Sistemas Inteligentes para la Gestión de la Empresa

2016 - 2017

- Tema 1. Introducción a la Ciencia de Datos
- Tema 2. Depuración y Calidad de Datos. Preprocesamiento de datos
- Tema 3. Análisis Predictivo para la Empresa
- Tema 5. Análisis de Transacciones y Mercados
- Tema 4. Modelos avanzados de Analítica de Empresa
- Tema 6. Big Data
- Tema 7. Aplicaciones de la Ciencia de Datos en la Empresa

Sistemas Inteligentes para la Gestión de la Empresa

TEMA 3. Análisis Predictivo para la Empresa

(Modelos predictivos avanzados de clasificación)

1. Clasificación no balanceada
2. Multiclasificadores: Bagging y Boosting
3. Multiples clases: Descomposición binaria
4. Redes Neuronales y Maquinas de soporte Vectorial

Bibliografía:

G. Shmueli, N.R. Patel, P.C. Bruce
Data mining for business intelligence (Part IV)
Wiley 2010 (2nd. edition)
Data Mining and Analysis: Fundamental Concepts and Algorithms (Part 4)
M. Zaki and W. Meira Jr.
Cambridge University Press, 2014.
http://www.dataminingbook.info/DokuWiki/doku.php

V. Cherkassky, F.M. Mulier
Learning from Data: Concepts, Theory, and Methods (Sections 8 and 9)
2nd Edition, Wiley-IEE Prees, 2007

Objetivos

- Estudiar modelos avanzados de predicción.
- Conocer problemas presentes en clasificación: desequilibrio de clases y otros.
- Conocer las técnicas de resolución de problemas de clasificación con múltiples clases vía la descomposición.
- Conocer modelos avanzados de clasificación: multiclasificadores, máquinas de soporte vectorial y redes neuronales.

Sistemas Inteligentes para la Gestión de la Empresa

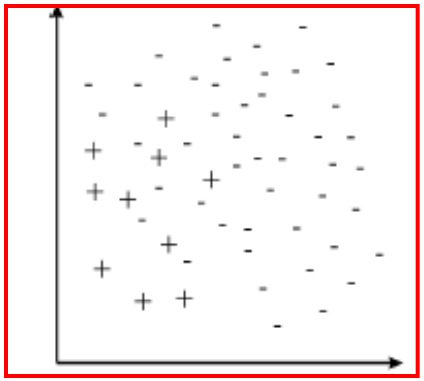
TEMA 3. Análisis Predictivo para la Empresa

(Modelos predictivos avanzados de clasificación)

1. Clasificación no balanceada
2. Multiclasificadores: Bagging y Boosting
3. Multiples clases: Descomposición binaria
4. Redes Neuronales y Maquinas de soporte Vectorial

Classification with Imbalanced Data Sets Presentation

In a concept-learning problem, the data set is said to present a class imbalance if it contains many more examples of one class than the other.



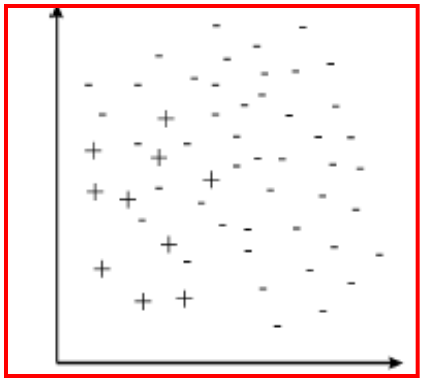
There exist many domains that do not have a balanced data set. There are a lot of problems where the most important knowledge usually resides in the minority class.

Ej.: Detection of uncommon diseases presents Imbalanced data:
Few sick persons and lots of healthy persons.

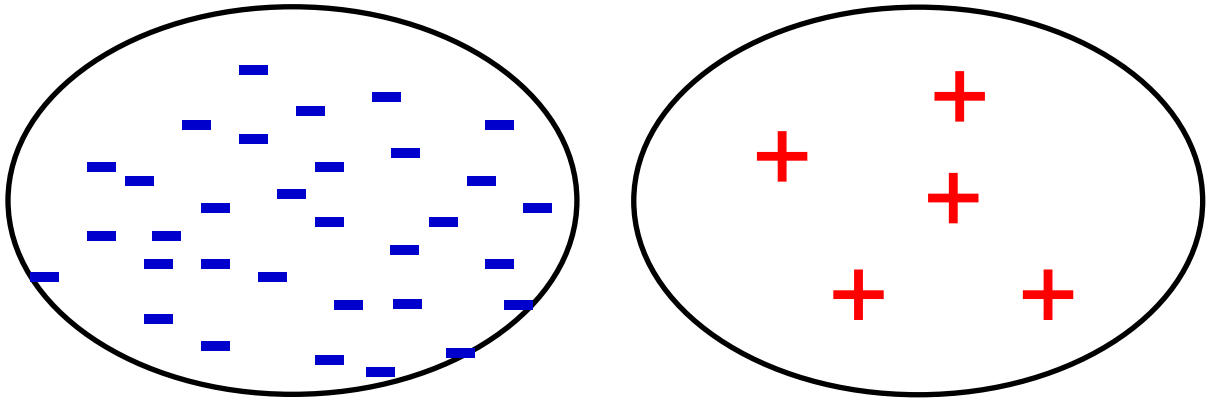
Some real-problems: Fraudulent credit card transactions, Learning word pronunciation, Prediction of telecommunications equipment failures, Detection oil spills from satellite images, Detection of Melanomas, Intrusion detection, Insurance risk modeling, Hardware fault detection

Classification with Imbalanced Data Sets Presentation

Such a situation introduce challenges for typical classifiers (such as decision tree) "systems that are designed to optimize overall accuracy without taking into account the relative distribution of each class".

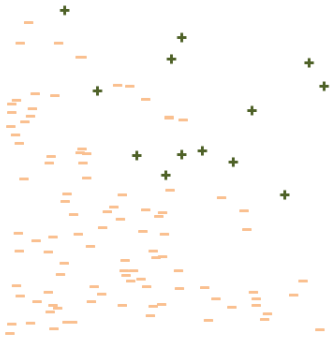


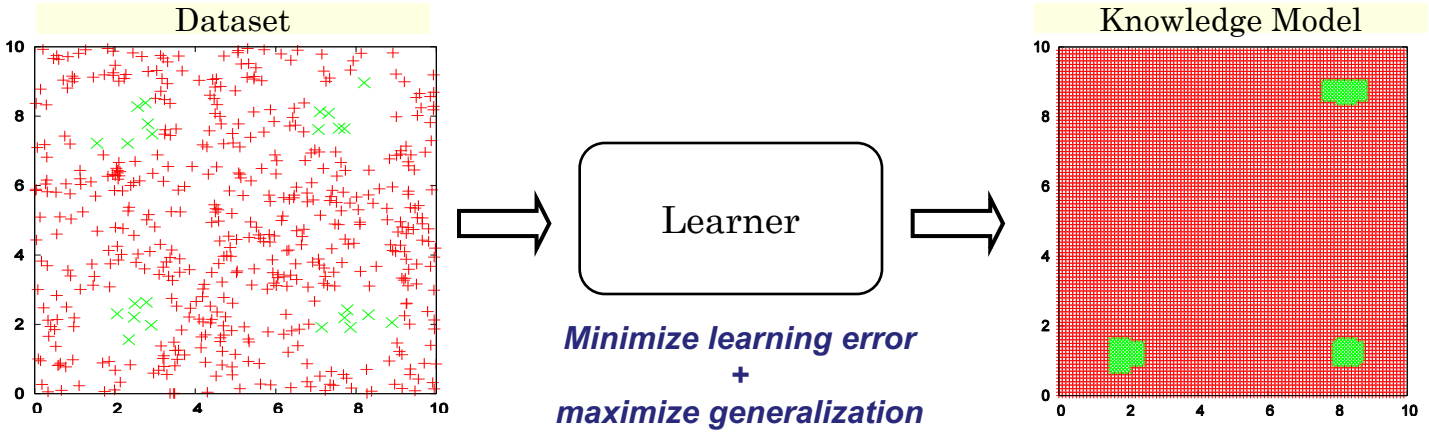
As a result, these classifiers tend to ignore small classes while concentrating on classifying the large ones accurately.



