ADDRESSING MULTI-CLASS PROBLEMS BY Binarization. Novel Approaches.
Outline

1. Introduction
2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All
3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis
4. Discussion: Lessons Learned and Future Work
5. Conclusions for OVO vs OVA
6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)
   - Difficult Classes Problem in OVO Strategy
Outline

1. Introduction
2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All
3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis
4. Discussion: Lessons Learned and Future Work
5. Conclusions for OVO vs OVA
6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)
Introduction

Classification

- 2 class of classification problems:
  - Binary: medical diagnosis (yes / no)
  - Multicategory: Letter recognition (A, B, C...)

- Binary problems are usually easier
- Some classifiers do not support multiple classes
  - SVM, PDFC...
For this competition, we have provided a dataset with 93 features for more than 200,000 products. The objective is to build a predictive model which is able to distinguish between our main product categories. The winning models will be open sourced.
Introduction. Ejemplo

Una aplicación real en KAGGLE de Problema Multiclase

Submission Format

You must submit a csv file with the product id, all candidate class names, and a probability for each class. The order of the rows does not matter. The file must have a header and should look like the following:

```
id,Class_1,Class_2,Class_3,Class_4,Class_5,Class_6,Class_7,Class_8,Class_9
1,.0,.0,.0,.0,.0,.0,.0,.0,.0
2,.0,.0,.0,.0,.3,.0,.0,.0,.0
...  
```

1987 teams
2117 players
15502 entries

Started: 3:56 pm, Tuesday 17 March 2015 UTC
Ends: 11:59 pm, Monday 18 May 2015 UTC (62 total days)
Points: this competition awards standard ranking points
Tiers: this competition counts towards tiers
Introduction. Ejemplo

Una aplicación real en KAGGLE de Problema Multiclase

Evaluation

Submissions are evaluated using the multi-class logarithmic loss. Each product has been labeled with one true category. For each product, you must submit a set of predicted probabilities (one for every category). The formula is then,

$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log(p_{ij}),$$

where $N$ is the number of products in the test set, $M$ is the number of class labels, $\log$ is the natural logarithm, $y_{ij}$ is 1 if observation $i$ is in class $j$ and 0 otherwise, and $p_{ij}$ is the predicted probability that observation $i$ belongs to class $j$.

The submitted probabilities for a given product are not required to sum to one because they are rescaled prior to being scored (each row is divided by the row sum). In order to avoid the extremes of the log function, predicted probabilities are replaced with $\max(\min(p, 1 - 10^{-15}), 10^{-15})$. 
Una aplicación real en KAGGLE de Problema Multiclase
Una aplicación real en KAGGLE de Problema Multiclase

Introduction. Ejemplo
1. Introduction
2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All
3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis
4. Discussion: Lessons Learned and Future Work
5. Conclusions for OVO vs OVA
6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)
Binarization

- **Decomposition of the multi-class problem**
  - Divide and conquer strategy
  - Multi-class $\rightarrow$ Multiple easier to solve binary problems
    - For each binary problem
      - 1 binary classifier = base classifier
    - Problem
      - How we should make the decomposition?
      - How we should aggregate the outputs?

- Ensemble of classifiers

  Aggregation or Combination

  Final Output
Decomposition Strategies

- “One-vs-One” (OVO)
  - 1 binary problem for each pair of classes
    - Pairwise Learning, Round Robin, All-vs-All...
    - Total = $m(m-1) / 2$ classifiers
One-vs-One

- **Advantages**
  - Smaller (number of instances)
  - Simpler decision boundaries
    - Digit recognition problem by pairwise learning
      - linearly separable [Knerr90] (first proposal)
  - Parallelizable
  - ...

One-vs-One

- Disadvantages
  - Higher testing times (more classifiers)
  - Non-competent examples [Fürnkranz06]

- Many different aggregation proposals
  - Simplest: Voting strategy
    - Each classifier votes for the predicted class
    - Predicted = class with the largest nº of votes

\[
R = \begin{pmatrix}
- & r_{12} & \cdots & r_{1m} \\
\vdots & - & \cdots & \vdots \\
r_{m1} & r_{m2} & \cdots & - \\
\end{pmatrix}
\]

Related works

- Round Robin Ripper (R3) [Fürnkranz02]
- Fuzzy R3 (FR3) [Huhn09]
- Probability estimates by Pairwise Coupling [Wu04]
- Comparison between OVO, Boosting and Bagging
- Many aggregation proposals
  - There is not a proper comparison between them


Decomposition Strategies

- **“One-vs-All” (OVA)**
  - 1 binary problem for each class
  - All instances in each problem
    - Positive class: instances from the class considered
    - Negative class: instances from all other classes
  - Total = \( m \) classifiers

![Diagram](image)
One-vs-All

Advantages
- Less nº of classifiers
- All examples are “competent”

Disadvantages
- Less studied in the literature
  - low nº of aggregations
    - Simplest: Maximum confidence rule ($\max(r_{ii})$)

More complex problems
Imbalance training sets
One-vs-All

- Related Works
  - Rifkin and Klautau [Rifkin04]
    - Critical with all previous literature about OVO
    - OVA classifiers are as accurate as OVO when the base classifier are fine-tuned (about SVM)

- In general
  - Previous works proved goodness of OVO
    - Ripper and C4.5, cannot be tuned

Decomposition Strategies

- Other approaches
  - ECOC (Error Correcting Output Code) [Allwein00]
    - Unify (generalize) OVO and OVA approach
    - Code-Matrix representing the decomposition
      - The outputs forms a code-word
      - An ECOC is used to decode the code-word
        - The class is given by the decodification

<table>
<thead>
<tr>
<th>Class</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C5</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Class2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>Class3</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>Class4</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Other approaches

Hierarchical approaches
- Distinguish groups of classes in each node

Detailed review of decomposition strategies in [Lorena09]
- Only an enumeration of methods
- Low importance to the aggregation step

Combination of the outputs

- Aggregation phase
  - The way in which the outputs of the base classifiers are combined to obtain the final output.
  - Key-factor in OVO and OVA ensembles
  - Ideally, voting and max confidence works
    - In real problems
      - Contradictions between base classifiers
      - Ties
      - Base classifiers are not 100% accurate
      - ...
1. Introduction
2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All
3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis
4. Discussion: Lessons Learned and Future Work
5. Conclusions for OVO vs OVA
6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)
State-of-the-art on aggregation for OVO

Starting from the score-matrix

\[ R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \]

- \( r_{ij} \) = confidence of classifier in favor of class \( i \)
- \( r_{ji} \) = confidence of classifier in favor of class \( j \)
  - Usually: \( r_{ji} = 1 - r_{ij} \) (required for probability estimates)
State-of-the-art on aggregation for OVO

- **Voting strategy (VOTE) [Friedman96]**
  - Each classifier gives a vote for the predicted class
  - The class with the largest number of votes is predicted

\[
Class = \arg \max_{i=1,\ldots,m} \sum_{1 \leq j \neq i \leq m} s_{ij}
\]

- where \( s_{ij} \) is \( 1 \) if \( r_{ij} > r_{ji} \) and \( 0 \) otherwise.

