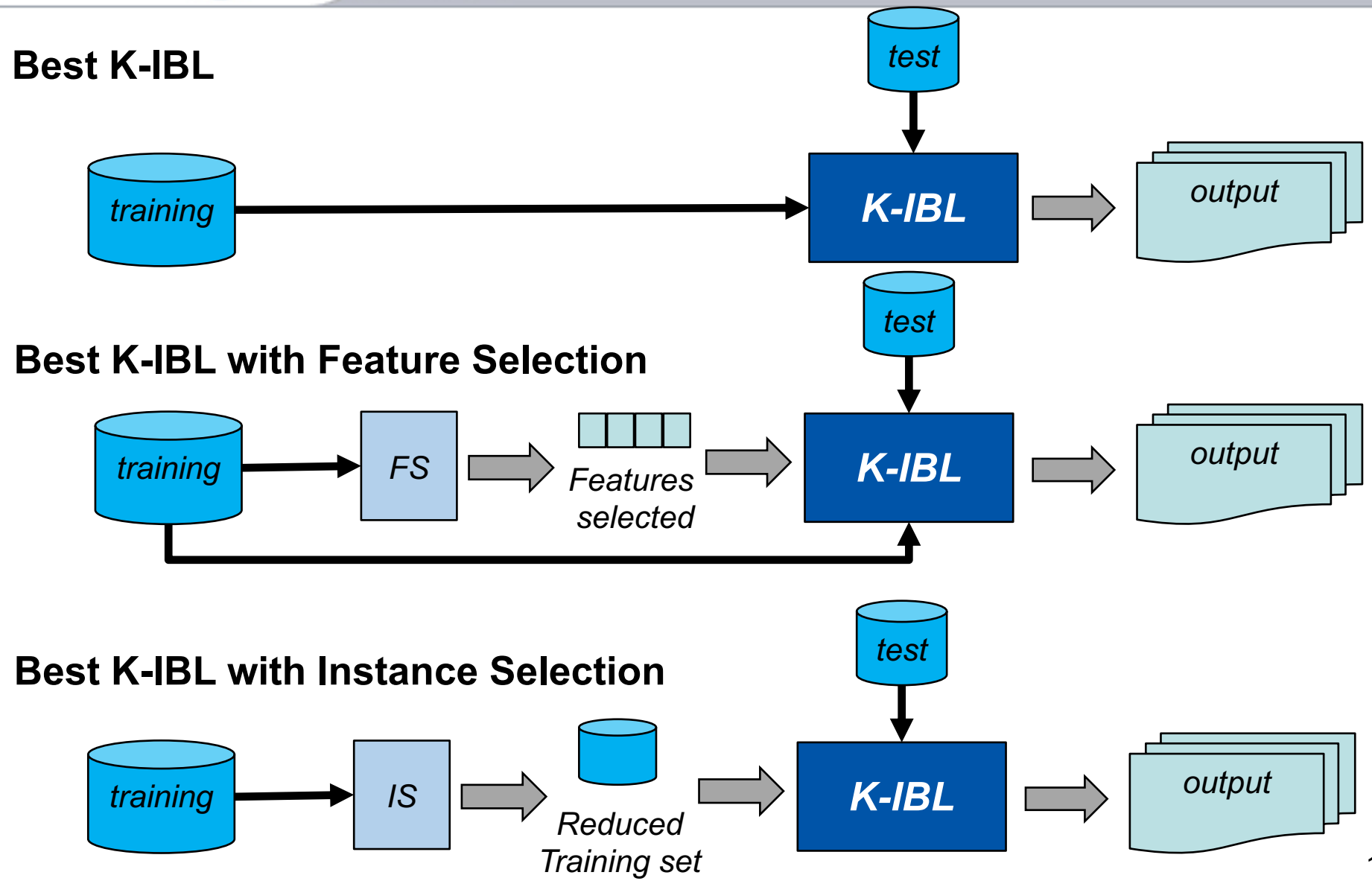


What to do in Work 3?