Why learning from imbalanced data-sets might be difficult?

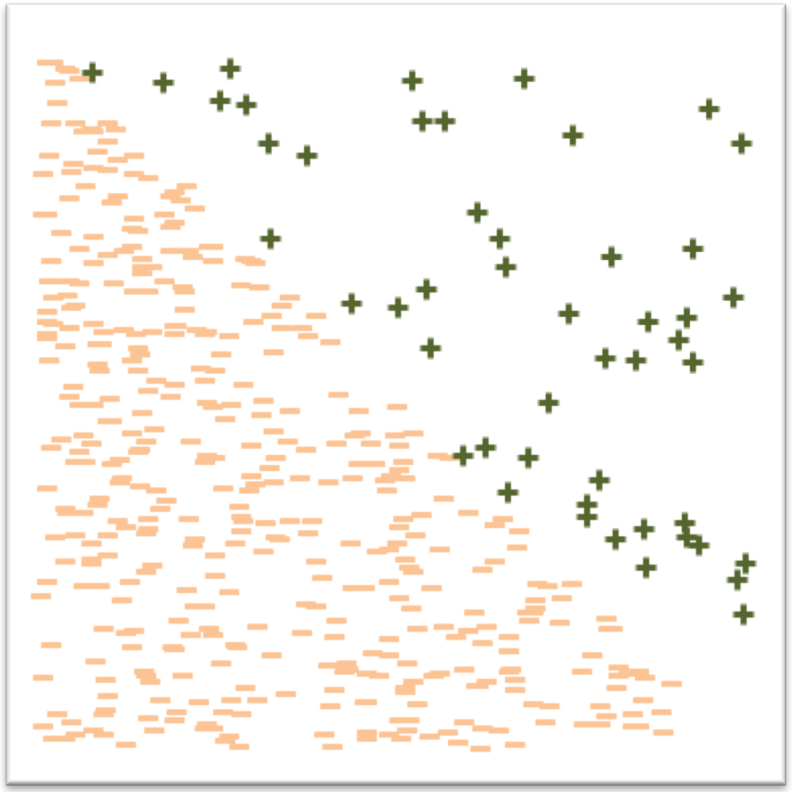
1. Search process guided by global error rates.
2. Classification rules over the positive class are highly specialized.
3. Classifiers tend to ignore small classes concentrating on classifying large ones accurately





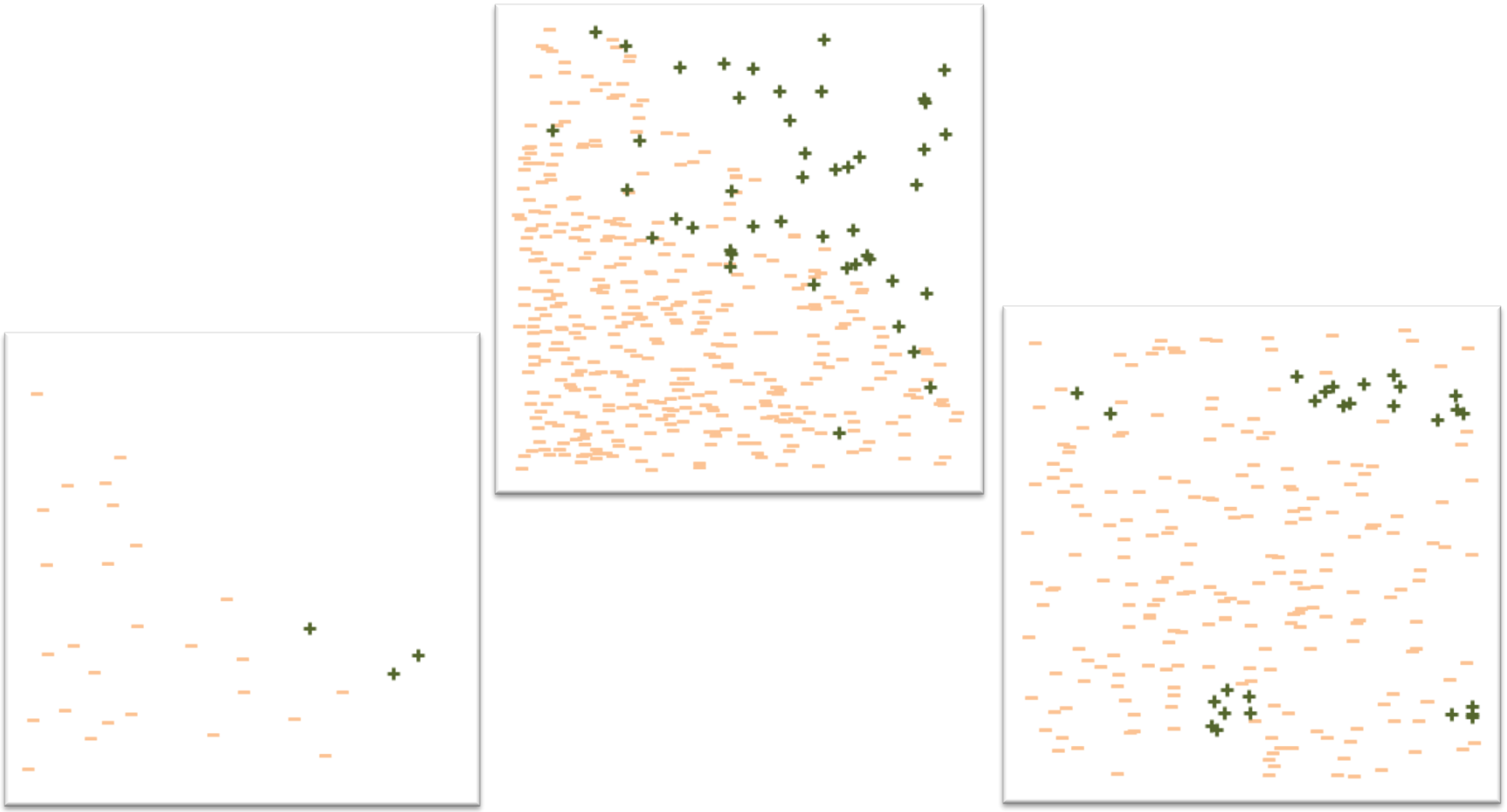
Why learning from imbalanced data-sets might be difficult?

- Skewed class distribution:
  - Measured by the fraction between majority and minority samples
  - Imbalance ratio (IR)
- **Intrinsic Data Characteristics**
  - Not only imbalance hinders classification performance
  - IR ≈ 9



V. López, A. Fernandez, S. García, V. Palade, F. Herrera, An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics. Information Sciences 250 (2013) 113-141

Why learning from imbalanced data-sets might be difficult?