- **Weighted voting strategy (WV)**
  - WV = VOTE but weight = confidence

\[
Class = \arg \max_{i=1,\ldots,m} \sum_{1 \leq j \neq i \leq m} r_{ij}
\]

State-of-the-art on aggregation for OVO

- Classification by Pairwise Coupling (PC)[Hastie98]
  - Estimates the joint probability for all classes
    - Starting from the pairwise class probabilities
      - \( r_{ij} = \text{Prob} (\text{Class}_i \mid \text{Class}_i \text{ or Class}_j) \)
    - Find the best approximation
    - Predicts:
  - Algorithm: Minimization of Kullback-Leibler (KL) distance
    
    \[
    l(p) = \sum_{1 \leq i < j \leq m} n_{ij} r_{ij} \log \frac{r_{ij}}{\mu_{ij}} = \sum_{i < j} n_{ij} \left( r_{ij} \log \frac{r_{ij}}{\mu_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \mu_{ij}} \right)
    \]
    - where \( \mu_{ij} = p_i / (p_i + p_j) \), \( r_{ji} = 1 - r_{ij} \) and \( n_{ij} \) is the number of examples of classes i and j.

State-of-the-art on aggregation for OVO

- **Decision Directed Acyclic Graph (DDAG) [Platt00]**
  - Constructs a rooted binary acyclic graph
    - Each node is associated to a list of classes and a binary classifier
    - In each level a classifier discriminates between two classes
      - The class which is not predicted is removed
    - The last class remaining on the list is the final output class.

State-of-the-art on aggregation for OVO

- Learning Valued Preference for Classification (LVPC)
  - Score-matrix = fuzzy preference relation
  - Decomposition in 3 different relations
    - Strict preference
      \[ P_{ij} = r_{ij} - \min\{r_{ij}, r_{ji}\} \]
      \[ P_{ji} = r_{ji} - \min\{r_{ij}, r_{ji}\} \]
    - Conflict
      \[ C_{ij} = \min\{r_{ij}, r_{ji}\} \]
    - Ignorance
      \[ I_{ij} = 1 - \max\{r_{ij}, r_{ji}\} \]
  - Decision rule based on voting from the three relations
    \[ \text{Class} = \arg \max_{i=1, \ldots, m} \sum_{1 \leq j \neq i \leq m} P_{ij} + \frac{1}{2} C_{ij} + \frac{N_i}{N_i + N_j} I_{ij} \]
    - where \( N_i \) is the number of examples of class \( i \) in training


[Hüllermeier08, Huhn09]
State-of-the-art on aggregation for OVO

- **Non-Dominance Criterion (ND) [Fernandez09]**
  - Decision making and preference modeling [Orlovsky78]
  - Score-Matrix = preference relation
    - $r_{ji} = 1 - r_{ij}$, if not $\to$ normalize
    - Compute the maximal non-dominated elements
      - Construct the strict preference relation
      - Compute the non-dominance degree
        - the degree to which the class $i$ is dominated by no one of the remaining classes
    - Output

\[
Class = \arg \max_{i=1,\ldots,m} \{ND_i\}
\]

State-of-the-art on aggregation for OVO

- **Binary Tree of Classifiers (BTC)**
  - From Binary Tree of SVM [Fei06]
  - Reduce the number of classifiers
  - Idea: Some of the binary classifiers which discriminate between two classes
    - Also can distinguish other classes at the same time
  - Tree constructed recursively
    - Similar to DDAG
      - Each node: class list + classifier
      - More than 1 class can be deleted in each node
      - To avoid false assumptions: probability threshold for examples from other classes near the decision boundary

State-of-the-art on aggregation for OVO

- **BTC for a six class problem**
  - Classes 3 and 5 are assigned to two leaf nodes
    - Class 3 by reassignment (probability threshold)
    - Class 5 by the decision function between class 1 and 2
State-of-the-art on aggregation for OVO

- Nesting One-vs-One (NEST) [Liu07,Liu08]
  - Tries to tackle the unclassifiable produced by VOTE
  - Use VOTE
    - But if there are examples within the unclassifiable region
    - Build a new OVO system only with the examples in the region in order to make them classifiable
    - Repeat until no examples remain in the unclassifiable region
  - The convergence is proved
    - No maximum nested OVOs parameter

State-of-the-art on aggregation for OVO

- Wu, Lin and Weng Probability Estimates by Pairwise Coupling approach (PE) [Wu04]
  - Obtains the posterior probabilities
    - Starting from pairwise probabilities
  - Predicts
  - Similar to PC
    - But solving a different optimization

\[ \text{Class} = \arg \max_{i=1,\ldots,m} \hat{p}_i \]

\[
\min_p \sum_{i=1}^{m} \sum_{1 \leq j \neq i \leq m} (r_{ji}p_i - r_{ij}p_j)^2 \quad \text{subject to} \quad \sum_{i=1}^{k} p_i = 1, p_i \geq 0, \forall i.
\]
1. Introduction
2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All
3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis
4. Discussion: Lessons Learned and Future Work
5. Conclusions for OVO vs OVA
6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)
State-of-the-art on aggregation for OVA

- Starting from the score-vector

\[ R = (r_1, r_2, \ldots, r_i, \ldots, r_m) \]

- \( r_i \) = confidence of classifier in favor of class \( i \)
  - Respect to all other classes

- Usually more than 1 classifier predicts the positive class
  - Tie-breaking techniques
State-of-the-art on aggregation for OVA

- **Maximum confidence strategy (MAX)**
  - Predicts the class with the largest confidence
  \[
  \text{Class} = \arg \max_{i=1,...,m} r_i
  \]

- **Dynamically Ordered One-vs-All (DOO) [Hong08]**
  - It is not based on confidences
  - Train a Naïve Bayes classifier
    - Use its predictions to Dynamically execute each OVA
      - Predict the first class giving a positive answer
  - Ties avoided a priori by a Naïve Bayes classifier

---

Binarization strategies

- **But...**
  - Should we do binarization?
    - When it is not needed? (Ripper, C4.5, kNN...)
      - There exist previous works showing their goodness
        [Fürnkranz02, Fürnkranz03, Rifkin04]
  - Given that we want or have to use binarization...
    - How we should do it?
      - Some comparisons between OVO and OVA
        - Only for SVM [Hsu02]
      - No comparison for aggregation strategies

Outline

1. Introduction
2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All
3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis
4. Discussion: Lessons Learned and Future Work
5. Conclusions for OVO vs OVA
6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)

Experimental Framework

- Different base learners
  - Support Vector Machines (SVM)
  - C4.5 Decision Tree
  - Ripper Decision List
  - k-Nearest Neighbors (kNN)
  - Positive Definite Fuzzy Classifier (PDFC)
Experimental Framework

- **Performance measures**
  - **Accuracy rate**
    - Can be confusing evaluating multi-class problems
  - **Cohen’s kappa**
    - Takes into account random hits due to number of instances

\[
kappa = \frac{n \sum_{i=1}^{m} h_{ii} - \sum_{i=1}^{m} T_{ri} T_{ci}}{n^2 - \sum_{i=1}^{m} T_{ri} T_{ci}}
\]
## Experimental Framework

