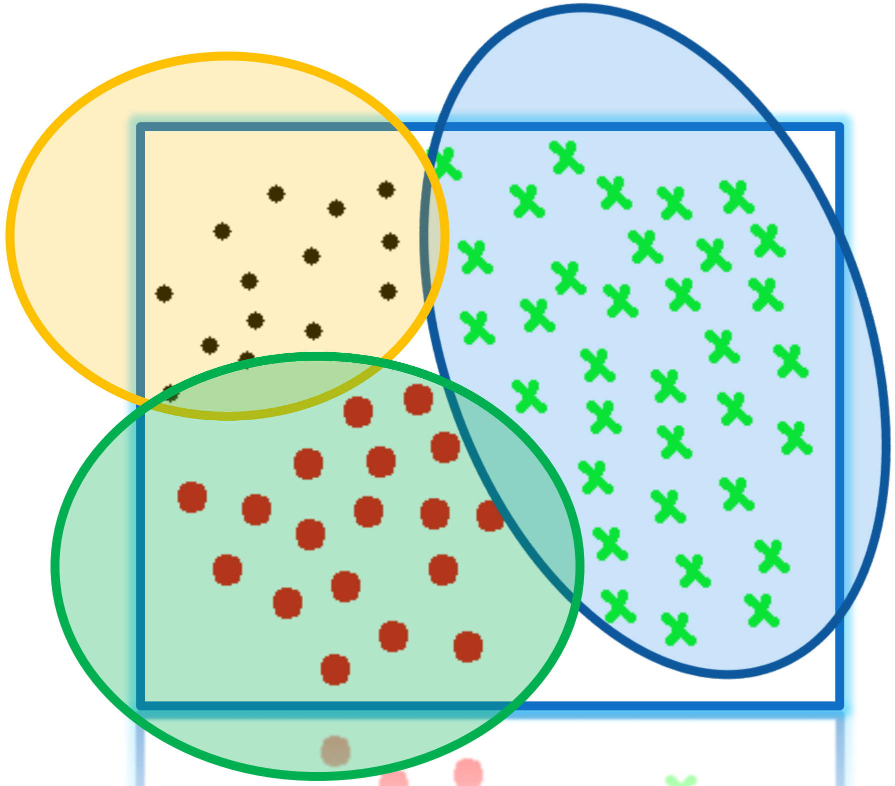


ADDRESSING MULTI-CLASS PROBLEMS BY BINARIZATION. NOVEL APPROACHES

Alberto Fernández
Mikel Galar
Francisco Herrera

State-of-the-art on One-vs-One, One-vs-All. Novel approaches.

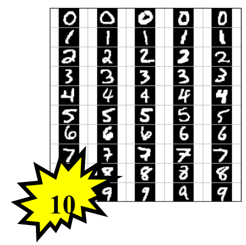
Outline

Outline

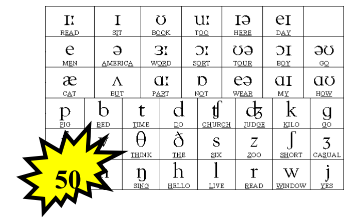
Introduction

- 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…

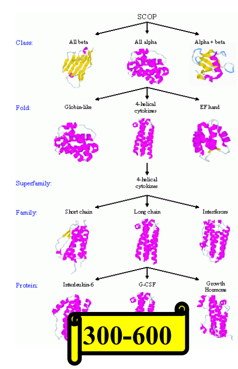
Digit recognition



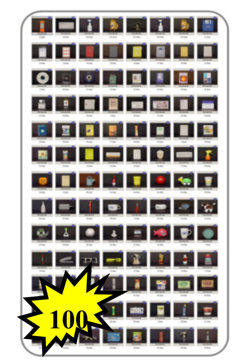
Phoneme recognition



Automated protein classification



Object recognition



Introduction. Ejemplo

Una aplicación real en KAGGLE de Problema Multiclase

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
1,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0
2,0.0,0.2,0.3,0.3,0.0,0.0,0.1,0.1,0.0
...
etc.
```

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,

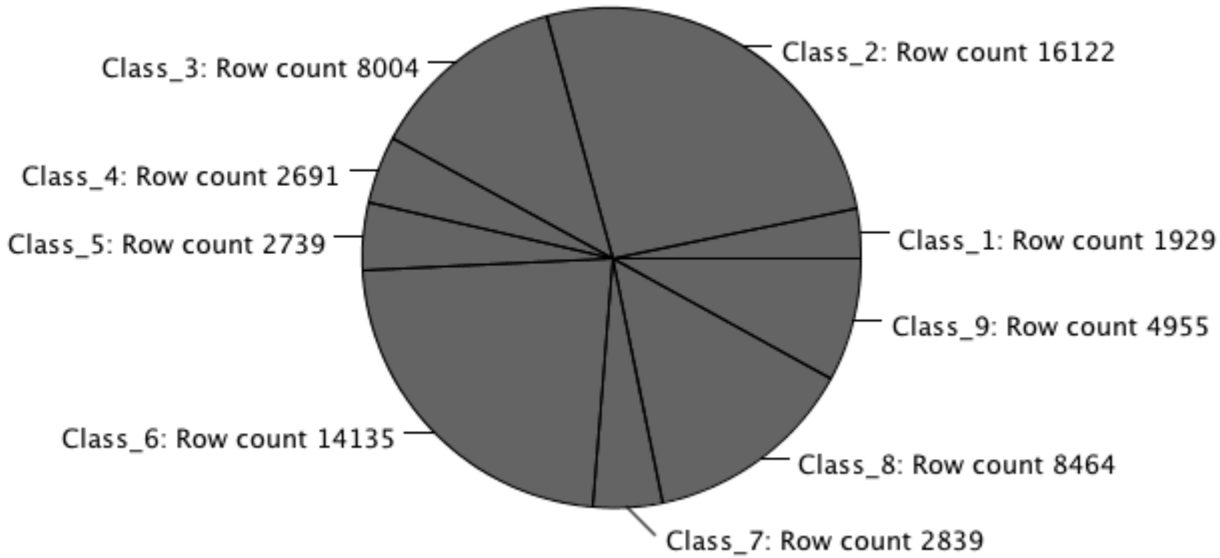
$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

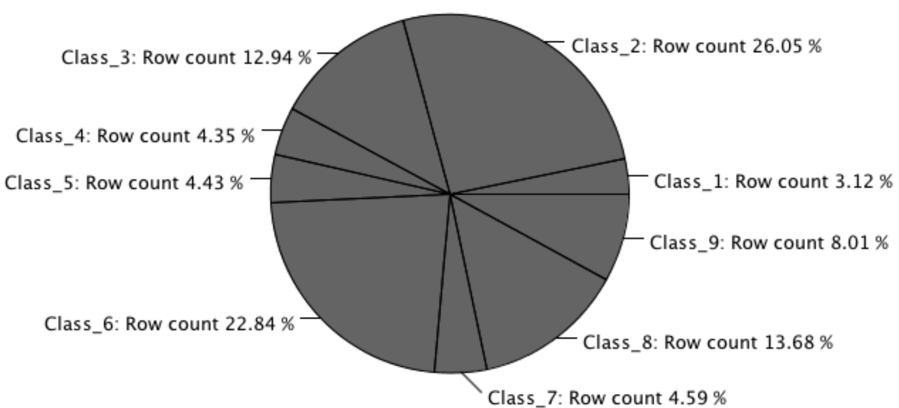
where N is the number of products in the test set, M is the number of class labels, log is the natural logarithm, y_{ij} is 1 if observation i is in class j and 0 otherwise, and p_{ij} is the predicted probability that observation i belongs to class j.

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 $max(min(p, 1 - 10^{-15}), 10^{-15})$.

Introduction. Ejemplo

Una aplicación real en KAGGLE de Problema Multiclase





Introduction. Ejemplo

Una aplicación real en KAGGLE de Problema Multiclase

otto group

$10,000 • 2,296 teams

Otto Group Product Classification Challenge

Enter/Merge by

Tue 17 Mar 2015 — Mon 18 May 2015 (32 days to go)

Dashboard

Public Leaderboard - Otto Group Product Classification Challenge

This leaderboard is calculated on approximately 70% of the test data.
The final results will be based on the other 30%, so the final standings may be different.