V. López, A. Fernandez, S. García, V. Palade, F. Herrera, An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics. Information Sciences 250 (2013) 113-141

Contents

Contents

Introduction to Imbalanced Data Sets

Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

Resampling the original training set

Cost Modifying: Cost-sensitive learning

Ensembles to address class imbalance

Introduction to Imbalanced Data Sets

Some recent applications

- **Significance of the topic in recent applications**

  - **Tan, Shing Chiang; Watada, Junzo; Ibrahim, Zuwairie; et ál.; Evolutionary Fuzzy ARTMAP Neural Networks for Classification of Semiconductor Defects. IEEE Transactions on Neural Networks and Learning Systems 26 (5): 933-950 (MAY 2015)**
  - **Danenas, Paulius; Garsva, Gintautas; Selection of Support Vector Machines based classifiers for credit risk domain Experty Systems with Applications 42 (6) : 3194-3204 (APR 2015)**
  - **Liu, Nan; Koh, Zhi Xiong; Chua, Eric Chern-Pin; et ál..; Risk Scoring for Prediction of Acute Cardiac Complications from Imbalanced Clinical Data. IEEE Journal of Biomedical and Health Informatics 18 (6) : 1894-1902 (NOV 2014)**

Introduction to Imbalanced Data Sets

Some recent applications

- **Significance of the topic in recent applications**

  - **Radtke, Paulo V. W.; Granger, Eric; Sabourin, Robert; et ál..; Skew-sensitive boolean combination for adaptive ensembles - An application to face recognition in video surveillance Information Fusion 20: 31-48 (NOV 2014)**
  - **Yu, Hualong; Ni, Jun; An Improved Ensemble Learning Method for Classifying High-Dimensional and Imbalanced Biomedicine Data IEEE-ACM Transactions on Computational Biology and Bioinformatics 11(4) : 657-666 (AUG 2014)**
  - **Wang, Kung-Jeng; Makond, Bunjira; Chen, Kun-Huang; et ál.; A hybrid classifier combining SMOTE with PSO to estimate 5-year survivability of breast cancer patients. Applied Soft Computing 20: 15-24 (JUL 2014)**
  - **B. Krawczyk, M. Galar, L. Jelen, F. Herrera. Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. Applied Soft Computing 38 (2016) 714-726.**

Introduction to Imbalanced Data Sets

Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

Resampling the original training set

Cost Modifying: Cost-sensitive learning

Ensembles to address class imbalance

Introduction to Imbalanced Data Sets

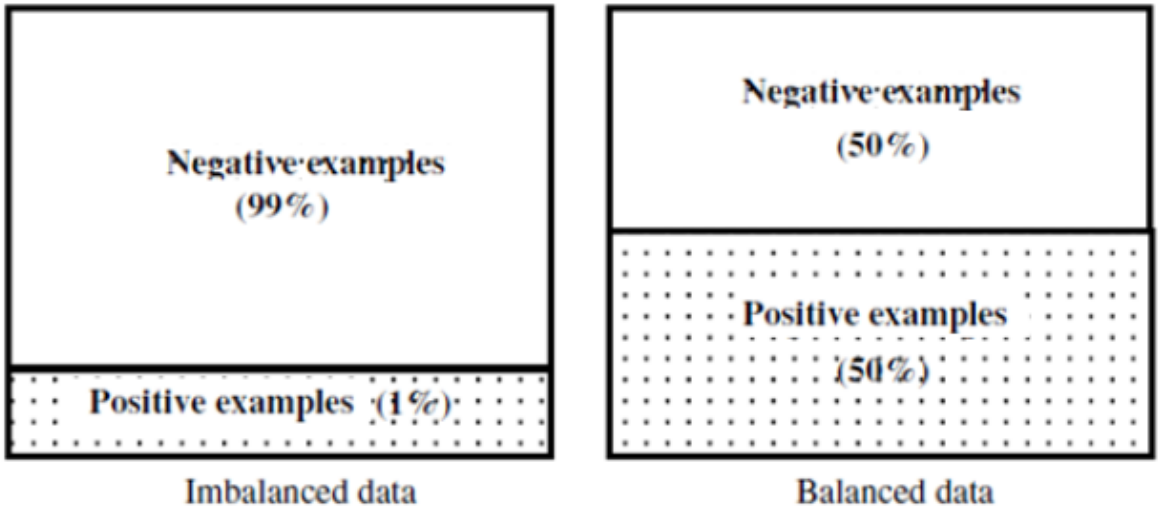


Fig. 1. Imbalanced and balanced data sets.

biased towards the majority class

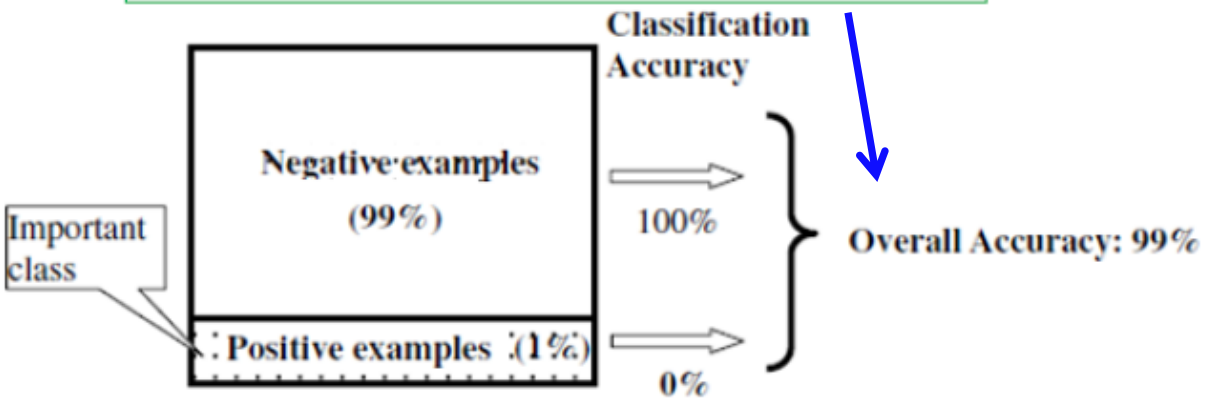


Fig. 2. The illustration of class imbalance problems.

Imbalanced classes problem: standard learners are often biased towards the majority class.

We need to change the way to evaluate a model performance!

Introduction to Imbalanced Data Sets

How can we evaluate an algorithm in imbalanced domains?

Confusion matrix for a two-class problem

	Positive Prediction	Negative Prediction
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

It doesn't take into account the False Negative Rate, which is very important in imbalanced problems

Classical evaluation:

Error Rate: (FP + FN)/N
Accuracy Rate: (TP + TN) /N

Introduction to Imbalanced Data Sets

Imbalanced evaluation based on the geometric mean:

Positive true ratio: $a^+ = TP/(TP+FN)$

Negative true ratio: $a^- = TN / (FP+TN)$

Evaluation function: **True ratio**

$$g = \sqrt{(a^+ \cdot a^-)}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

Precision = TP/(TP+FP)

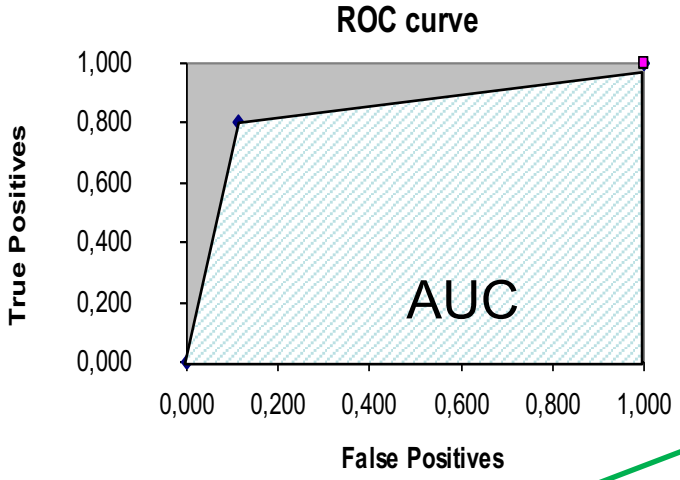
Recall = TP/(TP+FN)