- Adapt the parser to read the class and the 10 fold cross-validation sets

- Find the **best K-IBL algorithm**
  - K = 1, 3, 5
  - Euclidean metric
  - Voting schemes
    - Modified Plurality and Borda Count
  - Weighting strategies
    - Equal weight
  - Retention policies
    - Never retain, Always retain, Different Class retention, Degree of Disagreement
- Apply feature selection (FS) to the **best k-IBL algorithm**
- Apply instance selection (IS) to the **best k-IBL algorithm**
- Evaluate the results with an statistical analysis
- Write the report

What to do in Work 3?



Parameters in k-IBL

Parameters in k-IBL

Work 3 concentrates on the following parameters:

1. **Distance metrics**
   - Implement your own code
   - You will implement an Euclidean metric that is one of the most used in the literature for lazy learning classifiers
2. **K parameter**
   - Analyze several predefined values
   - It is out of the scope of this work to deal with algorithms that automatically set up this parameter
3. **Voting schemes**
   - Implement your own code
4. **Weighting**
   - You will assume equal weight to all features in this work

Distance metrics

- Computes the distance between two examples
  - So that we can find the “nearest neighbor” to a given example

Derived from Minskowski’s metric:

$$d(C_i,C_j) = \left(\sum_{k=1}^{n} |C_{ik} - C_{jk}|^r \right)^{1/r} \qquad r \geq 1$$

- **Manhattan or city block distance (r = 1)**
  - Sum of axis-parallel line segments
- **Euclidean distance (r = 2)**
  - Straight-line between two points

UNIVERSITAT DE BARCELONA

(Wilson, D.R., Martínez, T.R., 1997)

Minkowsky:

$$D(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{m} |x_i - y_i|^r \right)^{1/r}$$

Euclidean:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

Manhattan / city-block

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} |x_i - y_i|$$

Camberra:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|}$$

Chebychev:

$$D(\boldsymbol{x},\boldsymbol{y}) = \max_{i=1}^{m} |x_i - y_i|$$

Quadratic:

$$D(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})^T Q(\boldsymbol{x}-\boldsymbol{y}) = \sum_{j=1}^{m} \left(\sum_{i=1}^{m} (x_i - y_i) q_{ji} \right) (x_j - y_j)$$

Q is a problem-specific positive definite $m \times m$ weight matrix

Mahalanobis:

$$D(\boldsymbol{x},\boldsymbol{y}) = [\det V]^{1/m} (\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}} V^{-1} (\boldsymbol{x}-\boldsymbol{y})$$

V is the covariance matrix of $A_1..A_m$, and A_j is the vector of values for attribute j occuring in the training set instances 1..n.

Correlation:

$$D(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{m} (x_i - \mu_i)(y_i - \mu_i)}{\sqrt{\sum_{i=1}^{m} (x_i - \mu_i)^2 \sum_{i=1}^{m} (y_i - \mu_i)^2}}$$

μ_i is the average value for attribute i occuring in the training set.

Chi-square:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{1}{sum_i} \left(\frac{x_i}{size_{\boldsymbol{x}}} - \frac{y_i}{size_{\boldsymbol{y}}} \right)^2$$

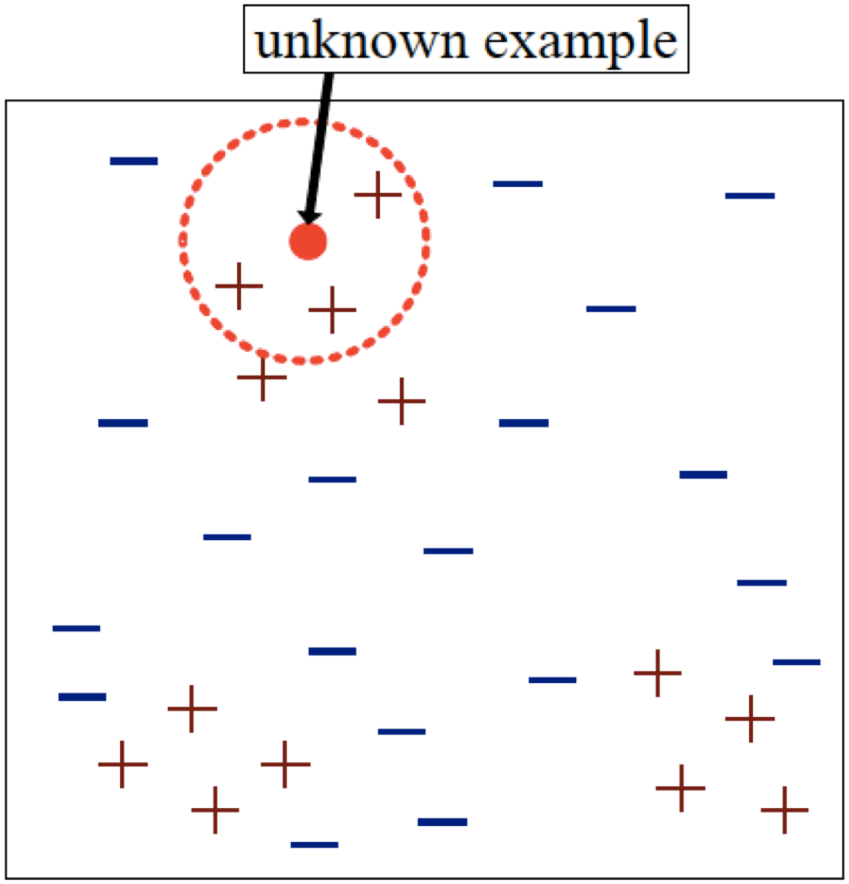
sum_i is the sum of all values for attribut i occuring in the training set, and $size_x$ i the sum of all values in the vector $\boldsymbol{x}$.