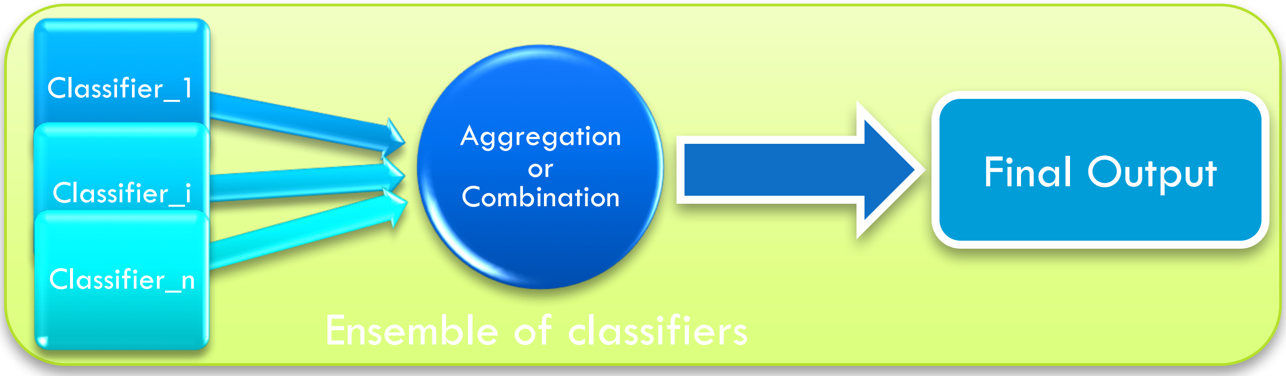
See someone using multiple accou
Let us kn

#	Δ1w	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Last Submissio
1	—	i dont know *	0.39067	67	Thu, 16 Apr 2015 21:37:26
2	—	team *	0.40017	20	Thu, 16 Apr 2015 14:45:41
3	new	tks *	0.40110	1	Thu, 16 Apr 2015 16:54:01
4	↓1	IzuiT	0.40311	43	Thu, 16 Apr 2015 05:29:26
5	↓1	Hoang Duong	0.40382	32	Thu, 16 Apr 2015 05:44:21 (-8.8d)
6	↓1	Nicholas Guttenberg	0.40857	55	Thu, 16 Apr 2015 15:46:43 (-2.6d)

Outline

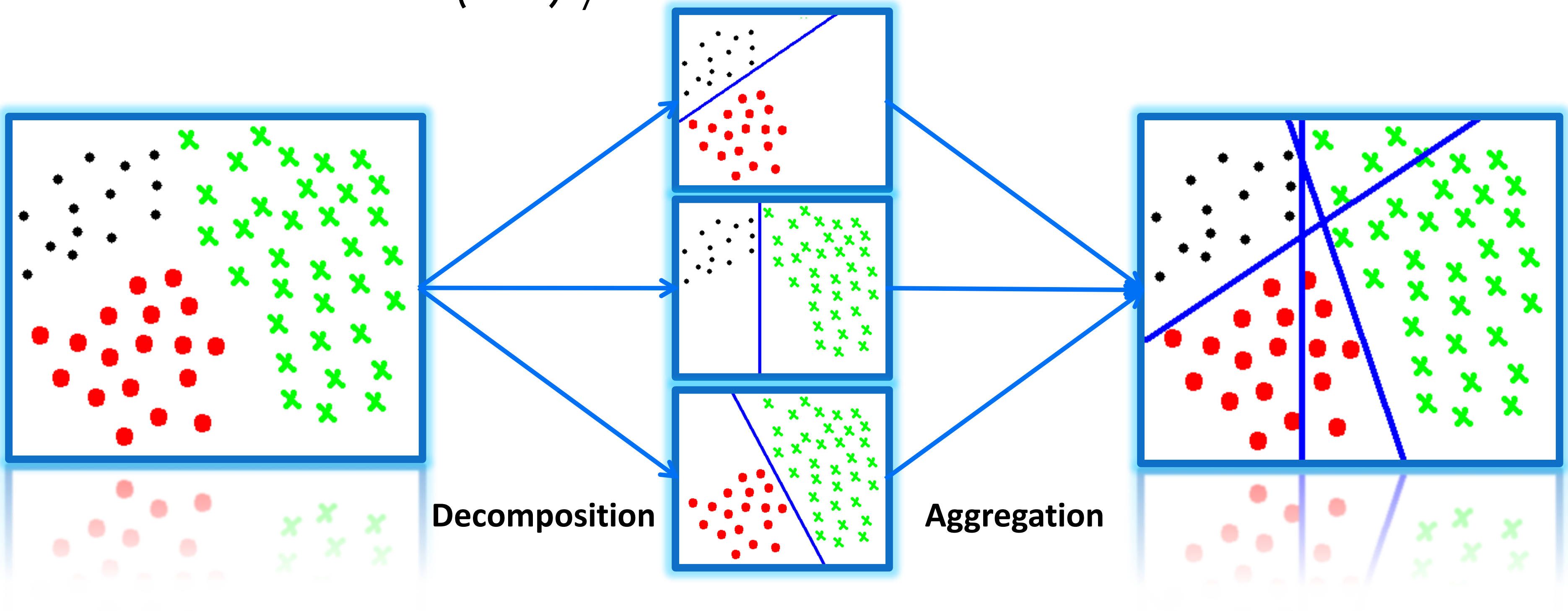
Binarization

- Decomposition of the multi-class problem
  - Divide and conquer strategy
  - Multi-class → 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?



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
  - …

[Knerr90] S. Knerr, L. Personnaz, G. Dreyfus, Single-layer learning revisited: A stepwise procedure for building and training a neural network, in: F. Fogelman Soulie, J. Herault (eds.), Neurocomputing: Algorithms, Architectures and Applications, vol. F68 of NATO ASI Series, Springer-Verlag, 1990, pp. 41–50.

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 n° of votes

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

[Fürnkranz06] J. Fürnkranz, E. Hüllermeier, S. Vanderlooy, Binary decomposition methods for multipartite ranking, in: W. L. Buntine, M. Grobelnik, D. Mladenic, J. Shawe-Taylor (eds.), Machine Learning and Knowledge Discovery in Databases, vol. 5781(1) of LNCS, Springer, 2006, pp. 359–374.

One-vs-One

- Related works
  - Round Robin Ripper (R3) [Fürnkranz02]
  - Fuzzy R3 (FR3) [Huhn09]
  - Probability estimates by Pairwise Coupling [Wu04]
  - Comparison between OVO, Boosting and Bagging [Fürnkranz03]
  - Many aggregation proposals
    - There is not a proper comparison between them

[Fürnkranz02] J. Fürnkranz, Round robin classification, Journal of Machine Learning Research 2 (2002) 721–747.

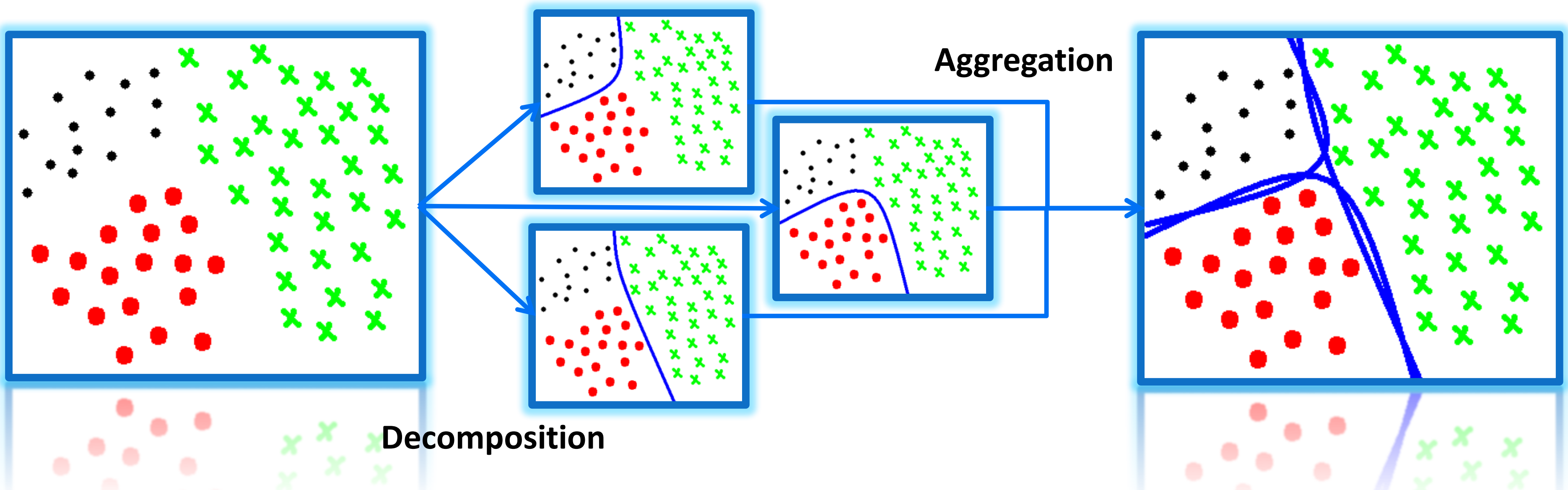
[Huhn09] J. C. Huhn, E. Hüllermeier, FR3: A fuzzy rule learner for inducing reliable classifiers, IEEE Transactions on Fuzzy Systems 17 (1) (2009) 138–149.