F-measure: (2 x precision x recall) / (recall + precision)

R. Barandela, J.S. Sánchez, V. García, E. Rangel. Strategies for learning in class imbalance problems. Pattern Recognition 36:3 (2003) 849-851

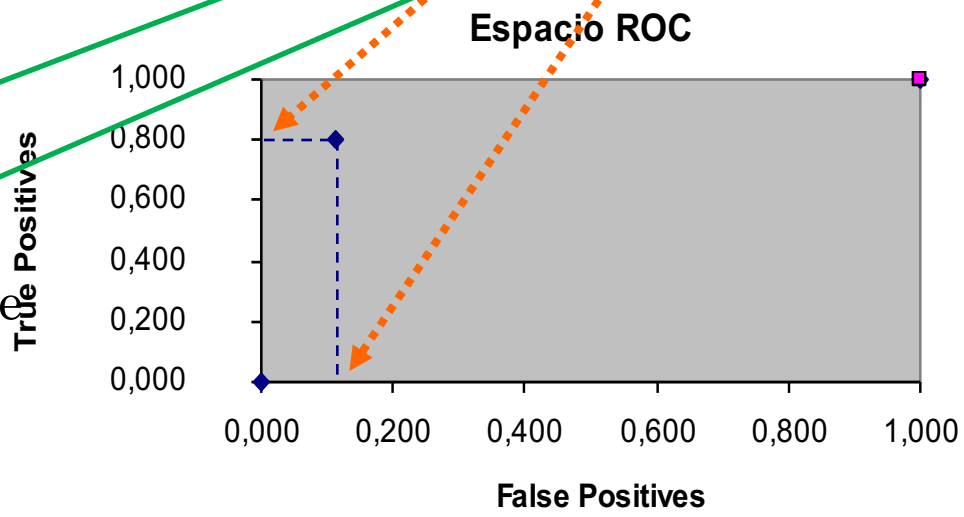
Introduction to Imbalanced Data Sets

AUC: Area under ROC curve. Scalar quantity widely used for estimating classifiers performance.