Kendall's Rank Correlation:

$$D(\boldsymbol{x},\boldsymbol{y}) = 1 - \frac{2}{n(n-1)} \sum_{i=1}^{m} \sum_{j=1}^{i-1} \mathrm{sign}(x_i - x_j)\,\mathrm{sign}(y_i - y_j)$$

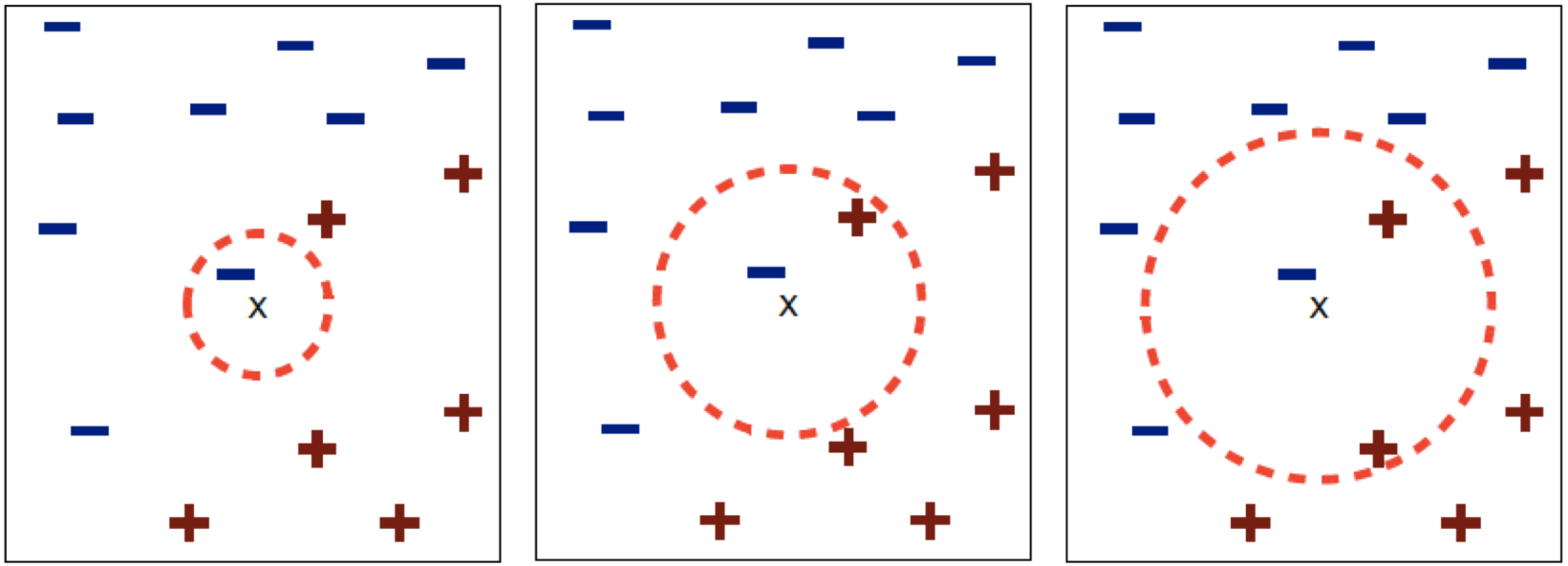
sign(x)=-1, 0 or 1 if $x < 0$, $x = 0$, or $x > 0$, respectively.