[Wu04] T. F. Wu, C. J. Lin, R. C. Weng, Probability estimates for multi-class classification by pairwise coupling, Journal of Machine Learning Research 5 (2004) 975–1005.

[Fürnkranz03] J. Fürnkranz, Round robin ensembles, Intelligent Data Analysis 7 (5) (2003) 385–403.

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*



One-vs-All

- Advantages
  - Less n° of classifiers
  - All examples are "competent"
- Disadvantages
  - Less studied in the literature
    - low n° of aggregations
      - Simplest: Maximum confidence rule (max(r_{ij}))

$$R = (r_1, r_2, \ldots, r_i, \ldots, r_m)$$

  - More complex problems
  - Imbalance training sets

One-vs-All

- Related Works
  - Rifkin and Klatau [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

[Rifkin04] R. Rifkin, A. Klautau, In defense of one-vs-all classification, Journal of Machine Learning Research 5 (2004) 101–141.

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

Class	Classifier								
	C1	C2	C3	C4	C5	C5	C7	C8	C9
Class1	1	1	1	0	0	0	1	1	1
Class2	0	0	-1	1	1	0	1	-1	-1
Class3	-1	0	0	-1	0	1	-1	1	-1
Class4	0	-1	0	0	-1	-1	-1	-1	1

[Allwein00] E. L. Allwein, R. E. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, Journal of Machine Learning Research 1 (2000) 113–141.

Decomposition Strategies

- Other approaches
  - Hierarchical approaches
    - Distinguish groups of classes in each nodes
  - Detailed review of decomposition strategies in [Lorena09]
    - Only an enumeration of methods
    - Low importance to the aggregation step

[Lorena09] A. C. Lorena, A. C. Carvalho, J. M. Gama, A review on the combination of binary classifiers in multiclass problems, Artificial Intelligence Review 30 (1-4) (2008) 19–37.

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
      - …

Outline

State-of-the-art on aggregation for OVO

- Starting from the score-matrix

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

  - 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

$$Class = arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} s_{ij}$$

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

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

$$Class = arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} r_{ij}$$

[Friedman96] J. H. Friedman, Another approach to polychotomous classification, Tech. rep., Department of Statistics, Stanford University (1996).

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} = *Prob(Class$_i$ | Class$_i$ or Class$_j$)*
    - Find the best approximation $\widehat{\mathbf{p}} = (\widehat{p}_1, \ldots, \widehat{p}_m)$
    - Predicts: $Class = arg \max_{i=1,\ldots,m} \widehat{p}_i$
  - Algorithm: Minimization of Kullack-Leibler (KL) distance

$$l(\mathbf{p}) = \sum_{1 \leq j \neq i \leq m} n_{ij} r_{ij} log \frac{r_{ij}}{\mu_{ij}} = \sum_{i<j} n_{ij} \left(r_{ij} log \frac{r_{ij}}{\mu_{ij}} + (1 - r_{ij}) log \frac{1 - r_{ij}}{1 - \mu_{ij}} \right)$$

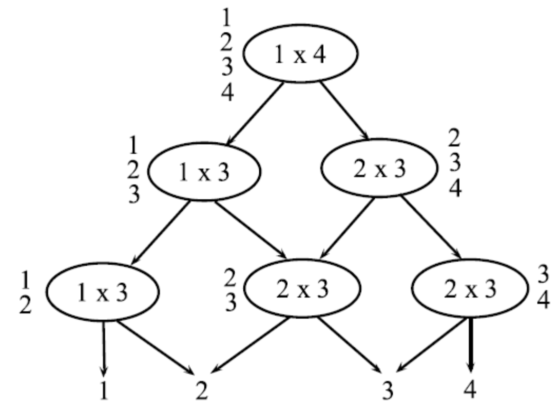
- where $\mu_{ij} = p_i/(p_i + p_j)$, $r_{ji} = 1 - r_{ij}$ and n_{ij} is the number of examples of classes i and j

[Hastie98] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Annals of Statistics 26 (2) (1998) 451–471.

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.

[Platt00] J. C. Platt, N. Cristianini and J. Shawe-Taylor, Large Margin DAGs for Multiclass Classification, Proc. Neural Information Processing Systems (NIPS'99), S.A. Solla, T.K. Leen and K.-R. Müller (eds.), (2000) 547-553.



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
    $$Class = arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} P_{ij} + \frac{1}{2} C_{ij} + \frac{N_i}{N_i + N_j} I_{ij}$$
    - where Ni is the number of examples of class i in training

[Hüllermeier08,Huhn09]

[Hüllermeier08] E. Hüllermeier and K. Brinker. Learning valued preference structures for solving classification problems. Fuzzy Sets and Systems, 159(18):2337–2352, 2008.

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 → normalize $\bar{r}_{ij} = \frac{r_{ij}}{r_{ij} + r_{ji}}$
    - Compute the maximal non-dominated elements
      - Construct the strict preference relation $r'_{ij} = \begin{cases} \bar{r}_{ij} - \bar{r}_{ji}, & \text{when } \bar{r}_{ij} > \bar{r}_{ji} \\ 0, & \text{otherwise.} \end{cases}$
      - Compute the non-dominance degree $ND_i = 1 - \sup_{j \in C}[r'_{ji}]$
        - *the degree to which the class i is dominated by no one of the remaining classes*
      - Output $Class = arg \max_{i=1,\ldots,m} \{ND_i\}$

[Fernandez10] A. Fernández, M. Calderón, E. Barrenechea, H. Bustince, F. Herrera, Solving mult-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations, Fuzzy Sets and System *161:23 (2010) 3064-3080,*

[Orlovsky78] S. A. Orlovsky, Decision-making with a fuzzy preference relation, Fuzzy Sets and Systems 1 (3) (1978) 155–167.

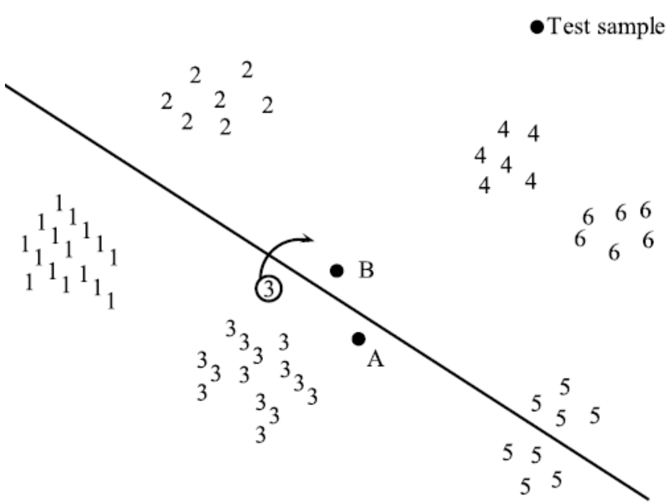
State-of-the-art on aggregation for OVO

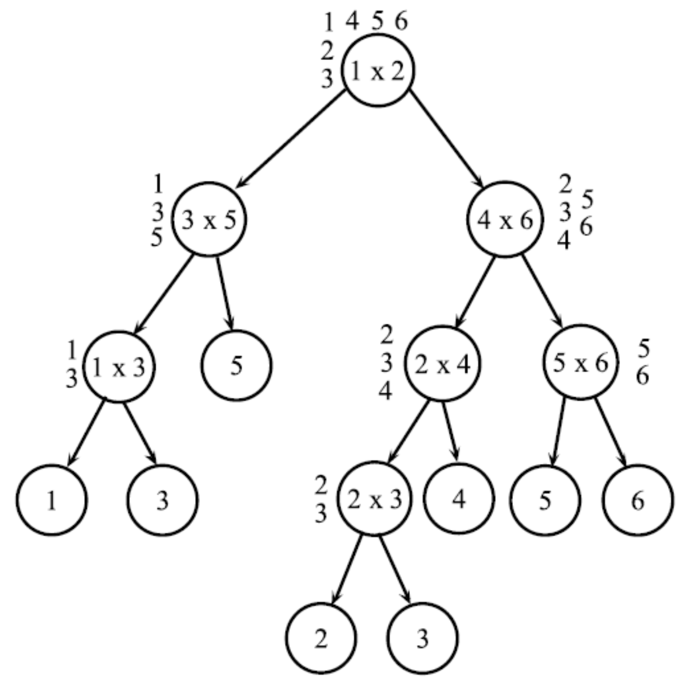
- Binary Tree of Classifiers (BTC)
  - From Binary Tree of SVM [Fei06]
  - Reduce the number of classifiers
  - Idea: Some of the binary classifiers which discriminate between two classes
    - Also can distinguish other classes at the same time
  - Tree constructed recursively
    - Similar to DDAG
      - 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

[Fei06] B. Fei and J. Liu. Binary tree of SVM: a new fast multiclass training and classification algorithm. IEEE Transactions on Neural Networks, 17(3):696–704, 2006

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 class1 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

[Liu07] Z. Liu, B. Hao and X. Yang. Nesting algorithm for multi-classification problems. Soft Computing, 11(4):383–389, 2007.