Real

Pred	PP	NP
PC	0,8	0,121
NC	0,2	0,879



$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2}$$

Introduction to Imbalanced Data Sets

Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

Resampling the original training set

Cost Modifying: Cost-sensitive learning

Ensembles to address class imbalance

Introduction to Imbalanced Data Sets

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- Random
- Focused

Under-Sampling
- Random
- Focused

Retain influential examples

Balance the training set

Remove noisy instances in the decision boundaries

Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

Introduction to Imbalanced Data Sets

Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

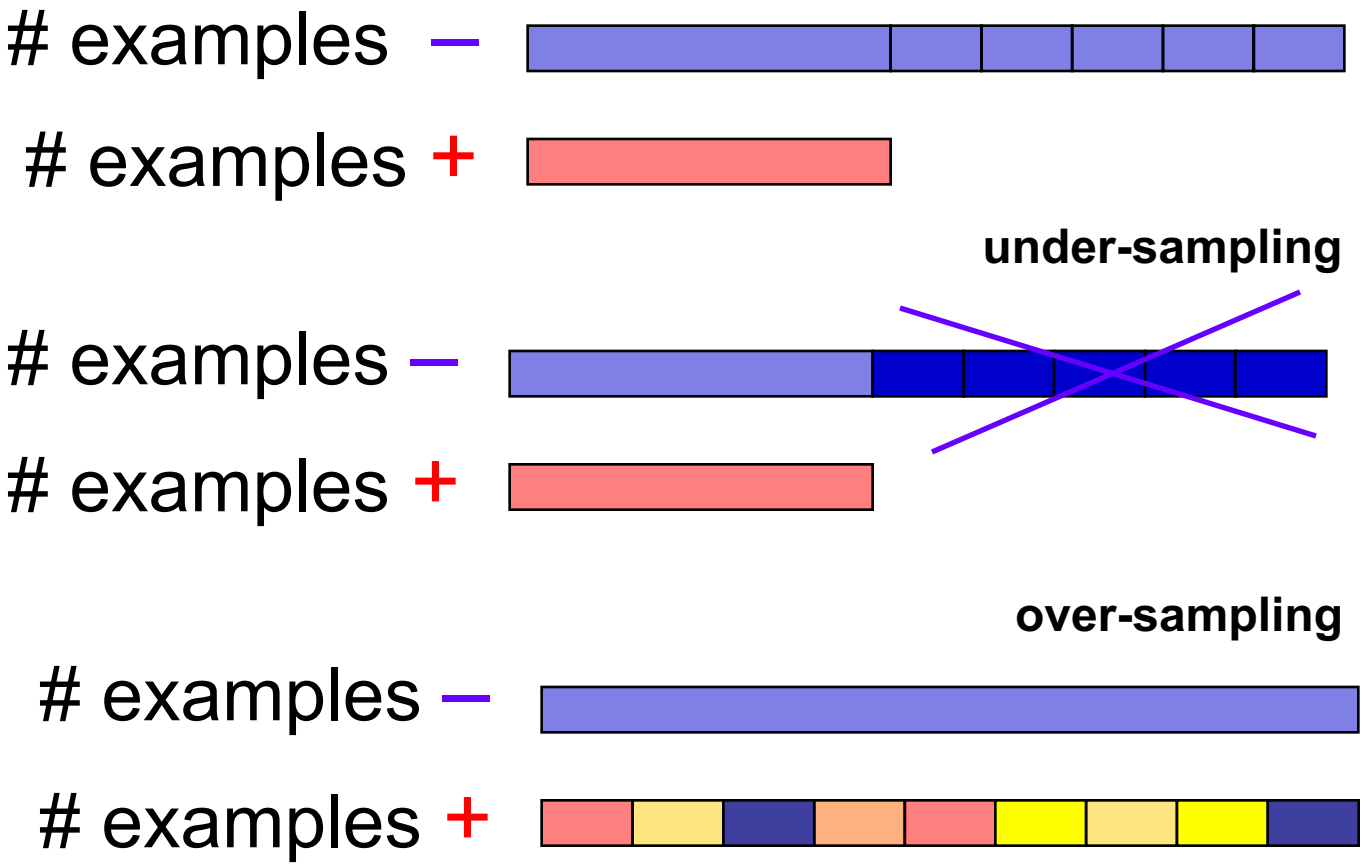
Resampling the original training set

Cost Modifying: Cost-sensitive learning

Ensembles to address class imbalance

Resampling the original data sets

Undersampling vs oversampling



Resampling the original data sets

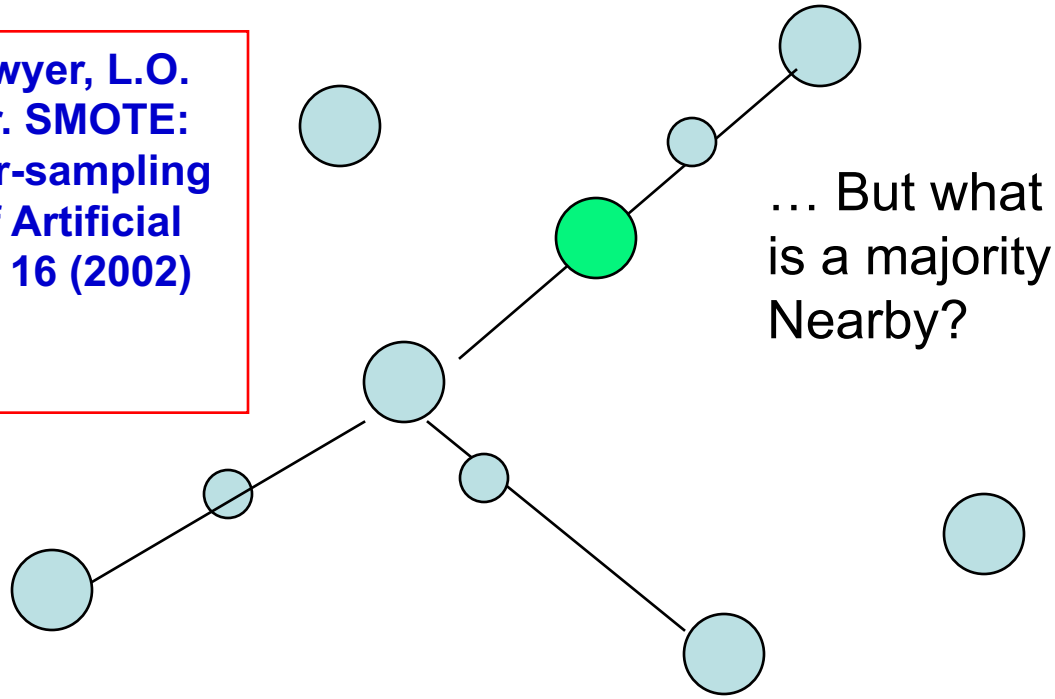
Oversampling: Replicating examples

SMOTE: Instead of replicating, let us invent some new instances.

Resampling the original data sets

Oversampling: State-of-the-art algorithm, SMOTE

N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16 (2002) 321-357



… But what if there is a majority sample Nearby?

: Minority sample

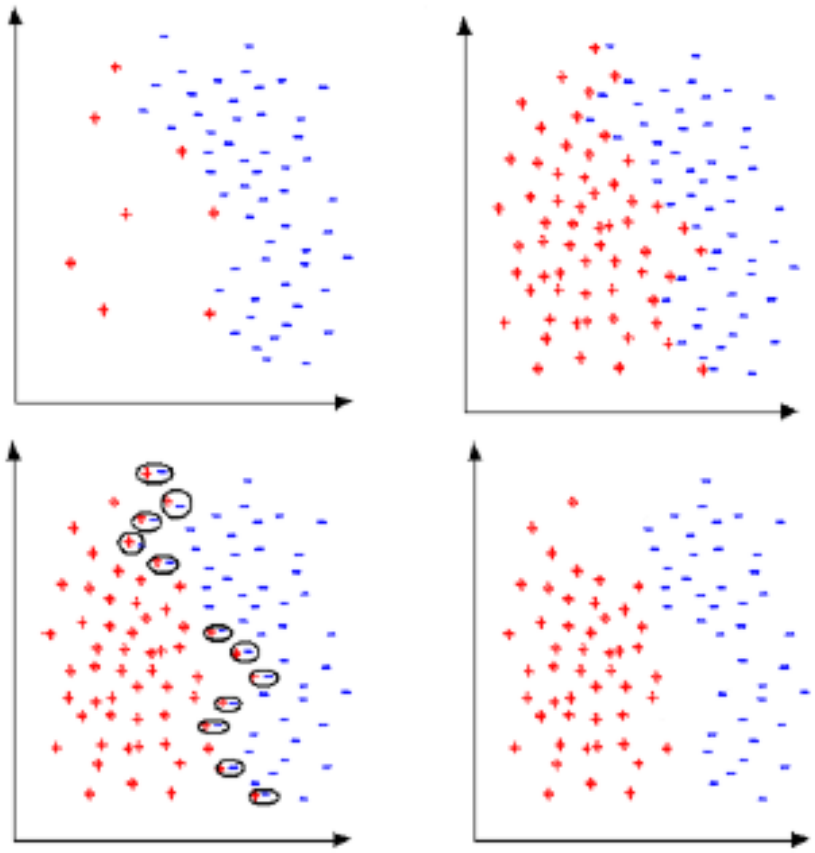
: Synthetic sample

: Majority sample

Resampling the original data sets

SMOTE hybridization: SMOTE + Tomek links

Figure: SMOTE+TomekLink



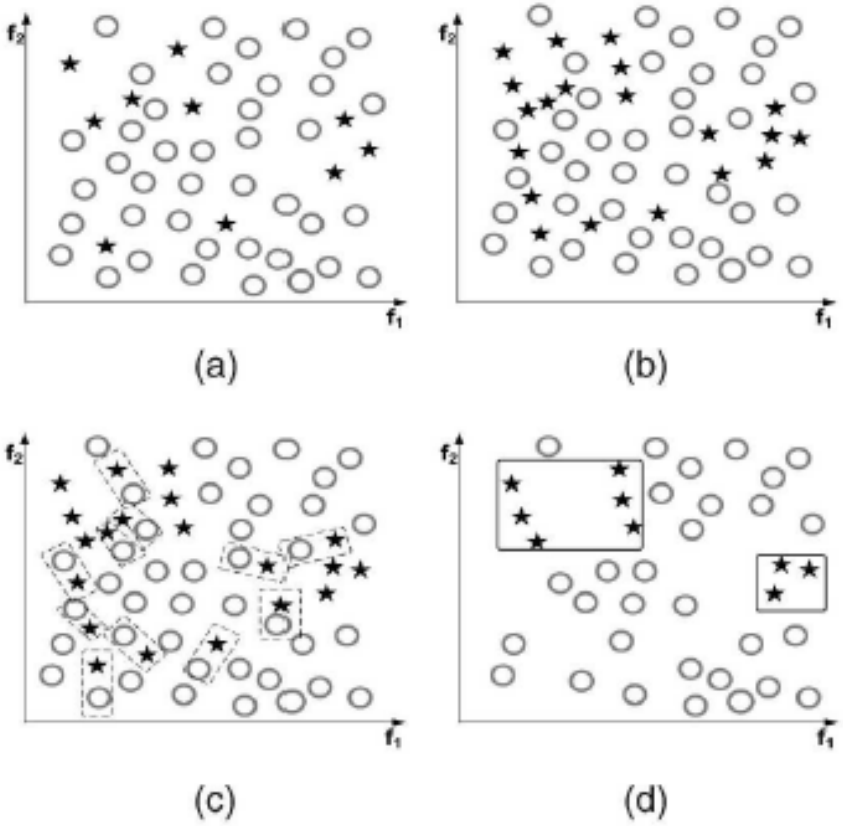


Figure 17: (a) Original data-set distribution. (b) Post-SMOTE data-set. (c) The identified Tomek Links. (d) The data-set after removing Tomek links

Resampling the original data sets

SMOTE and hybridization: Analysis

Table 6: Performance ranking for original and balanced data sets for pruned decision trees.