K-Nearest Neighbor



- Require three things
  - The set of stored examples
  - Distance Metric to compute distance between examples
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unknown example:
  - Compute distance to other training examples
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown example (e.g., by taking majority vote)

K-Nearest Neighbors

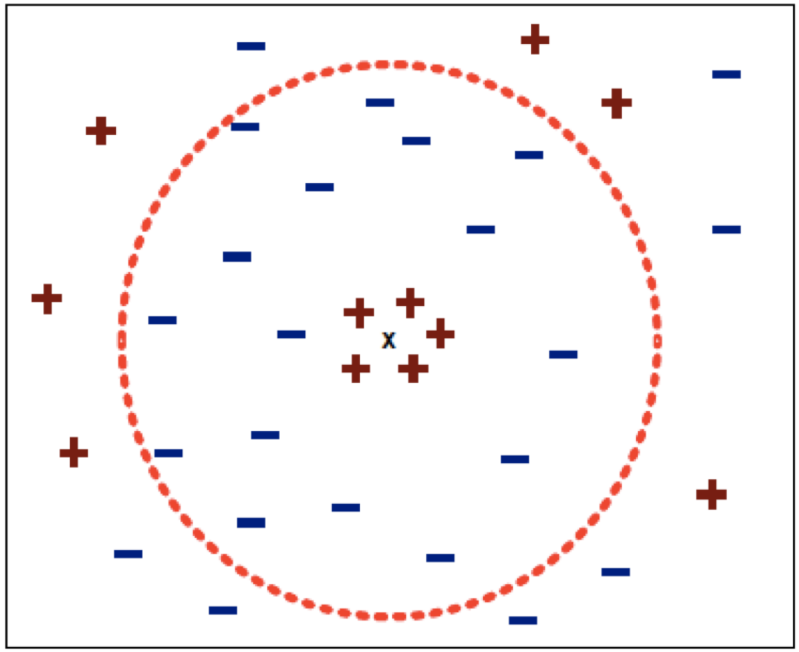


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

k nearest neighbors of an example x are the data points that have the k smallest distances to x

Choosing the k value?

- If k is too small
  - sensitive to noise in the data (misclassified examples)
- If k is too large
  - neighborhood may include points from other classes
  - limiting case: $k \geq |D|$
    - all examples are considered
    - largest class is predicted



- good values can be found
  - e.g, by evaluating various values with cross-validation on the training data

Voting Schemes

- **Modified Plurality:** computes the most voted solution but in case of ties, it removes the last k nearest neighbor and computes again the most voted solution
  - In case of ties, the process is repeated. The process finishes when there is a winner solution or when just one nearest neighbor remains

- **Borda count:** Borda voting rule assigns k – 1 points to the solution of the most similar instance, k – 2 points to the second k nearest neighbor, or k – k to the solution that is ranked in k-th
  - The winner is the solution that amasses the highest total number of points
  - A method for breaking ties should also be specified. You should decide what to do in this case

Retention policies

- **Never retain (NR)**: The IBL never retains the current instance q in the instance base
  - The generalization process is based exclusively on the instances present in the training set.
- **Always retain (AR):** The algorithm retains all new solved instances in the instance base
- **Different Class retention (DF):** The algorithm retains all the instances that have been solved incorrectly
  - Similar to IB2
- **Degree of Disagreement (DD):** Measures the dispersion -or disagreement- of the k most similar instances.
  - The proposal is to use the k set as a virtual committee. So, each member of the committee can be seen as a case in k, and the vote it generates as the class associated to this case.

$$d = \frac{\#remaining\_cases}{(\#classes - 1) \times \#majority\_cases}$$

Course. Introduction to Machine Learning

Work 3. Lazy Learning exercise

Session 1

Dr. Maria Salamó Llorente

Dept. Mathematics and Informatics,

Faculty of Mathematics and Informatics,
University of Barcelona