[Liu08] Z. Liu, B. Hao and E.C.C. Tsang. Nesting one-against-one algorithm based on SVMs for pattern classification. IEEE Transactions on Neural Networks, 19(12):2044–2052, 2008.

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 $Class = arg \max_{i=1,\ldots,m} \widehat{p}_i$
  - Similar to PC
    - But solving a different optimization

$$\min_{\mathbf{p}} \sum_{i=1}^{m} \sum_{1 \leq j \neq i \leq m} (r_{ji}p_i - r_{ij}p_j)^2 \quad \text{subject to } \sum_{i=1}^{k} p_i = 1, p_i \geq 0, \forall i.$$

Outline

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

$$Class = arg \max_{i=1,\ldots,m} r_i$$

- 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

[Hong08] J.-H. Hong, J.-K. Min, U.-K. Cho, and S.-B. Cho. Fingerprint classification using one-vs-all support vector machines dynamically ordered with naïve bayes classifiers. Pattern Recognition, 41(2):662–671, 2008.

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

[Hsu02] C. W. Hsu, C. J. Lin, A comparison of methods for multiclass support vector machines, IEEE Transactions Neural Networks 13 (2) (2002) 415–425.

Outline

M. Galar, A.Fernández, E. Barrenechea, H. Bustince, F. Herrera, **An Overview of Ensemble Methods for Binary Classifiers in Multi-class Problems: Experimental Study on One-vs-One and One-vs-All Schemes.** *Pattern Recognition 44:8 (2011) 1761-1776, doi: 10.1016/j.patcog.2011.01.017*

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

Correct Class	Predicted Class				
	C_1	C_2	$\ldots$	C_m	Total
C_1	h_{11}	h_{12}	$\ldots$	h_{1m}	T_{r1}
C_2	h_{21}	h_{22}	$\ldots$	h_{2m}	T_{r1}
$\vdots$			$\ddots$		$\vdots$
C_m	h_{m1}	h_{m2}	$\ldots$	h_{mm}	T_{rm}
$Total$	T_{c1}	T_{c2}	$\ldots$	T_{cm}	T_{r1}

$$kappa = \frac{n \sum_{i=1}^{m} h_{ii} - \sum_{i=1}^{m} T_{ri} T_{ci}}{n^2 - \sum_{i=1}^{m} T_{ri} T_{ci}}$$

Experimental Framework

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

Data-set	#Ex.	#Atts.	#Num.	#Nom.	#Cl.
Car	1728	6	6	0	4
Lymphography	148	18	3	15	4
Vehicle	846	18	18	0	4
Cleveland	297	13	13	0	5
Nursery	1296	8	0	8	5
Page-blocks	548	10	10	0	5
Autos	159	25	15	10	6
Dermatology	366	33	1	32	6
Flare	1389	10	0	10	6
Glass	214	9	9	0	6
Satimage	643	36	36	0	7
Segment	2310	19	19	0	7
Shuttle	2175	9	9	0	7
Zoo	101	16	0	16	7
Ecoli	336	7	7	0	8
Led7digit	500	7	0	7	10
Penbased	1099	16	16	0	10
Yeast	1484	8	8	0	10
Vowel	990	13	13	0	11

Experimental Framework

- Algorithms parameters
  - Default configuration

Algorithm	*Parameters*
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
PDFC	C = 100.0 Tolerance Parameter = 0.001 Epsilon = 1.0E-12 Kernel Type = Polynomial Polynomial Degree = 1 PDRF Type = Gaussian

Experimental Framework

- Confidence estimations
  - SVM: Logistic model
    - SVM for probability estimates
  - C4.5: Purity of the predictor leaf
    - N° of instances correctly classified by the leaf / Total n° of instances in the leaf
  - kNN: $Confidence = \frac{\sum_{l=1}^{k} \frac{e_l}{d_l}}{\sum_{l=1}^{k} \frac{1}{d_l}}$
    - where d_l = distance between the input pattern and the l^{th} neighbor
    - $e_l = 1$ if the neighbor l is from the class and 0 otherwise
  - Ripper: Purity of the rule
    - N° of instances correctly classified by the rule / Total n° of instances in the rule
  - PDFC: confidence = 1 is given for the predicted class

Outline

M. Galar, A.Fernández, E. Barrenechea, H. Bustince, F. Herrera, **An Overview of Ensemble Methods for Binary Classifiers in Multi-class Problems: Experimental Study on One-vs-One and One-vs-All Schemes.** *Pattern Recognition 44:8 (2011) 1761-1776, doi: 10.1016/j.patcog.2011.01.017*

Experimental Study

- Average accuracy and kappa results

Method	Aggregation	SVM		C4.5		1NN	
		Acc_{tst}	Avg. Rank	Acc_{tst}	Avg. Rank	Acc_{tst}	Avg. Rank
Base	-	-	-	80.51 ± 3.85	-	81.24 ± 2.98	-
OVO	VOTE	81.14 ± 3.22	**4.37 (1)**	81.57 ± 3.29	4.63 (4)	82.06 ± 3.38	3.82 (3)
	WV	81.05 ± 2.92	5.08 (6)	**81.59 ± 3.28**	3.97 (2)	-	-
	DDAG	81.01 ± 3.28	5.39 (8)	81.02 ± 3.56	6.21 (9)	81.86 ± 3.31	4.32 (5)
	PC	81.08 ± 2.89	5.29 (7)	81.49 ± 3.32	4.34 (3)	82.26 ± 3.33	3.21 (2)
	LVPC	81.14 ± 3.11	4.50 (3)	81.57 ± 3.28	**3.87 (1)**	-	-
	ND	81.01 ± 3.15	4.92 (5)	81.12 ± 3.24	5.58 (6)	81.48 ± 3.51	4.97 (7)