Data set	1°	2°	3°	4°	5°	6°	7°	8°	9°	10°	11°
Pima	Smt	RdOvr	Smt+Tmk	Smt+ENN	Tmk	NCL	Original	RdUdr	CNN+Tmk	CNN*	OSS*
German	RdOvr	Smt+Tmk	Smt+ENN	Smt	RdUdr	CNN	CNN+Tmk*	OSS*	Original*	Tmk*	NCL*
Post-operative	RdOvr	Smt+ENN	Smt	Original	CNN	RdUdr	CNN+Tmk	OSS*	Tmk*	NCL*	Smt+Tmk*
Haberman	Smt+ENN	Smt+Tmk	Smt	RdOvr	NCL	RdUdr	Tmk	OSS*	CNN*	Original*	CNN+Tmk*
Splice-ie	RdOvr	Original	Tmk	Smt	CNN	NCL	Smt+Tmk	Smt+ENN*	CNN+Tmk*	RdUdr*	OSS*
Splice-ei	Smt	Smt+Tmk	Smt+ENN	CNN+Tmk	OSS	RdOvr	Tmk	CNN	NCL	Original	RdUdr
Vehicle	RdOvr	Smt	Smt+Tmk	OSS	CNN	Original	CNN+Tmk	Tmk	NCL*	Smt+ENN*	RdUdr*
Letter-vowel	Smt+ENN	Smt+Tmk	Smt	RdOvr	Tmk*	NCL*	Original*	CNN*	CNN+Tmk*	RdUdr*	OSS*
New-thyroid	Smt+ENN	Smt+Tmk	Smt	RdOvr	RdUdr	CNN	Original	Tmk	CNN+Tmk	NCL	OSS
E.Coli	Smt+Tmk	Smt	Smt+ENN	RdOvr	NCL	Tmk	RdUdr	Original	OSS	CNN+Tmk*	CNN*
Satimage	Smt+ENN	Smt	Smt+Tmk	RdOvr	NCL	Tmk	Original*	OSS*	CNN+Tmk*	RdUdr*	CNN*
Flag	RdOvr	Smt+ENN	Smt+Tmk	CNN+Tmk	Smt	RdUdr	CNN*	OSS*	Tmk*	Original*	NCL*
Glass	Smt+ENN	RdOvr	NCL	Smt	Smt+Tmk	Original	Tmk	RdUdr	CNN+Tmk*	OSS*	CNN*
Letter-a	Smt+Tmk	Smt+ENN	Smt	RdOvr	OSS	Original	Tmk	CNN+Tmk	NCL	CNN	RdUdr*
Nursery	RdOvr	Tmk	Original	NCL	CNN*	OSS*	Smt+Tmk*	Smt*	CNN+Tmk*	Smt+ENN*	RdUdr*

G.E.A.P.A. Batista, R.C. Prati, M.C. Monard. A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations 6:1 (2004) 20-29

Resampling the original data sets

Other SMOTE hybridizations

Safe_Level_SMOTE: C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap. Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-09). LNAI 5476, Springer-Verlag 2005, Bangkok (Thailand, 2009) 475-482

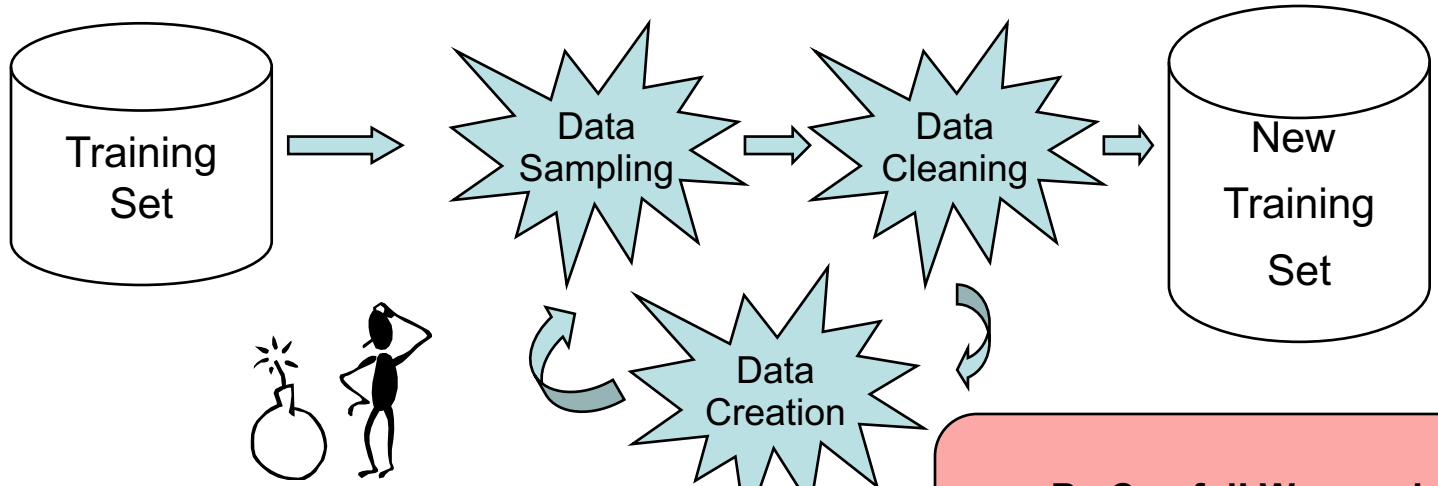
Borderline_SMOTE: H. Han, W.Y. Wang, B.H. Mao. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. International Conference on Intelligent Computing (ICIC'05). Lecture Notes in Computer Science 3644, Springer-Verlag 2005, Hefei (China, 2005) 878-887

SMOTE_LLE: J. Wang, M. Xu, H. Wang, J. Zhang. Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding. IEEE 8th International Conference on Signal Processing, 2006.

LN-SMOTE: T. Maciejewski and J. Stefanowski. Local Neighbourhood Extension of SMOTE for Mining Imbalanced Data. IEEE SSCI , Paris, CIDM , 2011.

SMOTE-RSB: E. Ramentol, Y. Caballero, R. Bello, F. Herrera, SMOTE-RSB*: A Hybrid Preprocessing Approach based on Oversampling and Undersampling for High Imbalanced Data-Sets using SMOTE and Rough Sets Theory. *Knowledge and Information Systems 33:2 (2012) 245-265.*

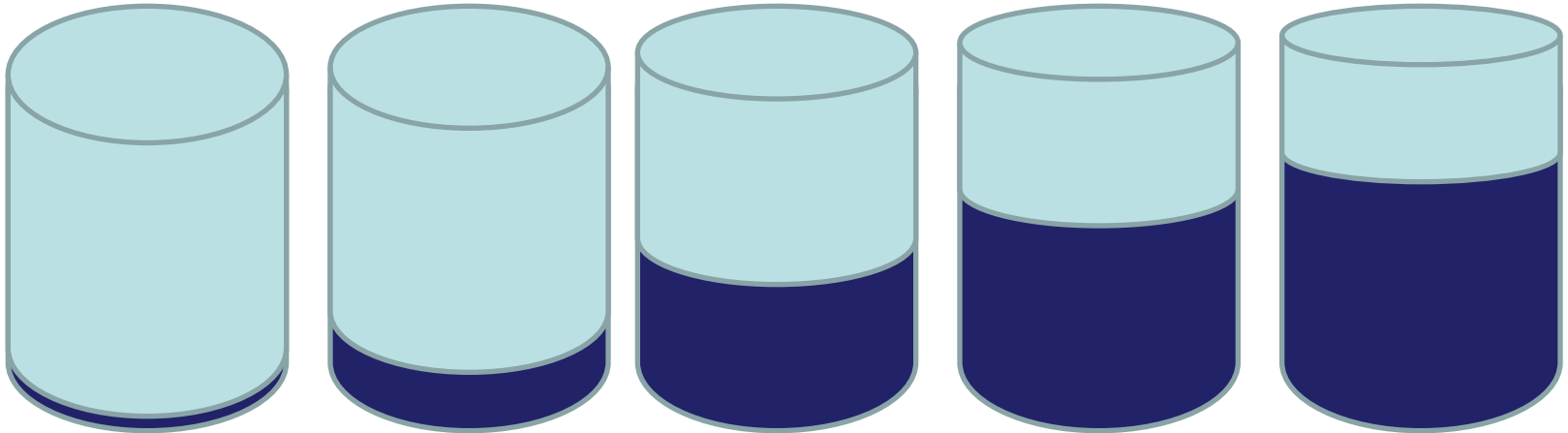
Resampling the original data sets
Final comments



Class 0

Class 1

Be Careful! We are changing what we were supposed to learn! We change the data distribution!