- 19 real-world Data-sets
- 5 fold-cross validation

<table>
<thead>
<tr>
<th>Data-set</th>
<th>#Ex.</th>
<th>#Atts.</th>
<th>#Num.</th>
<th>#Nom.</th>
<th>#Cl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>1728</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Lymphography</td>
<td>148</td>
<td>18</td>
<td>3</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>Vehicle</td>
<td>846</td>
<td>18</td>
<td>18</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Cleveland</td>
<td>297</td>
<td>13</td>
<td>13</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Nursery</td>
<td>1296</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Page-blocks</td>
<td>548</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Autos</td>
<td>159</td>
<td>25</td>
<td>15</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Dermatology</td>
<td>366</td>
<td>33</td>
<td>1</td>
<td>32</td>
<td>6</td>
</tr>
<tr>
<td>Flare</td>
<td>1389</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Glass</td>
<td>214</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Satimage</td>
<td>643</td>
<td>36</td>
<td>36</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Segment</td>
<td>2310</td>
<td>19</td>
<td>19</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Shuttle</td>
<td>2175</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Zoo</td>
<td>101</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>Ecoli</td>
<td>500</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Led7digit</td>
<td>1099</td>
<td>16</td>
<td>16</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Penbased</td>
<td>1484</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Yeast</td>
<td>990</td>
<td>13</td>
<td>13</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>Vowel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experimental Framework

- Algorithms parameters
  - Default configuration

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| SVM       | $C = 1.0$
|           | Tolerance Parameter = 0.001                     |
|           | Epsilon = 1.0E-12                               |
|           | Kernel Type = Polynomial                        |
|           | Polynomial Degree = 1                           |
|           | Fit Logistic Models = True                      |
| C4.5      | Prune = True                                    |
|           | Confidence level = 0.25                         |
|           | Minimum number of item-sets per leaf = 2        |
| 1NN       | $k = 1$                                         |
|           | Distance metric = Heterogeneous Value Difference Metric (HVDM) |
| 3NN       | $k = 3$                                         |
|           | Distance metric = Heterogeneous Value Difference Metric (HVDM) |
| Ripper    | Size of growing subset = 66%                   |
|           | Repetitions of the optimization stage = 2       |
| PDIFF     | $C = 100.0$                                     |
|           | Tolerance Parameter = 0.001                     |
|           | Epsilon = 1.0E-12                               |
|           | Kernel Type = Polynomial                        |
|           | Polynomial Degree = 1                           |
|           | PDIFF Type = Gaussian                           |
Experimental Framework

- **Confidence estimations**
  - **SVM**: Logistic model
    - SVM for probability estimates
  - **C4.5**: Purity of the predictor leaf
    - \( \frac{\text{N° of instances correctly classified by the leaf}}{\text{Total n° of instances in the leaf}} \)
  - **kNN**: \( \frac{1}{k} \sum_{i=1}^{k} \frac{e_i}{d_i} \)
    - Between the input pattern and the \( l^{th} \) neighbor
    - \( e_i = 1 \) if the neighbor \( i \) is from the class and 0 otherwise
  - **Ripper**: Purity of the rule
    - \( \frac{\text{N° of instances correctly classified by the rule}}{\text{Total n° of instances in the rule}} \)
  - **PDFC**: confidence = 1 is given for the predicted class
Outline

1. Introduction
2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All
3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis
4. Discussion: Lessons Learned and Future Work
5. Conclusions for OVO vs OVA
6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)
   - Difficult Classes Problem in OVO Strategy

Experimental Study

- Average accuracy and kappa results

<table>
<thead>
<tr>
<th>Method Aggregation</th>
<th>SVM</th>
<th>C4.5</th>
<th>1NN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc&lt;br&gt;talkt&lt;br&gt;</td>
<td>Avg. Rank</td>
<td>Acc&lt;br&gt;talkt&lt;br&gt;</td>
</tr>
<tr>
<td>Base</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VOTE</td>
<td>81.14 ± 3.22</td>
<td>4.37 (1)</td>
<td>81.57 ± 3.29</td>
</tr>
<tr>
<td>WV</td>
<td>81.05 ± 2.92</td>
<td>5.08 (5)</td>
<td>81.50 ± 3.28</td>
</tr>
<tr>
<td>DDAG</td>
<td>81.01 ± 3.28</td>
<td>5.39 (8)</td>
<td>81.02 ± 3.56</td>
</tr>
<tr>
<td>PC</td>
<td>81.08 ± 2.89</td>
<td>5.29 (7)</td>
<td>81.49 ± 3.82</td>
</tr>
<tr>
<td>OVO</td>
<td>LVPC</td>
<td>81.14 ± 3.11</td>
<td>4.50 (3)</td>
</tr>
<tr>
<td>ND</td>
<td>81.01 ± 3.15</td>
<td>4.92 (5)</td>
<td>81.12 ± 3.24</td>
</tr>
<tr>
<td>BTC</td>
<td>80.82 ± 3.24</td>
<td>6.18 (9)</td>
<td>81.22 ± 3.87</td>
</tr>
<tr>
<td>NEST</td>
<td>81.14 ± 3.36</td>
<td>4.47 (2)</td>
<td>81.20 ± 3.47</td>
</tr>
<tr>
<td>PE</td>
<td>81.03 ± 3.35</td>
<td>4.79 (9)</td>
<td>81.42 ± 3.22</td>
</tr>
<tr>
<td>OVA</td>
<td>MAX</td>
<td>78.76 ± 3.00</td>
<td>1.53 (2)</td>
</tr>
<tr>
<td></td>
<td>DOO</td>
<td>78.75 ± 3.15</td>
<td>1.47 (1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Aggregation</th>
<th>SVM</th>
<th>C4.5</th>
<th>1NN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Kappa&lt;br&gt;talkt&lt;br&gt;</td>
<td>Avg. Rank</td>
<td>Kappa&lt;br&gt;talkt&lt;br&gt;</td>
</tr>
<tr>
<td>Base</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VOTE</td>
<td>.7283 ± .0548</td>
<td>4.82 (2)</td>
<td>.7383 ± .0490</td>
</tr>
<tr>
<td>WV</td>
<td>.7229 ± .0506</td>
<td>5.05 (6)</td>
<td>.7348 ± .0485</td>
</tr>
<tr>
<td>DDAG</td>
<td>.7230 ± .0555</td>
<td>5.11 (7)</td>
<td>.7304 ± .0535</td>
</tr>
<tr>
<td>PC</td>
<td>.7234 ± .0520</td>
<td>5.18 (8)</td>
<td>.7341 ± .0493</td>
</tr>
<tr>
<td>OVO</td>
<td>LVPC</td>
<td>.7211 ± .0531</td>
<td>5.03 (5)</td>
</tr>
<tr>
<td>ND</td>
<td>.7205 ± .0582</td>
<td>4.82 (3)</td>
<td>.7266 ± .0489</td>
</tr>
<tr>
<td>BTC</td>
<td>.7204 ± .0551</td>
<td>6.05 (9)</td>
<td>.7297 ± .0428</td>
</tr>
<tr>
<td>NEST</td>
<td>.7243 ± .0559</td>
<td>4.03 (1)</td>
<td>.7291 ± .0514</td>
</tr>
<tr>
<td>PE</td>
<td>.7228 ± .0537</td>
<td>4.92 (4)</td>
<td>.7330 ± .0480</td>
</tr>
<tr>
<td>OVA</td>
<td>MAX</td>
<td>.6868 ± .0553</td>
<td>1.55 (2)</td>
</tr>
<tr>
<td></td>
<td>DOO</td>
<td>.6868 ± .0565</td>
<td>1.45 (1)</td>
</tr>
</tbody>
</table>
Experimental Study