	BTC	80.82 ± 3.24	6.18 (9)	81.22 ± 2.87	5.61 (7)	82.21 ± 3.12	3.89 (4)
	NEST	**81.14 ± 3.32**	4.47 (2)	81.20 ± 3.47	5.74 (8)	81.68 ± 3.47	4.68 (6)
	PE	81.03 ± 3.35	4.79 (4)	81.42 ± 3.22	5.05 (5)	**82.30 ± 3.11**	**3.11 (1)**
OVA	MAX	78.66 ± 3.00	1.53 (2)	78.01 ± 4.19	1.84 (2)	81.18 ± 4.51	1.63 (2)
	DOO	**78.75 ± 3.15**	**1.47 (1)**	**78.78 ± 4.36**	**1.16 (1)**	**81.77 ± 4.45**	**1.37 (1)**

Method	Aggregation	SVM		C4.5		1NN	
		$Kappa_{tst}$	Avg. Rank	$Kappa_{tst}$	Avg. Rank	$Kappa_{tst}$	Avg. Rank
Base	-	-	-	.7203 ± .0554	-	.7369 ± .0475	-
OVO	VOTE	.7233 ± .0548	4.82 (2)	.7331 ± .0490	5.16 (5)	.7419 ± .0535	3.84 (3)
	WV	.7229 ± .0506	5.05 (6)	**.7348 ± .0485**	**3.76 (1)**	-	-
	DDAG	.7230 ± .0555	5.11 (7)	.7304 ± .0535	5.92 (8)	.7402 ± .0522	3.89 (4)
	PC	.7234 ± .0520	5.18 (8)	.7341 ± .0493	4.13 (3)	.7449 ± .0525	3.00 (2)
	LVPC	.7211 ± .0531	5.03 (5)	.7341 ± .0488	4.03 (2)	-	-
	ND	.7225 ± .0533	4.82 (2)	.7286 ± .0489	5.53 (7)	.7340 ± .0556	5.37 (7)
	BTC	.7204 ± .0551	6.05 (9)	.7297 ± .0428	5.42 (6)	.7438 ± .0498	4.29 (5)
	NEST	**.7243 ± .0559**	**4.03 (1)**	.7291 ± .0514	6.34 (9)	.7366 ± .0547	4.79 (6)
	PE	.7228 ± .0537	4.92 (4)	.7330 ± .0480	4.71 (4)	**.7453 ± .0497**	**2.82 (1)**
OVA	MAX	.6868 ± .0553	1.55 (2)	.6826 ± .0629	1.89 (2)	.7298 ± .0705	1.63 (2)
	DOO	**.6868 ± .0565**	**1.45 (1)**	**.6938 ± .0649**	**1.11 (1)**	**.7368 ± .0701**	**1.37 (1)**

Experimental Study

- Average accuracy and kappa results

Method	Aggregation	3NN		Ripper		PDFC	
		Acc_{tst}	Avg. Rank	Acc_{tst}	Avg. Rank	Acc_{tst}	Avg. Rank
Base	-	81.54 ± 2.65	-	76.52 ± 4.00	-	-	-
OVO	VOTE	83.00 ± 2.92	5.05 (6)	80.57 ± 3.17	3.89 (3)	**84.33 ± 3.10**	3.37 (2)
	WV	**83.11 ± 2.87**	4.47 (3)	80.54 ± 3.03	3.87 (2)	-	-
	DDAG	82.73 ± 2.83	5.87 (8)	77.62 ± 3.61	7.08 (9)	84.05 ± 3.00	3.71 (3)
	PC	83.00 ± 2.96	5.11 (7)	80.33 ± 3.30	4.87 (5)	84.12 ± 3.05	**3.29 (1)**
	LVPC	83.07 ± 2.79	4.61 (4)	**80.58 ± 3.16**	**3.68 (1)**	-	-
	ND	83.07 ± 2.93	**4.29 (1)**	79.38 ± 3.27	5.29 (7)	84.05 ± 2.96	4.68 (6)
	BTC	82.99 ± 2.98	5.00 (5)	79.19 ± 3.07	6.39 (8)	84.24 ± 3.01	4.29 (5)
	NEST	82.67 ± 2.94	6.16 (9)	80.01 ± 3.50	5.08 (6)	83.88 ± 3.02	4.89 (7)
	PE	83.11 ± 2.94	4.45 (2)	80.07 ± 3.08	4.84 (4)	84.06 ± 3.04	3.76 (4)
OVA	MAX	82.75 ± 4.29	1.58 (2)	78.30 ± 4.94	1.71 (2)	**83.59 ± 3.12**	**1.39 (1)**
	DOO	**82.76 ± 4.38**	**1.42 (1)**	**79.12 ± 4.67**	**1.29 (1)**	83.01 ± 3.10	1.61 (2)

Method	Aggregation	3NN		Ripper		PDFC	
		$Kappa_{tst}$	Avg. Rank	$Kappa_{tst}$	Avg. Rank	$Kappa_{tst}$	Avg. Rank
Base	-	.7335 ± .0452	-	.6799 ± .0554	-	-	-
OVO	VOTE	.7507 ± .0500	5.03 (6)	**.7250 ± .0475**	4.26 (3)	**.7677 ± .0538**	3.63 (2)
	WV	.7519 ± .0487	4.71 (3)	.7249 ± .0455	**3.68 (1)**	-	-
	DDAG	.7479 ± .0487	5.87 (8)	.6957 ± .0489	6.42 (8)	.7659 ± .0518	3.97 (5)
	PC	.7505 ± .0505	4.89 (5)	.7227 ± .0483	4.61 (5)	.7670 ± .0529	**3.11 (1)**
	LVPC	.7496 ± .0475	5.18 (7)	.7246 ± .0469	4.00 (2)	-	-
	ND	.7524 ± .0500	**4.03 (1)**	.7098 ± .0479	5.92 (7)	.7625 ± .0524	5.32 (7)
	BTC	.7519 ± .0514	4.87 (4)	.7087 ± .0476	6.58 (9)	.7668 ± .0527	3.79 (3)
	NEST	.7461 ± .0505	6.24 (9)	.7195 ± .0496	4.97 (6)	.7641 ± .0514	4.37 (6)
	PE	**.7526 ± .0499**	4.18 (2)	.7193 ± .0457	4.55 (4)	.7653 ± .0524	3.82 (4)
OVA	MAX	**.7481 ± .0695**	1.58 (2)	.6896 ± .0743	1.79 (2)	**.7556 ± .0589**	**1.37 (1)**
	DOO	.7473 ± .0710	**1.42 (1)**	**.7004 ± .0716**	**1.21 (1)**	.7478 ± .0587	1.63 (2)

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 → confidences in {0,1}
      - In PDFC they also excel
    - ND the worst → 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

Base Classifier	Measure	R^+	R^-	Hypothesis ($\alpha = 0.05$)	p-value
SVM	Accuracy	82	108	Not Rejected	0.53213
	Kappa	86	104	Not Rejected	0.75637
C4.5	Accuracy	14	176	Rejected for DOO	0.00179
	Kappa	11.5	178.5	Rejected for DOO	0.00118
1NN	Accuracy	55	135	Not Rejected	0.09097
	Kappa	55	135	Not Rejected	0.09097
3NN	Accuracy	75	115	Not Rejected	0.73532
	Kappa	76	114	Not Rejected	0.86577
Ripper	Accuracy	44.5	145.5	Rejected for DOO	0.04286
	Kappa	42	148	Rejected for DOO	0.03294
PDFC	Accuracy	130.5	59.5	Rejected for MAX	0.0464
	Kappa	138	52	Rejected for MAX	0.02799

Should we do binarization? How should we do it?

- Representatives of OVO and OVA
  - By the previous analysis