Introduction to Imbalanced Data Sets

Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

Resampling the original training set

Cost Modifying: Cost-sensitive learning

Ensembles to address class imbalance

Cost-sensitive learning

Cost modification consists of weighting errors made on examples of the minority class higher than those made on examples of the majority class in the calculation of the training error.

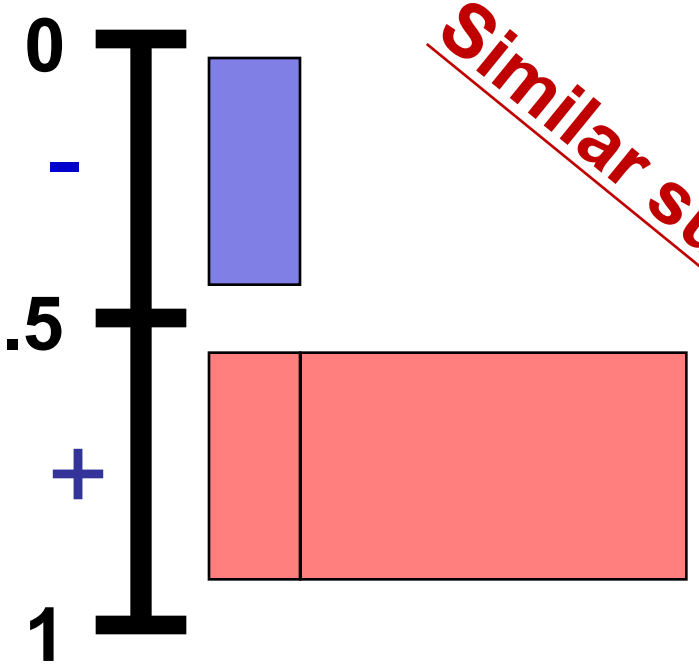
Over Sampling

- Random
- Focused

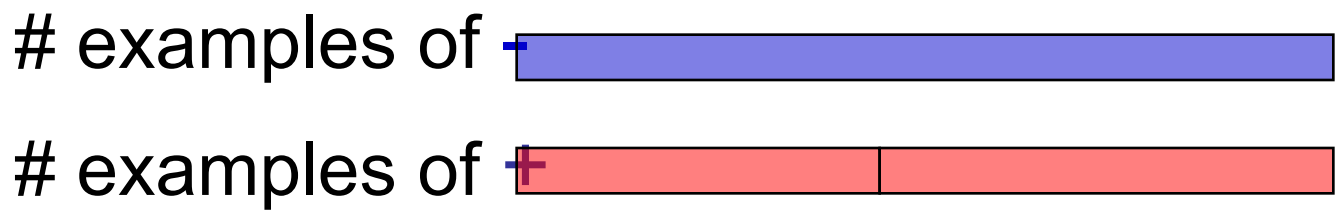
Under Sampling

- Random
- Focused

Cost Modifying



Similar sum of costs



Cost-sensitive learning

Results and Statistical Analysis

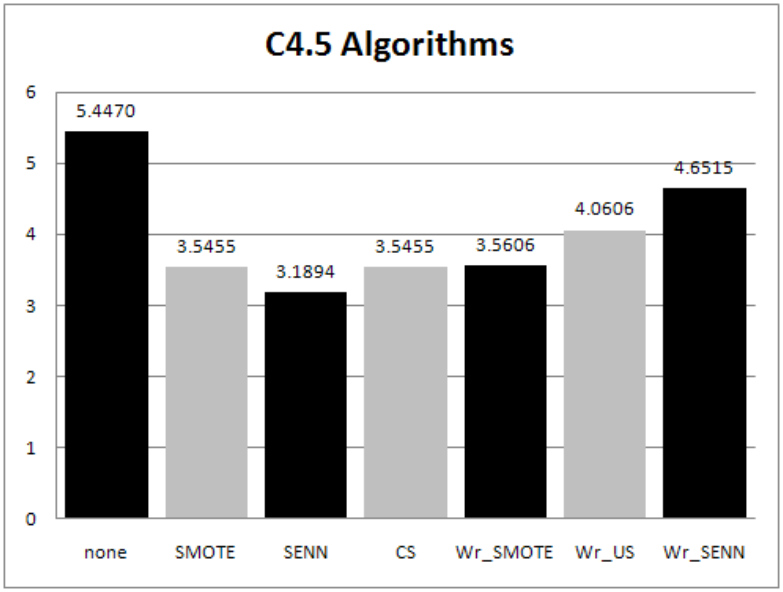
- **Case of Study: C4.5**
- **Similar results and conclusions for the remaining classification paradigms**

Algorithm	AUC_{tr}	AUC_{tst}
C45	0.8774 ± 0.0392	0.7902 ± 0.0804
C45 SMOTE	0.9606 ± 0.0142	0.8324 ± 0.0728
C45 SENN	0.9471 ± 0.0154	$\mathbf{0.8390 \pm 0.0772}$
C45CS	0.9679 ± 0.0103	0.8294 ± 0.0758
C45 Wr_SMOTE	0.9679 ± 0.0103	0.8296 ± 0.0763
C45 Wr_US	0.9635 ± 0.0139	0.8245 ± 0.0760
C45 Wr_SENN	0.9083 ± 0.0377	0.8145 ± 0.0712

V. López, A. Fernandez, J. G. Moreno-Torres, F. Herrera, **Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics**. *Expert Systems with Applications 39:7 (2012) 6585-6608.*

Cost-sensitive learning

Results and Statistical Analysis



- **Rankings obtained by Friedman test for the different approaches of C4.5.**
- **Shaffer test as post-hoc to detect statistical differences ($\alpha = 0.05$)**

C4.5	none	SMOTE	SENN	CS	Wr_SMOTE	Wr_US	Wr_SENN
none	x	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SMOTE	+(6.404E-6)	x	=(1.0)	=(1.0)	=(1.0)	=(1.0)	+(.04903)
SENN	+(4.058E-8)	=(1.0)	x	=(1.0)	=(1.0)	=(.22569)	+(.00152)
CS	+(6.404E-6)	=(1.0)	=(1.0)	x	=(1.0)	=(1.0)	+(.04903)
Wr_SMOTE	+(7.904E-6)	=(1.0)	=(1.0)	=(1.0)	x	=(1.0)	+(.04903)
Wr_US	+(.00341)	=(1.0)	=(.22569)	=(1.0)	=(1.0)	x	=(1.0)
Wr_SENN	=(.37846)	-(.04903)	-(.00152)	-(.04903)	-(.04903)	=(1.0)	x

Cost-sensitive learning
Final comments

- **Preprocessing and cost-sensitive learning improve the base classifier.**
- **No differences among the different preprocessing techniques.**
- **Both preprocessing and cost-sensitive learning are good and equivalent approaches to address the imbalance problem.**
- **In most cases, the preliminary versions of hybridization techniques do not show a good behavior in contrast to standard preprocessing and cost sensitive.**

Some authors claim: **"Cost-Adjusting is slightly more effective than random or directed over- or under- sampling although all approaches are helpful, and directed oversampling is close to cost-adjusting". Our study shows similar results.**

V. López, A. Fernandez, J. G. Moreno-Torres, F. Herrera, **Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics**. *Expert Systems with Applications 39:7 (2012) 6585-6608.*