- Average accuracy and kappa results

<table>
<thead>
<tr>
<th>Method</th>
<th>Aggregation</th>
<th>Base</th>
<th>3NN</th>
<th>Ripper</th>
<th>PDCF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Acctst</td>
<td>Avg. Rank</td>
<td>Acctst</td>
</tr>
<tr>
<td>VOTE</td>
<td></td>
<td></td>
<td>83.00 ± 2.92</td>
<td>5.05 (8)</td>
<td>80.57 ± 3.17</td>
</tr>
<tr>
<td>WV</td>
<td></td>
<td></td>
<td>83.11 ± 2.87</td>
<td>4.47 (3)</td>
<td>80.54 ± 3.03</td>
</tr>
<tr>
<td>DDAG</td>
<td></td>
<td></td>
<td>82.73 ± 2.83</td>
<td>3.87 (8)</td>
<td>77.62 ± 3.61</td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td></td>
<td>83.00 ± 2.96</td>
<td>5.11 (7)</td>
<td>80.33 ± 3.30</td>
</tr>
<tr>
<td>LVPC</td>
<td></td>
<td></td>
<td>83.07 ± 2.79</td>
<td>4.81 (4)</td>
<td>80.58 ± 3.16</td>
</tr>
<tr>
<td>ND</td>
<td></td>
<td></td>
<td>83.07 ± 2.93</td>
<td>4.29 (1)</td>
<td>79.38 ± 3.27</td>
</tr>
<tr>
<td>BTC</td>
<td></td>
<td></td>
<td>82.99 ± 2.98</td>
<td>5.00 (3)</td>
<td>70.19 ± 3.07</td>
</tr>
<tr>
<td>NEST</td>
<td></td>
<td></td>
<td>82.67 ± 2.94</td>
<td>4.16 (9)</td>
<td>80.01 ± 3.50</td>
</tr>
<tr>
<td>PE</td>
<td></td>
<td></td>
<td>83.11 ± 2.91</td>
<td>4.45 (2)</td>
<td>80.07 ± 3.08</td>
</tr>
<tr>
<td>MAX</td>
<td></td>
<td></td>
<td>82.75 ± 4.29</td>
<td>1.58 (2)</td>
<td>73.30 ± 4.94</td>
</tr>
<tr>
<td>DOO</td>
<td></td>
<td></td>
<td>82.76 ± 4.38</td>
<td>1.42 (1)</td>
<td>70.12 ± 4.87</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Aggregation</th>
<th>Base</th>
<th>3NN</th>
<th>Ripper</th>
<th>PDCF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Kappattst</td>
<td>Avg. Rank</td>
<td>Kappattst</td>
</tr>
<tr>
<td>VOTE</td>
<td></td>
<td></td>
<td>.7507 ± .0500</td>
<td>5.03 (8)</td>
<td>.7250 ± .0476</td>
</tr>
<tr>
<td>WV</td>
<td></td>
<td></td>
<td>.7519 ± .0487</td>
<td>4.17 (3)</td>
<td>.7249 ± .0455</td>
</tr>
<tr>
<td>DDAG</td>
<td></td>
<td></td>
<td>.7497 ± .0487</td>
<td>5.87 (8)</td>
<td>.6976 ± .0489</td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td></td>
<td>.7505 ± .0505</td>
<td>4.89 (5)</td>
<td>.7274 ± .0483</td>
</tr>
<tr>
<td>LVPC</td>
<td></td>
<td></td>
<td>.7496 ± .0475</td>
<td>5.18 (7)</td>
<td>.7246 ± .0469</td>
</tr>
<tr>
<td>ND</td>
<td></td>
<td></td>
<td>.7524 ± .0500</td>
<td>4.03 (1)</td>
<td>.7098 ± .0479</td>
</tr>
<tr>
<td>BTC</td>
<td></td>
<td></td>
<td>.7519 ± .0514</td>
<td>4.87 (4)</td>
<td>.7087 ± .0476</td>
</tr>
<tr>
<td>NEST</td>
<td></td>
<td></td>
<td>.7461 ± .0505</td>
<td>6.24 (9)</td>
<td>.7195 ± .0496</td>
</tr>
<tr>
<td>PE</td>
<td></td>
<td></td>
<td>.7526 ± .0499</td>
<td>1.18 (2)</td>
<td>.7193 ± .0457</td>
</tr>
<tr>
<td>MAX</td>
<td></td>
<td></td>
<td>.7481 ± .0695</td>
<td>1.58 (2)</td>
<td>.6896 ± .0713</td>
</tr>
<tr>
<td>DOO</td>
<td></td>
<td></td>
<td>.7473 ± .0710</td>
<td>1.42 (1)</td>
<td>.7004 ± .0716</td>
</tr>
</tbody>
</table>
Which is the most appropriate aggregation?

- **OVO aggregations Analysis**
  - **SVM:** NEST and VOTE, but no statistical differences
  - **C4.5:** Statistical differences
    - WV, LVPC and PC the most robust
    - NEST and DDAG the weakest
  - **1NN:** Statistical differences
    - PC and PE the best $\rightarrow$ confidences in \{0,1\}
      - In PDFC they also excel
    - ND the worst $\rightarrow$ poor confidences, excessive ties
Which is the most appropriate aggregation?

- **OVO aggregations Analysis**
  - **3NN**: No significant differences
    - ND stands out
  - **Ripper**: Statistical differences
    - WV and LVPC vs. BTC and DDAG
  - **PDFC**: No significant differences (low p-value in kappa)
    - VOTE, PC and PE overall better performance
Which is the most appropriate aggregation?