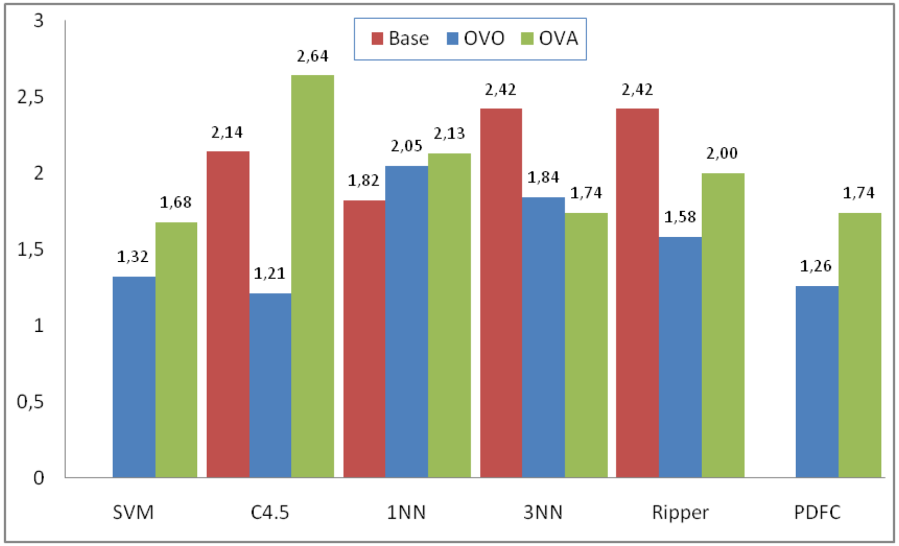
	SVM	C4.5	1NN	3NN	Ripper	PDFC
OVO	$NEST_{ovo}$	WV_{ovo}	PE_{ovo}	ND_{ovo}	WV_{ovo}	PC_{ovo}
OVA	DOO_{ova}	DOO_{ova}	DOO_{ova}	DOO_{ova}	DOO_{ova}	MAX_{ova}

  - Average results

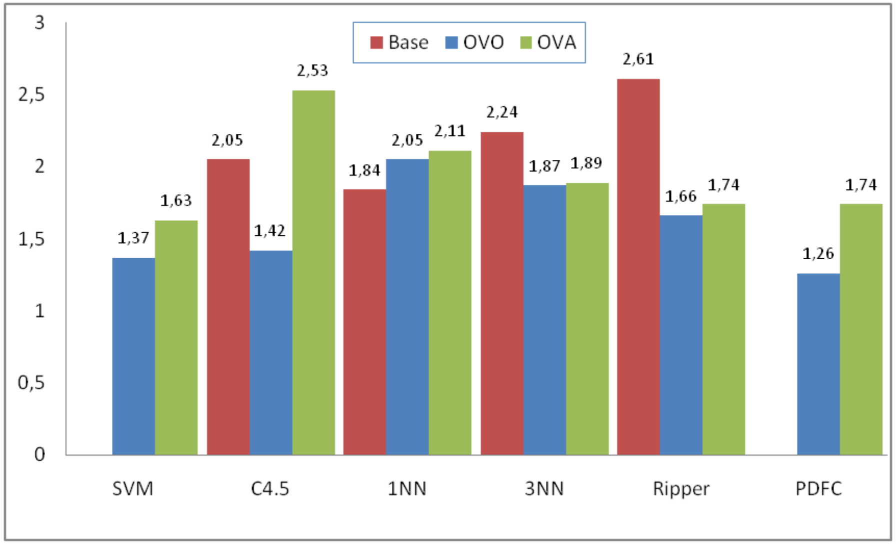
Base Classifier	Aggregation	Accuracy Test	Accuracy Avg. Rank	Kappa Test	Kappa Avg.Rank
SVM	$NEST_{ovo}$	**81.14 ± 3.32**	**1.37 (1)**	**.7243 ± .0559**	**1.32 (1)**
	DOO_{ova}	78.75 ± 3.15	1.63 (2)	.6868 ± .0565	1.68 (2)
C4.5	C45	80.51 ± 3.85	2.05 (2)	.7203 ± .0554	2.14 (2)
	WV_{ovo}	**81.59 ± 3.28**	**1.42 (1)**	**.7348 ± .0485**	**1.21 (1)**
	DOO_{ova}	78.78 ± 4.36	2.53 (3)	.6938 ± .0649	2.64 (3)
1NN	1NN	81.24 ± 2.98	**1.84 (1)**	.7369 ± .0475	**1.82 (1)**
	PE_{ovo}	**82.30 ± 3.11**	2.05 (2)	**.7453 ± .0497**	2.05 (2)
	DOO_{ova}	81.77 ± 4.45	2.11 (3)	.7368 ± .0701	2.13 (3)
3NN	3NN	81.54 ± 2.65	2.24 (3)	.7335 ± .0452	2.42 (3)
	ND_{ovo}	**83.07 ± 2.93**	**1.87 (1)**	**.7524 ± .0500**	1.84 (2)
	DOO_{ova}	82.76 ± 4.38	1.89 (2)	.7473 ± .0710	**1.74 (1)**
Ripper	Ripper	76.52 ± 4.00	2.61 (3)	.6799 ± .0554	2.42 (3)
	WV_{ovo}	**80.54 ± 3.03**	**1.66 (1)**	**.7249 ± .0455**	**1.58 (1)**
	DOO_{ova}	79.12 ± 4.67	1.74 (2)	.7004 ± .0716	2.00 (2)
PDFC	PC_{ovo}	**84.12 ± 3.05**	**1.26 (1)**	**.7670 ± .0529**	**1.26 (1)**
	MAX_{ova}	83.59 ± 3.12	1.74 (2)	.7556 ± .0589	1.74 (2)

Should we do binarization? How should we do it?

- Rankings within each classifier
  - In general, OVO is the most competitive



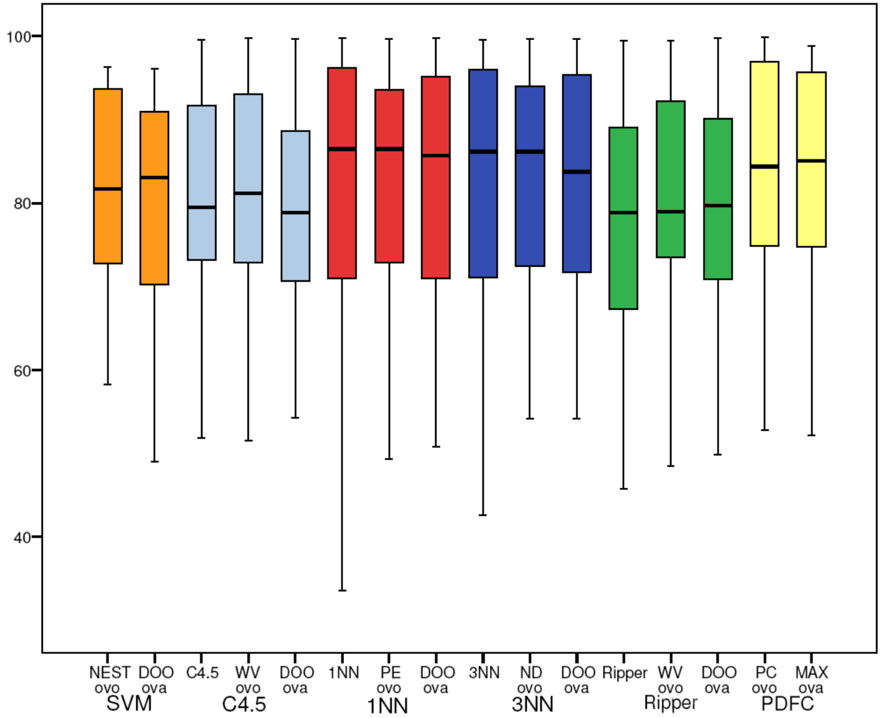
Accuracy

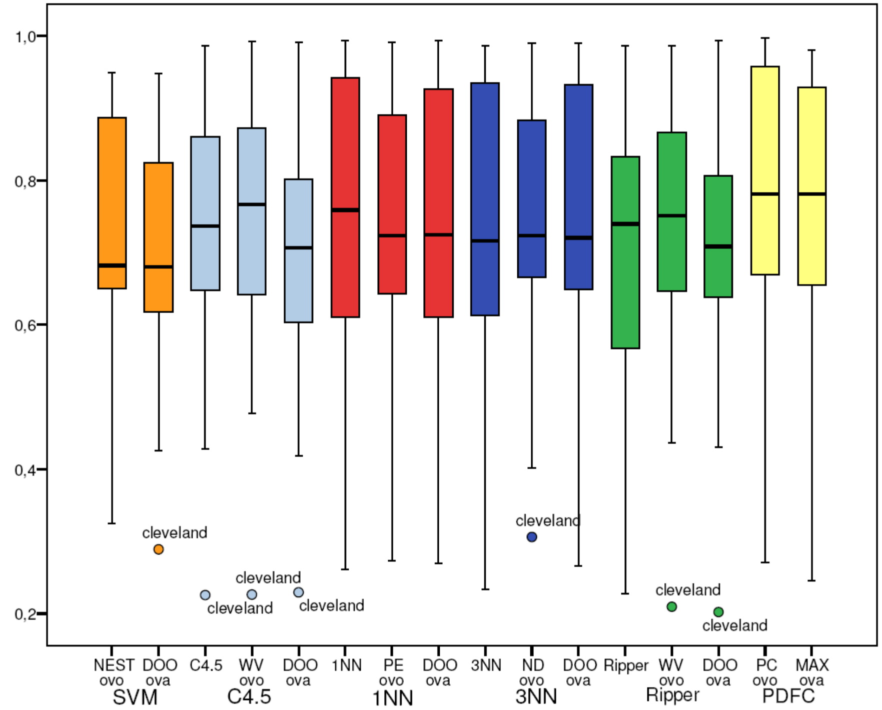


Kappa

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

Base Classifier	Comparison	Measure	R^+	R^-	Hypothesis ($\alpha = 0.05$)	p-value
SVM	$NEST_{ovo}$ vs. DOO_{ova}	Accuracy	153	37	Rejected for $NEST_{ovo}$	0.01959
		Kappa	156	34	Rejected for $NEST_{ovo}$	0.0141
PDFC	PC_{ovo} vs. MAX_{ova}	Accuracy	146	44	Rejected for PC_{ovo}	0.04014
		Kappa	147	43	Rejected for PC_{ovo}	0.03639

  - C4.5, 1NN, 3NN and Ripper
    - P-values returned by Iman-Davenport tests (* if rejected)

	C4.5	1NN	3NN	Ripper
Accuracy	0.00134 *	0.70296	0.45982	0.00296 *
Kappa	0.00026 *	0.61089	0.07585	0.02982 *

    - Post-hoc test for C4.5 and Ripper
    - 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

(a) Accuracy

i	Hypothesis	p-value
1	WV_{ovo} vs. DOO_{ova}	+(0.00197)
2	C4.5 vs. WV_{ovo}	=(0.05158)
3	C4.5 vs. DOO_{ova}	=(0.14429)

(b) Kappa

i	Hypothesis	p-value
1	WV_{ovo} vs. DOO_{ova}	+(0.00057)
2	C4.5 vs. WV_{ovo}	+(0.03496)
3	C4.5 vs. DOO_{ova}	=(0.10476)

  - Ripper
    - WV for OVO is the best
      - No statistical differences with OVA
        - But OVO differs statistically from Ripper while OVA do not

(a) Accuracy

i	Hypothesis	p-value
1	Ripper vs. WV_{ovo}	+(0.01050)
2	Ripper vs. DOO_{ova}	+(0.01050)
3	WV_{ovo} vs. DOO_{ova}	=(0.80775)

(b) Kappa

i	Hypothesis	p-value
1	Ripper vs. WV_{ovo}	+(0.02833)
2	Ripper vs. DOO_{ova}	=(0.19437)
3	WV_{ovo} vs. DOO_{ova}	=(0.19437)

Outline

M. Galar, A.Fernández, E. Barrenechea, H. Bustince, F. Herrera, **An Overview of Ensemble Methods for Binary Classifiers in Multi-class Problems: Experimental Study on One-vs-One and One-vs-All Schemes.** *Pattern Recognition 44:8 (2011) 1761-1776, doi: 10.1016/j.patcog.2011.01.017*

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

M. Galar, A.Fernández, E. Barrenechea, H. Bustince, F. Herrera, **An Overview of Ensemble Methods for Binary Classifiers in Multi-class Problems: Experimental Study on One-vs-One and One-vs-All Schemes.** *Pattern Recognition 44:8 (2011) 1761-1776, doi: 10.1016/j.patcog.2011.01.017*

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

M. Galar, A.Fernández, E. Barrenechea, H. Bustince, F. Herrera, **An Overview of Ensemble Methods for Binary Classifiers in Multi-class Problems: Experimental Study on One-vs-One and One-vs-All Schemes.** *Pattern Recognition 44:8 (2011) 1761-1776,* doi: 10.1016/j.patcog.2011.01.017

Outline

M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, **Dynamic Classifier Selection for One-vs-One Strategy: Avoiding Non-Competent Classifiers**. *Pattern Recognition 46:12 (2013) 3412–3424, doi: j.patcog.2013.04.018*

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
    - $$R(x) = \begin{pmatrix} & c1 & c2 & c3 & c4 & c5 \\ c1 & - & 0{,}55 & 0{,}6 & 0{,}75 & 0{,}7 \\ c2 & 0{,}45 & - & 0{,}4 & 1 & 0{,}8 \\ c3 & 0{,}4 & 0{,}6 & - & 0{,}5 & 0{,}4 \\ c4 & 0{,}25 & 0{,}0 & 0{,}5 & - & 0{,}1 \\ c5 & 0{,}30 & 0{,}2 & 0{,}6 & 0{,}9 & - \end{pmatrix}$$

Dynamic OVO: Avoiding Non-competence

- *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
  - $$R(x) = \begin{pmatrix} & c1 & c2 & c3 & c4 & c5 & WV \\ c1 & - & 0{,}55 & 0{,}6 & 0{,}75 & 0{,}7 & 2{,}6 \\ c2 & 0{,}45 & - & 0{,}4 & 1 & 0{,}8 & \mathbf{2{,}65} \\ c3 & 0{,}4 & 0{,}6 & - & 0{,}5 & 0{,}4 & 1{,}9 \\ c4 & 0{,}25 & 0{,}0 & 0{,}5 & - & 0{,}1 & 0{,}85 \\ c5 & 0{,}30 & 0{,}2 & 0{,}6 & 0{,}9 & - & 2{,}1 \end{pmatrix}$$

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
    - Asumming that several misses can be done (they are corrected)

Dynamic OVO: Avoiding Non-competence

- **Avoding 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** → reduced score matrix

[WOODS97] K. Woods, W. Philip Kegelmeyer, K. Bowyer. Combination of multiple classifiers using local accuracy estimates, IEEE Transactions on Pattern Analysis and Machine Intelligence 19(4):405-410, 1997.

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

Algorithm 1 Dynamic Classifier Selection for OVO scheme

procedure DYNAMIC OVO(e, R)
  $k = 3 \cdot m$ ▷ m is the number of classes
  repeat
    $Neighbours \leftarrow kNN(e)$
    $C \leftarrow Classes(Neighbours)$ ▷ We select the class labels in the neighbourhood
    $k{+}{+}$
  until $\#C > 1$ **or** $k == 6 \cdot m$
  if $C > 1$ **then**
    $R' \leftarrow [R - rows(i), cols(i)];\ i \notin C$
    return R' ▷ A subset of the score matrix
  else
    return R ▷ Standard OVO approach
  end if
end procedure

Dynamic OVO: Avoiding Non-competence

- Classify x, whose real class is c_1
- $R(x) = \begin{array}{c} \\ c1 \\ c2 \\ c3 \\ c4 \\ c5 \end{array} \begin{array}{c} \begin{array}{ccccc} c1 & c2 & c3 & c4 & c5 \end{array} \\ \begin{pmatrix} - & 0{,}55 & 0{,}6 & 0{,}75 & 0{,}7 \\ 0{,}45 & - & 0{,}4 & 1 & 0{,}8 \\ 0{,}4 & 0{,}6 & - & 0{,}5 & 0{,}4 \\ 0{,}25 & 0{,}0 & 0{,}5 & - & 0{,}1 \\ 0{,}30 & 0{,}2 & 0{,}6 & 0{,}9 & - \end{pmatrix} \end{array}$

Dynamic OVO: Avoiding Non-competence

- Consider WV aggregation, $c_2$ispredicted
- **None**of theclassifiers**considering $c_1$failed**
- **Non-competent**classifiers**stronglyvotedfor**c_2
- $$R(x) = \begin{array}{c} \\ c1 \\ c2 \\ c3 \\ c4 \\ c5 \end{array} \left(\begin{array}{ccccc|c} c1 & c2 & c3 & c4 & c5 & WV \\ - & 0{,}55 & 0{,}6 & 0{,}75 & 0{,}7 & 2{,}6 \\ 0{,}45 & - & 0{,}4 & 1 & 0{,}8 & \mathbf{2,65} \\ 0{,}4 & 0{,}6 & - & 0{,}5 & 0{,}4 & 1{,}9 \\ 0{,}25 & 0{,}0 & 0{,}5 & - & 0{,}1 & 0{,}85 \\ 0{,}30 & 0{,}2 & 0{,}6 & 0{,}9 & - & 2{,}1 \end{array} \right)$$

Dynamic OVO: Avoiding Non-competence

- Applying Dynamic*k*NN
  - Compute the*k*NN of x (k = 3 · 5 = 15)
  - Subset of classes = $\{c_1, c_4, c_5\}$
  - Remove $\{c_2, c_3\}$ fromthe score-matrix
  - Apply WV to thereduced score-matrix
- $$R_{dyn}(x) = \begin{array}{cc} & \begin{array}{ccccc|c} c1 & \cancel{c2} & \cancel{c3} & c4 & c5 & WV \end{array} \\ \begin{array}{c} c1 \\ \cancel{c2} \\ \cancel{c3} \\ c4 \\ c5 \end{array} & \left(\begin{array}{ccccc|c} - & \cancel{0{,}55} & \cancel{0{,}6} & 0{,}75 & 0{,}7 & \mathbf{1{,}45} \\ \cancel{0{,}45} & - & \cancel{0{,}4} & \cancel{1} & \cancel{0{,}8} & - \\ \cancel{0{,}4} & \cancel{0{,}6} & - & \cancel{0{,}5} & \cancel{0{,}4} & - \\ 0{,}25 & \cancel{0{,}0} & \cancel{0{,}5} & - & 0{,}1 & 0{,}35 \\ 0{,}30 & \cancel{0{,}2} & \cancel{0{,}6} & 0{,}9 & - & 1{,}2 \end{array}\right) \end{array}$$

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

M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, **Dynamic Classifier Selection for One-vs-One Strategy: Avoiding Non-Competent Classifiers.** *Pattern Recognition 46:12 (2013) 3412–3424, doi: j.patcog.2013.04.018*

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)

M. Galar, A. Fernández, E. Barrenechea, F. Herrera, **DRCW-OVO: Distance-based Relative Competence Weighting Combination for One-vs-One Strategy in Multi-class Problems**. *Pattern Recognition 48 (2015), 28-42, doi: 10.1016/j.patcog.2014.07.023*.