- OVA aggregations Analysis
  - DOO performs better when the base classifiers accuracy is not better than the Naïve Bayes ones.
  - It helps selecting the most appropriate classifier to use dynamically
  - In other cases, it can distort the results

<table>
<thead>
<tr>
<th>Base Classifier</th>
<th>Measure</th>
<th>$R^+$</th>
<th>$R^-$</th>
<th>Hypothesis ($\alpha = 0.05$)</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>Accuracy</td>
<td>82</td>
<td>108</td>
<td>Not Rejected</td>
<td>0.53213</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>86</td>
<td>104</td>
<td>Not Rejected</td>
<td>0.75637</td>
</tr>
<tr>
<td>C4.5</td>
<td>Accuracy</td>
<td>14</td>
<td>176</td>
<td>Rejected for DOC</td>
<td>0.00179</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>11.5</td>
<td>178.5</td>
<td>Rejected for DOC</td>
<td>0.00118</td>
</tr>
<tr>
<td>1NN</td>
<td>Accuracy</td>
<td>55</td>
<td>135</td>
<td>Not Rejected</td>
<td>0.09097</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>55</td>
<td>135</td>
<td>Not Rejected</td>
<td>0.09097</td>
</tr>
<tr>
<td>2NN</td>
<td>Accuracy</td>
<td>75</td>
<td>115</td>
<td>Not Rejected</td>
<td>0.73532</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>75</td>
<td>114</td>
<td>Not Rejected</td>
<td>0.86577</td>
</tr>
<tr>
<td>Ripper</td>
<td>Accuracy</td>
<td>44.5</td>
<td>145.5</td>
<td>Rejected for DOC</td>
<td>0.04286</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>42</td>
<td>148</td>
<td>Rejected for DOC</td>
<td>0.03294</td>
</tr>
<tr>
<td>PDFC</td>
<td>Accuracy</td>
<td>130.5</td>
<td>59.5</td>
<td>Rejected for MAX</td>
<td>0.0464</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>138</td>
<td>52</td>
<td>Rejected for MAX</td>
<td>0.62799</td>
</tr>
</tbody>
</table>
Should we do binarization?
How should we do it?

- Representatives of OVO and OVA
- By the previous analysis
- Average results
Should we do binarization?
How should we do it?

- Rankings within each classifier
  - In general, OVO is the most competitive
Should we do binarization?
How should we do it?

- Box plots for test results
  - OVA reduce performance in kappa
  - OVO is more compact (hence, robust)
Should we do binarization? How should we do it?

- **Statistical analysis**
  - SVM and PDFC
    - OVO outperforms OVA with significant differences
  - C4.5, 1NN, 3NN and Ripper
    - P-values returned by Iman-Davenport tests (* if rejected)

<table>
<thead>
<tr>
<th>Base Classifier</th>
<th>Comparison</th>
<th>Measure</th>
<th>$R^1$</th>
<th>$R^2$</th>
<th>Hypothesis (α = 0.05)</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>VESTemo vs. DOOemo</td>
<td>Accuracy</td>
<td>163</td>
<td>37</td>
<td>Rejected for VESTemo</td>
<td>0.01969</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kappa</td>
<td>166</td>
<td>43</td>
<td>Rejected for VESTemo</td>
<td>0.0141</td>
</tr>
<tr>
<td>PDFC</td>
<td>PCmob vs. MAXmob</td>
<td>Accuracy</td>
<td>146</td>
<td>44</td>
<td>Rejected for PCmob</td>
<td>0.04614</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kappa</td>
<td>147</td>
<td>43</td>
<td>Rejected for PCmob</td>
<td>0.00329</td>
</tr>
</tbody>
</table>

- kNN, no statistical differences, but also not worse results
Should we do binarization?
How should we do it?

- Statistical analysis
  - C4.5
    - WV for OVO outperforms the rest
  - Ripper
    - WV for OVO is the best
      - No statistical differences with OVA
        - But OVO differs statistically from Ripper while OVA do not
1. Introduction
2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All
3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis
4. Discussion: Lessons Learned and Future Work
5. Conclusions for OVO vs OVA
6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)

Discussion

- Lessons learned
  - Binarization is beneficial
    - Also when the problem can be tackled without it
  - The most robust aggregations for OVO
    - WV, LVPC, PC and PE
  - The most robust aggregations for OVA
    - Not clear
    - Need more attention, can be improved
  - Too many approaches to deal with the unclassifiable region in OVO (NEST, BTC, DDAG)
Discussion

- **Lessons learned**
  - **OVA problem**
    - Imbalanced data-sets
    - Not against Rifkin’s findings
      - But, this means that OVA are less robust
        - Need more fine-tuned base classifiers
  - **Importance of confidence estimates of base classifiers**
  - **Scalability**
    - Number of classes: OVO seems to work better
    - Number of instances: OVO natures make it more adequate
Discussion

- **Future work**
  - Detection of non-competent examples
  - Techniques for imbalanced data-sets
  - Studies on scalability
  - OVO as a decision making problem
    - Suppose inaccurate or erroneous base classifiers
  - New combinations for OVA
    - Something more than a tie-breaking technique
  - Data-complexity measures
    - A priori knowledge extraction to select the proper mechanism
Outline

1. Introduction
2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All
3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis
4. Discussion: Lessons Learned and Future Work
5. Conclusions for OVO vs OVA
6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)

Conclusions

- **Goodness of using binarization**
  - Concretely, OVO approach
    - WV, LVPC, PC and PE
    - The aggregation is base learner dependant

- **Low attention to OVA strategy**
  - Problems with imbalanced data

- **Importance of confidence estimates**

- **Many work remind to be addressed**

---

Outline

1. Introduction
2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All
3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis
4. Discussion: Lessons Learned and Future Work
5. Conclusions for OVO vs OVA
6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)

Dynamic OVO: Avoiding Non-competence

- **Non-Competent Classifiers:**
  - Those whose output is not relevant for the classification of the query instance
  - They have not been trained with instances of the real class of the example to be classified

- Classify $x$, whose real class is $c_1$

R(x) =

$$
\begin{pmatrix}
  c_1 & c_2 & c_3 & c_4 & c_5 \\
  c_1 & - & 0,55 & 0,6 & 0,75 & 0,7 \\
  c_2 & 0,45 & - & 0,4 & 1 & 0,8 \\
  c_3 & 0,4 & 0,6 & - & 0,5 & 0,4 \\
  c_4 & 0,25 & 0,0 & 0,5 & - & 0,1 \\
  c_5 & 0,30 & 0,2 & 0,6 & 0,9 & - \\
\end{pmatrix}
$$
Dynamic OVO: Avoiding Non-competence

- Non-Competent Classifiers:
  - Consider WV aggregation, $c_2$ is predicted
  - None of the classifiers considering $c_1$ failed
  - Non-competent classifiers strongly voted for $c_2$

$$R(x) = \begin{pmatrix}
  c1 & c2 & c3 & c4 & c5 & WV \\
  c1 & - & 0.55 & 0.6 & 0.75 & 0.7 & 2.6 \\
  c2 & 0.45 & - & 0.4 & 1 & 0.8 & 2.65 \\
  c3 & 0.4 & 0.6 & - & 0.5 & 0.4 & 1.9 \\
  c4 & 0.25 & 0.0 & 0.5 & - & 0.1 & 0.85 \\
  c5 & 0.30 & 0.2 & 0.6 & 0.9 & - & 2.1 \\
\end{pmatrix}$$
Dynamic OVO: Avoiding Non-competence

- Dynamic Classifier Selection:
  - Classifiers specialized in different areas of the input space
  - Classifiers *complement* themselves
  - The most competent one for the instance is selected:
    - Instead of combining them all
    - Assuming that several misses can be done (they are corrected)
Dynamic OVO: Avoiding Non-competence