Distance-based Relative Competence Weighting Approach

- *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

$$R(\mathbf{x}) = \begin{array}{c} \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \mathbf{c}_4 \\ \mathbf{c}_5 \end{array} \begin{pmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 & \mathbf{c}_5 \\ - & 0.55 & 0.45 & 0.80 & 0.90 \\ 0.45 & - & 0.55 & 1.00 & 0.80 \\ 0.55 & 0.45 & - & 0.45 & 0.40 \\ 0.20 & 0.00 & 0.55 & - & 0.10 \\ 0.10 & 0.20 & 0.60 & 0.90 & - \end{pmatrix}$$

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

$$R(\mathbf{x}) = \left(\begin{array}{cccccc|c|c} & \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 & \mathbf{c}_5 & \mathrm{V} & \mathrm{WV} \\ \mathbf{c}_1 & - & 0.55 & 0.45 & 0.80 & 0.90 & 3 & 2.70 \\ \mathbf{c}_2 & 0.45 & - & 0.55 & 1.00 & 0.80 & 3 & \mathbf{2.80} \\ \mathbf{c}_3 & 0.55 & 0.45 & - & 0.45 & 0.40 & 1 & 1.85 \\ \mathbf{c}_4 & 0.20 & 0.00 & 0.55 & - & 0.10 & 1 & 0.85 \\ \mathbf{c}_5 & 0.10 & 0.20 & 0.60 & 0.90 & - & 2 & 1.80 \end{array}\right)$$

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 ***k*NN 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, …, 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

   $$w_{ij} = \frac{d_j^2}{d_i^2 + d_j^2},$$

   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.

$$Class = \arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} r_{ij} \cdot w_{ij}$$

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

$$R(\mathbf{x}) = \begin{array}{c} \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \mathbf{c}_4 \\ \mathbf{c}_5 \end{array} \begin{pmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 & \mathbf{c}_5 \\ - & 0.55 & 0.45 & 0.80 & 0.90 \\ 0.45 & - & 0.55 & 1.00 & 0.80 \\ 0.55 & 0.45 & - & 0.45 & 0.40 \\ 0.20 & 0.00 & 0.55 & - & 0.10 \\ 0.10 & 0.20 & 0.60 & 0.90 & - \end{pmatrix}$$

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 iscomputed to representall w_{ij}
- $$\mathrm{W}(x) = \begin{array}{c} \\ c1 \\ c2 \\ c3 \\ c4 \\ c5 \end{array} \begin{pmatrix} c1 & c2 & c3 & c4 & c5 \\ - & 0{,}56 & 0{,}36 & 0{,}69 & 0{,}80 \\ 0{,}44 & - & 0{,}31 & 0{,}64 & 0{,}76 \\ 0{,}64 & 0{,}69 & - & 0{,}80 & 0{,}88 \\ 0{,}31 & 0{,}36 & 0{,}5 & - & 0{,}64 \\ 0{,}20 & 0{,}24 & 0{,}6 & 0{,}36 & - \end{pmatrix}$$

Distance-based Relative Competence Weighting Approach

- Apply theweight-matrix W to the score-matrix R
- **WV** isapplied to obtainthepredictedclass in DRCW-OVO
- $$R^w(x) = \begin{array}{c|ccccc|c} & c1 & c2 & c3 & c4 & c5 & WV \\ \hline c1 & - & 0{,}31 & 0{,}16 & 0{,}55 & 0{,}72 & \mathbf{1,74} \\ c2 & 0{,}20 & - & 0{,}17 & 0{,}64 & 0{,}61 & 1{,}66 \\ c3 & 0{,}35 & 0{,}31 & - & 0{,}36 & 0{,}35 & 1{,}37 \\ c4 & 0{,}06 & 0{,}00 & 0{,}11 & - & 0{,}06 & 0{,}24 \\ c5 & 0{,}02 & 0{,}05 & 0{,}07 & 0{,}32 & - & 0{,}47 \end{array}$$

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.

Data-set	C45			SVM_{Poly}			SVM_{Puk}			3NN			PDFC			Ripper		
	WV	DCS	DRCW	PE	DCS	DRCW	PE	DCS	DRCW	ND	DCS	DRCW	PC	DCS	DRCW	WV	DCS	DRCW
Autos	76.24	74.96	**80.96**	73.75	73.81	**79.48**	69.02	70.27	**71.45**	**78.88**	76.96	75.14	78.82	79.40	**80.74**	**85.09**	84.42	84.58
Car	94.68	94.50	**96.99**	93.58	93.58	**97.16**	64.99	**84.84**	81.65	93.57	93.40	**96.93**	99.77	**99.88**	99.42	92.59	93.52	**96.35**
Cleveland	52.55	53.55	**55.23**	58.97	**59.31**	58.66	47.53	47.87	**48.88**	**58.31**	57.96	56.61	53.92	55.93	**56.61**	52.18	54.54	**56.90**
Dermatology	95.24	**98.32**	98.06	94.71	94.99	**95.55**	97.20	97.20	**97.48**	92.14	95.49	**96.90**	84.66	**93.85**	91.90	93.32	94.43	**95.27**
Ecoli	81.06	81.94	**85.58**	79.37	79.63	**82.25**	77.11	77.11	**81.64**	81.66	82.52	**84.30**	84.07	83.78	**84.68**	78.47	78.74	**82.34**
Flare	**75.34**	73.62	75.27	75.43	75.46	**75.86**	69.28	**73.39**	72.04	71.21	71.59	**72.43**	73.64	**73.92**	73.69	75.24	74.83	**75.60**
Glass	72.03	71.63	**74.81**	62.14	63.14	**71.04**	73.72	74.15	**76.19**	73.35	74.27	**74.33**	68.72	**70.12**	**70.12**	68.56	68.12	**75.40**
Led7digit	64.51	**65.35**	65.33	67.90	**68.09**	66.47	61.33	61.57	**62.54**	66.68	67.88	**68.26**	62.17	62.60	**65.42**	63.98	63.86	**64.19**
Lymphography	74.50	**76.44**	**76.44**	82.48	82.48	**83.10**	81.87	81.87	**82.50**	68.19	**79.55**	79.52	**83.19**	**83.19**	**83.19**	75.68	75.68	**77.04**
Nursery	89.66	89.81	**90.90**	92.13	92.13	**94.53**	80.33	89.05	**90.83**	93.29	93.29	**93.68**	**97.92**	**97.92**	97.84	90.66	90.81	**92.44**
Pageblocks	95.64	95.46	**95.82**	94.90	94.53	**95.27**	94.58	94.76	**95.11**	**95.63**	95.46	95.09	**95.09**	94.91	**95.09**	95.45	95.11	**96.00**
Penbased	91.10	91.11	**95.64**	95.92	96.01	**97.01**	97.55	97.64	**98.00**	**97.00**	96.64	96.91	**98.19**	98.10	98.10	91.38	91.11	**96.01**
Satimage	82.15	82.92	**85.41**	84.48	84.16	**86.34**	84.77	85.70	**87.56**	87.58	87.73	**88.34**	86.79	86.95	**87.25**	82.61	82.14	**86.01**
Segment	96.28	96.71	**97.97**	92.68	92.90	**95.58**	97.23	97.36	**97.40**	96.58	96.80	**96.84**	97.32	**97.36**	97.27	96.58	96.88	**97.84**
Shuttle	99.59	99.68	**99.72**	96.55	97.61	**99.50**	99.59	99.63	**99.63**	**99.50**	99.40	99.40	97.43	98.03	**98.76**	99.40	**99.68**	99.54
Vehicle	72.33	72.81	**73.88**	73.53	74.00	**74.48**	81.92	81.92	**82.04**	72.11	72.23	**72.23**	**84.53**	84.40	84.41	69.27	70.20	**71.29**
Vowel	83.43	83.64	**94.75**	71.41	71.82	**95.05**	**99.70**	**99.70**	99.29	**97.78**	97.37	97.27	98.28	98.08	**98.59**	80.20	79.39	**94.44**
Yeast	59.57	59.84	**60.46**	60.52	59.98	**60.92**	59.31	59.51	**62.14**	56.68	56.54	**58.30**	60.25	59.98	**60.92**	58.30	58.10	**61.81**
Zoo	92.17	92.17	**93.22**	95.72	95.72	**96.77**	78.06	84.13	**90.80**	89.90	91.86	**94.64**	96.77	96.77	**97.82**	94.05	94.05	**96.10**
Average	81.48	81.81	**84.02**	81.38	81.55	**84.48**	79.74	81.98	**83.01**	82.63	83.52	**84.06**	84.29	85.01	**85.36**	81.21	81.35	**84.17**

¿Questions?

- Thank you for your attention!