- Avoiding non-competence problem
- Adapting Dynamic Classifier Selection (DCS) to OVO
  - Baseline classifiers competent over their pair of classes
- **Search for a lower set of classes** than those that probably the instance belongs to.
  - Remove those (probably) non-competent classifiers
  - Avoid misclassifications
- **Neighbourhood of the instance** is considered\([Woods97]\)
  - Local precisions cannot be estimated
  - **Classes in the neighbourhood** \(\rightarrow\) reduced score matrix

Dynamic OVO: Avoiding Non-competence

**DCS ALGORITHM FOR OVO STRATEGY**

1. Compute the $k$ nearest neighbors of the instance ($k = 3 \cdot m$)
2. Select the classes in the neighborhood (if it is unique $k+$)
3. Consider the subset of classes in the reduced-score matrix

- **Any** existing OVO aggregation can be used
- Difficult to misclassify instances
- $k$ value is larger than the usual value for classification

```python
Algorithm 1 Dynamic Classifier Selection for OVO scheme

1: procedure DYNAMIC OVO($v, R$)
2: \hspace{1cm} $k = 3 \cdot m \quad \triangleright m$ is the number of classes
3: \hspace{1cm} repeat
4: \hspace{2cm} Neighbours \leftarrow kNN(v)
5: \hspace{2cm} $C \leftarrow \text{Classes}(\text{Neighbours}) \quad \triangleright$ We select the class labels in the neighbourhood
6: \hspace{2cm} $k++$
7: \hspace{1cm} until $\#C > 1$ or $k = 6 \cdot m$
8: \hspace{2cm} if $C > 1$ then
9: \hspace{3cm} $R' \leftarrow [R - \text{rows}(i), \text{cols}(i)]; \ i \neq C$
10: \hspace{3cm} return $R'$ \hspace{0.2em} \triangleright A subset of the score matrix
11: \hspace{2cm} else
12: \hspace{3cm} return $R$ \hspace{0.2em} \triangleright Standard OVO approach
13: \hspace{1cm} end if
14: end procedure
```
Dynamic OVO: Avoiding Non-competence

- Classify $x$, whose real class is $c_1$

\[ R(x) = \begin{pmatrix}
    c1 & c2 & c3 & c4 & c5 \\
    c1 & -   & 0.55 & 0.6 & 0.75 & 0.7 \\
    c2 & 0.45 & -   & 0.4 & 1    & 0.8 \\
    c3 & 0.4  & 0.6  & -   & 0.5  & 0.4 \\
    c4 & 0.25 & 0.0  & 0.5  & -    & 0.1 \\
    c5 & 0.30 & 0.2  & 0.6  & 0.9  & - 
\end{pmatrix} \]
Dynamic OVO: Avoiding Non-competence

- Consider WV aggregation, $c_2$ is predicted
- None of the classifiers considering $c_1$ failed
- Non-competent classifiers strongly voted for $c_2$

$$R(x) = \begin{pmatrix} c1 & c2 & c3 & c4 & c5 & WV \\ c1 & - & 0.55 & 0.6 & 0.75 & 0.7 & 2.6 \\ c2 & 0.45 & - & 0.4 & 1 & 0.8 & 2.65 \\ c3 & 0.4 & 0.6 & - & 0.5 & 0.4 & 1.9 \\ c4 & 0.25 & 0.0 & 0.5 & - & 0.1 & 0.85 \\ c5 & 0.30 & 0.2 & 0.6 & 0.9 & - & 2.1 \end{pmatrix}$$
Dynamic OVO: Avoiding Non-competence

- Applying Dynamic kNN
  - Compute the kNN of x (k = 3 \cdot 5 = 15)
  - Subset of classes = \{c_1, c_4, c_5\}
  - Remove \{c_2, c_3\} from the score-matrix
  - Apply WV to the reduced score-matrix

\[
R_{\text{dyn}}(x) = \begin{pmatrix}
  c_1 & c_2 & c_3 & c_4 & c_5 & \text{WV} \\
  c_1 & - & 0.55 & 0.6 & 0.75 & 0.7 \\
  c_2 & 0.45 & - & 0.4 & 1 & 0.8 \\
  c_3 & 0.4 & 0.6 & - & 0.5 & 0.4 \\
  c_4 & 0.25 & 0.0 & 0.5 & - & 0.1 \\
  c_5 & 0.30 & 0.2 & 0.6 & 0.9 & -
\end{pmatrix}
\]
Dynamic OVO: Avoiding Non-competence

Summary:
- We avoid some of the non-competent classifiers by DCS
- It is simple, yet powerful
- Positive synergy between Dynamic OVO and WV
- All the differences are due to the aggregations
  - Tested with same score-matrices in all methods
  - Significant differences only changing the aggregation

1. Introduction

2. Binarization
   - Decomposition strategies (One-vs-One, One-vs-All and Others)
   - State-of-the-art on Aggregations
     - One-vs-One
     - One-vs-All

3. Experimental Study
   - Experimental Framework
   - Results and Statistical Analysis

4. Discussion: Lessons Learned and Future Work

5. Conclusions for OVO vs OVA

6. Novel Approaches for the One-vs-One Learning Scheme
   - Dynamic OVO: Avoiding Non-competence
   - Distance-based Relative Competence Weighting Approach (DRCW-OVO)

Non-Competent Classifiers:
- Those whose output is not relevant for the classification of the query instance
- They have not been trained with instances of the real class of the example to be classified

\[ R(x) = \begin{pmatrix}
  c_1 & c_2 & c_3 & c_4 & c_5 \\
  c_1 & 0.55 & 0.45 & 0.80 & 0.90 \\
  c_2 & 0.45 & 0.55 & 1.00 & 0.80 \\
  c_3 & 0.55 & 0.45 & 0.45 & 0.40 \\
  c_4 & 0.20 & 0.00 & 0.55 & 0.10 \\
  c_5 & 0.10 & 0.20 & 0.60 & 0.90 & -
\end{pmatrix} \]
Distance-based Relative Competence Weighting Approach

- **Non-Competent Classifiers:**

  - Consider WV aggregation, $c_2$ is predicted
  - **None** of the classifiers **considering** $c_1$ **failed**
  - **Non-competent** classifiers **strongly voted for** $c_2$

$$R(x) = \begin{pmatrix}
  c_1 & c_2 & c_3 & c_4 & c_5 \\
  c_1 & - & 0.55 & 0.45 & 0.80 & 0.90 & 3 & 2.70 \\
  c_2 & 0.45 & - & 0.55 & 1.00 & 0.80 & 3 & 2.80 \\
  c_3 & 0.55 & 0.45 & - & 0.45 & 0.40 & 1 & 1.85 \\
  c_4 & 0.20 & 0.00 & 0.55 & - & 0.10 & 1 & 0.85 \\
  c_5 & 0.10 & 0.20 & 0.60 & 0.90 & - & 2 & 1.80
\end{pmatrix}$$
Distance-based Relative Competence Weighting Approach

- Designed to address the non-competence classifier problem
- It carries out a dynamic adaptation of the score-matrix
  - More competent classifiers should be those whose pair of classes are “nearer” to the query instance.
  - Confidence degrees are weighted in accordance to the former distance.
- This distance is computed by using the standard kNN approach
Distance-based Relative Competence Weighting Approach

DRCW ALGORITHM FOR OVO STRATEGY

1. Compute the k nearest neighbors of each class for the given instance and store the average distances of the k neighbors of each class in a vector \( d = (d_1, \ldots, d_m) \).

2. A new score-matrix \( R^w \) is created where the output \( r_{ij} \) of a classifier distinguishing classes \( i, j \) are weighted as follows,

\[
    r^w_{ij} = r_{ij} \cdot w_{ij},
\]

where \( w_{ij} \) is the relative competence of the classifier on the corresponding output computed as

\[
    w_{ij} = \frac{r_{ij}^2}{d_i^2 + d_j^2},
\]

being \( d_i \) the distance of the instance to the nearest neighbor of class \( i \).

3. Use weighted voting strategy on the modified score-matrix \( R^w \) to obtain the final class.

\[
    \text{Class} = \arg \max_{i=1, \ldots, m} \sum_{1 \leq j \neq i \leq m} r_{ij} \cdot w_{ij}
\]

Distance is computed with respect to all classes:

- \( k \cdot m \) neighbors are used
- \( k = 1 \) is not the same as using 1NN classifier

With \( k = 1 \) a neighbor for each class is obtained, therefore it would use the \( m \) neighbours (1 per class). Next experimental example use \( k=5 \).
Distance-based Relative Competence Weighting Approach

- Classify $x$, whose real class is $c_1$

$$R(x) = \begin{pmatrix}
   c_1 & c_2 & c_3 & c_4 & c_5 \\
   c_1 & -   & 0.55 & 0.45 & 0.80 & 0.90 \\
   c_2 & 0.45 & -   & 0.55 & 1.00 & 0.80 \\
   c_3 & 0.55 & 0.45 & -   & 0.45 & 0.40 \\
   c_4 & 0.20 & 0.00 & 0.55 & -   & 0.10 \\
   c_5 & 0.10 & 0.20 & 0.60 & 0.90 & -
\end{pmatrix}$$
Distance-based Relative Competence Weighting Approach

- Distances to k nearest neighbors of each class ($d$) are computed: $d = (0.8, 0.9, 0.6, 1.2, 1.6)$
- A Weight-matrix $W$ is computed to represent all $w_{ij}$

$$W(x) = \begin{pmatrix}
  c1 & c2 & c3 & c4 & c5 \\
  c1 & - & 0.56 & 0.36 & 0.69 & 0.80 \\
  c2 & 0.44 & - & 0.31 & 0.64 & 0.76 \\
  c3 & 0.64 & 0.69 & - & 0.80 & 0.88 \\
  c4 & 0.31 & 0.36 & 0.5 & - & 0.64 \\
  c5 & 0.20 & 0.24 & 0.6 & 0.36 & - 
\end{pmatrix}$$
Distance-based Relative Competence Weighting Approach

- Apply the weight-matrix $W$ to the score-matrix $R$
- $WV$ is applied to obtain the predicted class in DRCW-OVO

$$R^W(x) = \begin{pmatrix}
  c1 & 0,31 & 0,16 & 0,55 & 0,72 & 1,74 \\
  c2 & 0,20 & 0,17 & 0,64 & 0,61 & 1,66 \\
  c3 & 0,35 & 0,31 & 0,36 & 0,35 & 1,37 \\
  c4 & 0,06 & 0,00 & 0,11 & 0,06 & 0,24 \\
  c5 & 0,02 & 0,05 & 0,07 & 0,32 & 0,47 \\
\end{pmatrix}$$
Distance-based Relative Competence Weighting Approach

- Experimental Analysis

Table 8: Average accuracy results in test of the representative combinations, DCS method and DRCW-OVO method (with k=5) for each base classifier.

<table>
<thead>
<tr>
<th>Data-set</th>
<th>UT5</th>
<th>DCS</th>
<th>DRCW</th>
<th>SVM_row</th>
<th>DCS</th>
<th>DRCW</th>
<th>SVM_end</th>
<th>DCS</th>
<th>DRCW</th>
<th>3NN</th>
<th>DCS</th>
<th>DRCW</th>
<th>PDHC</th>
<th>DCS</th>
<th>DRCW</th>
<th>Ripper</th>
<th>DCS</th>
<th>DRCW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autos</td>
<td>75.24</td>
<td>74.96</td>
<td>80.96</td>
<td>73.75</td>
<td>73.81</td>
<td>70.48</td>
<td>69.02</td>
<td>70.27</td>
<td>71.45</td>
<td>78.88</td>
<td>76.96</td>
<td>75.14</td>
<td>78.82</td>
<td>79.40</td>
<td>80.74</td>
<td>85.09</td>
<td>84.42</td>
<td>84.58</td>
</tr>
<tr>
<td>Car</td>
<td>91.68</td>
<td>91.50</td>
<td>95.99</td>
<td>93.58</td>
<td>93.58</td>
<td>97.16</td>
<td>64.99</td>
<td>84.84</td>
<td>81.66</td>
<td>93.57</td>
<td>93.40</td>
<td>96.93</td>
<td>99.77</td>
<td>99.88</td>
<td>99.42</td>
<td>92.59</td>
<td>93.52</td>
<td>96.35</td>
</tr>
<tr>
<td>Cleveland</td>
<td>52.55</td>
<td>53.55</td>
<td>55.23</td>
<td>58.97</td>
<td>59.31</td>
<td>58.65</td>
<td>47.53</td>
<td>47.87</td>
<td>48.88</td>
<td>58.31</td>
<td>57.96</td>
<td>56.61</td>
<td>53.92</td>
<td>55.03</td>
<td>56.61</td>
<td>52.18</td>
<td>54.54</td>
<td>56.90</td>
</tr>
<tr>
<td>Dermatology</td>
<td>95.24</td>
<td>93.32</td>
<td>93.05</td>
<td>94.71</td>
<td>94.99</td>
<td>95.55</td>
<td>97.20</td>
<td>97.20</td>
<td>97.48</td>
<td>92.14</td>
<td>95.49</td>
<td>96.90</td>
<td>84.66</td>
<td>93.85</td>
<td>91.90</td>
<td>93.32</td>
<td>94.43</td>
<td>95.27</td>
</tr>
<tr>
<td>Ecoli</td>
<td>81.05</td>
<td>81.04</td>
<td>85.58</td>
<td>70.37</td>
<td>79.63</td>
<td>82.25</td>
<td>77.11</td>
<td>77.11</td>
<td>81.64</td>
<td>81.66</td>
<td>82.52</td>
<td>84.30</td>
<td>84.07</td>
<td>83.78</td>
<td>84.68</td>
<td>78.47</td>
<td>78.74</td>
<td>82.34</td>
</tr>
<tr>
<td>Flame</td>
<td>75.34</td>
<td>74.62</td>
<td>75.27</td>
<td>75.43</td>
<td>75.46</td>
<td>75.85</td>
<td>69.28</td>
<td>73.29</td>
<td>72.01</td>
<td>71.21</td>
<td>71.59</td>
<td>72.43</td>
<td>73.64</td>
<td>73.92</td>
<td>73.69</td>
<td>75.24</td>
<td>74.83</td>
<td>75.60</td>
</tr>
<tr>
<td>Glass</td>
<td>72.03</td>
<td>71.63</td>
<td>74.81</td>
<td>62.14</td>
<td>63.14</td>
<td>71.04</td>
<td>73.72</td>
<td>74.15</td>
<td>76.19</td>
<td>73.35</td>
<td>74.27</td>
<td>74.33</td>
<td>68.72</td>
<td>70.12</td>
<td>70.12</td>
<td>68.56</td>
<td>68.42</td>
<td>75.40</td>
</tr>
<tr>
<td>Led7digit</td>
<td>64.51</td>
<td>63.35</td>
<td>63.33</td>
<td>67.90</td>
<td>68.09</td>
<td>63.47</td>
<td>61.33</td>
<td>61.57</td>
<td>62.54</td>
<td>66.08</td>
<td>67.88</td>
<td>68.26</td>
<td>62.17</td>
<td>62.60</td>
<td>65.42</td>
<td>63.98</td>
<td>63.06</td>
<td>64.19</td>
</tr>
<tr>
<td>Lymphography</td>
<td>74.50</td>
<td>76.44</td>
<td>76.44</td>
<td>82.48</td>
<td>82.48</td>
<td>83.10</td>
<td>81.87</td>
<td>81.87</td>
<td>82.50</td>
<td>68.19</td>
<td>79.55</td>
<td>79.52</td>
<td>83.19</td>
<td>83.19</td>
<td>83.19</td>
<td>76.58</td>
<td>76.68</td>
<td>77.04</td>
</tr>
<tr>
<td>Nursery</td>
<td>89.66</td>
<td>89.81</td>
<td>90.90</td>
<td>92.13</td>
<td>92.13</td>
<td>91.53</td>
<td>80.33</td>
<td>89.05</td>
<td>90.83</td>
<td>93.29</td>
<td>93.29</td>
<td>93.68</td>
<td>97.92</td>
<td>97.92</td>
<td>97.84</td>
<td>90.66</td>
<td>90.81</td>
<td>92.44</td>
</tr>
<tr>
<td>Pageblocks</td>
<td>95.64</td>
<td>95.46</td>
<td>95.82</td>
<td>94.90</td>
<td>94.53</td>
<td>95.27</td>
<td>94.58</td>
<td>94.76</td>
<td>95.11</td>
<td>95.63</td>
<td>95.46</td>
<td>95.09</td>
<td>95.09</td>
<td>95.09</td>
<td>95.09</td>
<td>95.45</td>
<td>95.11</td>
<td>96.00</td>
</tr>
<tr>
<td>Penbased</td>
<td>91.10</td>
<td>91.11</td>
<td>93.64</td>
<td>93.92</td>
<td>96.01</td>
<td>97.01</td>
<td>97.55</td>
<td>97.64</td>
<td>98.00</td>
<td>97.00</td>
<td>96.64</td>
<td>96.91</td>
<td>98.19</td>
<td>98.10</td>
<td>98.10</td>
<td>91.38</td>
<td>91.63</td>
<td>96.01</td>
</tr>
<tr>
<td>Sattimage</td>
<td>82.15</td>
<td>82.02</td>
<td>85.41</td>
<td>84.43</td>
<td>84.16</td>
<td>85.34</td>
<td>84.77</td>
<td>85.70</td>
<td>87.56</td>
<td>87.58</td>
<td>87.73</td>
<td>89.34</td>
<td>86.79</td>
<td>86.05</td>
<td>87.25</td>
<td>82.61</td>
<td>82.14</td>
<td>86.01</td>
</tr>
<tr>
<td>Segment</td>
<td>94.48</td>
<td>94.31</td>
<td>97.07</td>
<td>92.68</td>
<td>93.29</td>
<td>95.58</td>
<td>92.23</td>
<td>93.76</td>
<td>94.70</td>
<td>96.58</td>
<td>96.80</td>
<td>96.81</td>
<td>95.32</td>
<td>97.36</td>
<td>97.27</td>
<td>96.58</td>
<td>96.28</td>
<td>99.54</td>
</tr>
<tr>
<td>Vehicle</td>
<td>72.33</td>
<td>72.81</td>
<td>73.88</td>
<td>73.53</td>
<td>74.00</td>
<td>74.43</td>
<td>81.92</td>
<td>81.92</td>
<td>82.04</td>
<td>72.11</td>
<td>72.23</td>
<td>72.23</td>
<td>84.53</td>
<td>84.40</td>
<td>84.41</td>
<td>68.27</td>
<td>70.20</td>
<td>71.29</td>
</tr>
<tr>
<td>Vowel</td>
<td>83.43</td>
<td>83.64</td>
<td>94.75</td>
<td>71.41</td>
<td>71.82</td>
<td>95.05</td>
<td>99.70</td>
<td>99.70</td>
<td>99.20</td>
<td>97.78</td>
<td>97.37</td>
<td>97.27</td>
<td>88.28</td>
<td>98.08</td>
<td>98.59</td>
<td>89.20</td>
<td>92.59</td>
<td>94.44</td>
</tr>
<tr>
<td>Yeast</td>
<td>59.57</td>
<td>59.81</td>
<td>60.46</td>
<td>60.52</td>
<td>59.98</td>
<td>60.92</td>
<td>59.31</td>
<td>59.51</td>
<td>62.14</td>
<td>56.68</td>
<td>56.54</td>
<td>58.30</td>
<td>60.25</td>
<td>59.98</td>
<td>60.92</td>
<td>58.30</td>
<td>58.10</td>
<td>61.81</td>
</tr>
<tr>
<td>Zoo</td>
<td>92.17</td>
<td>92.17</td>
<td>93.22</td>
<td>95.72</td>
<td>95.72</td>
<td>96.77</td>
<td>78.06</td>
<td>84.13</td>
<td>90.30</td>
<td>89.50</td>
<td>91.86</td>
<td>94.63</td>
<td>96.77</td>
<td>96.77</td>
<td>97.82</td>
<td>94.05</td>
<td>94.05</td>
<td>96.10</td>
</tr>
<tr>
<td>Average</td>
<td>81.48</td>
<td>81.81</td>
<td>84.02</td>
<td>81.38</td>
<td>81.55</td>
<td>84.48</td>
<td>79.74</td>
<td>81.98</td>
<td>83.01</td>
<td>82.63</td>
<td>83.52</td>
<td>84.06</td>
<td>84.29</td>
<td>85.01</td>
<td>85.36</td>
<td>81.21</td>
<td>81.35</td>
<td>84.17</td>
</tr>
</tbody>
</table>
Questions?

- Thank you for your attention!