Introduction to Imbalanced Data Sets

Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

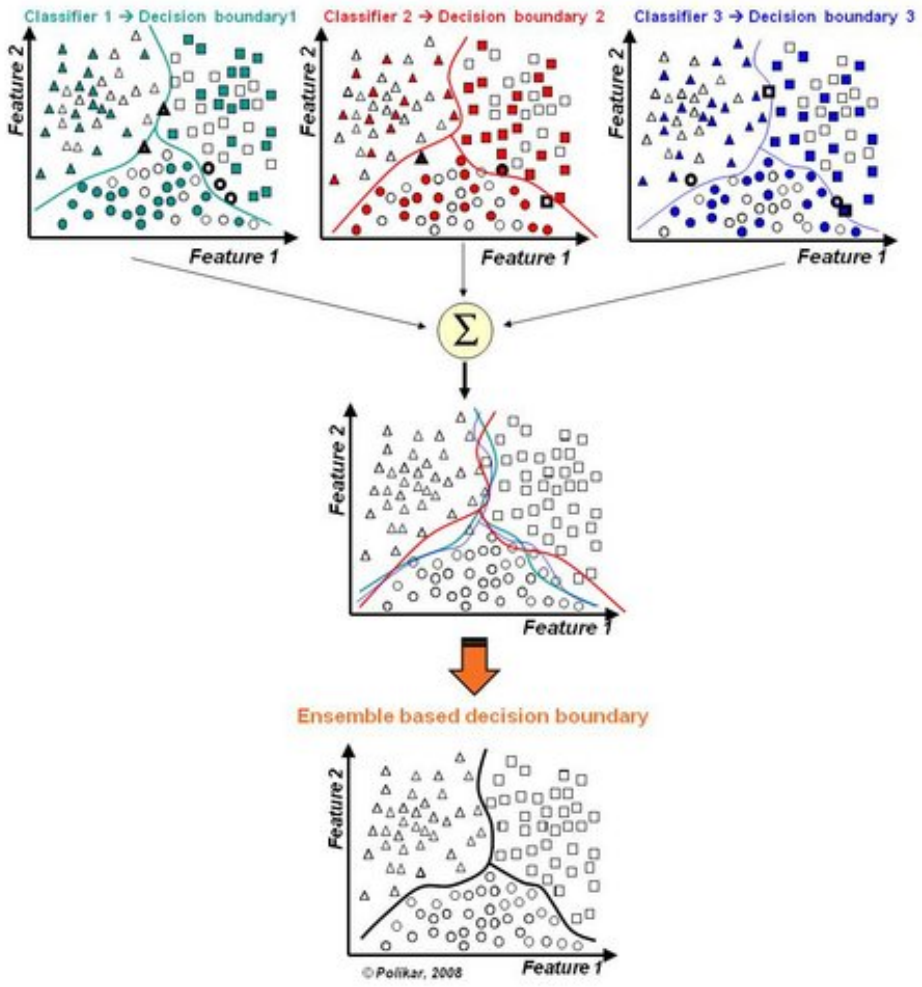
Resampling the original training set

Cost Modifying: Cost-sensitive learning

Ensembles to address class imbalance

Emsembles to address class imbalance

Ensemble-based classifiers try to improve the performance of single classifiers by inducing several classifiers and combining them to obtain a new classifier that outperforms every one of them.
Hence, the basic idea is to construct several classifiers from the original data and then aggregate their predictions when unknown instances are presented.
This idea follows human natural behavior which tend to seek several opinions before making any important decision.



Emsembles to address class imbalance

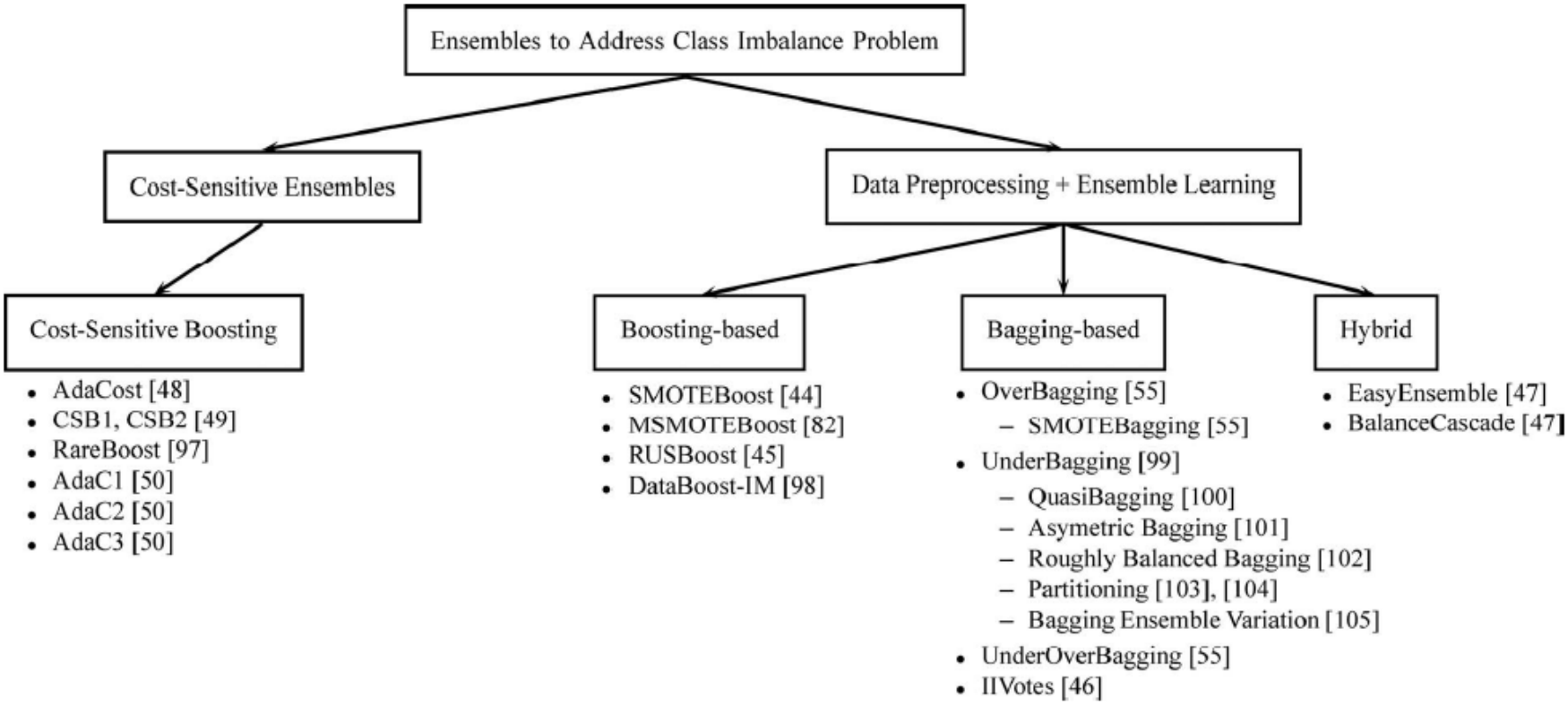


Fig. 3. Proposed taxonomy for ensembles to address the class imbalance problem.

M. Galar, A. Fernández, F. E. Barrenechea, H. Bustince, F. Herrera. A Review on Ensembles for Class Imbalance Problem: Bagging, Boosting and Hybrid Based Approaches. IEEE TSMC-Par C 42:4 (2012) 463-484

Emsembles to address class imbalance

TABLE XV
REPRESENTATIVE METHODS SELECTED FOR EACH FAMILY

Family	Abbr.	Method
Non-ensembles	SMT	SMOTE
Classic	M14	AdaBoost.M2 ($T = 40$)
Cost-sensitive	C24	AdaC2 ($T = 40$)
Boosting-based	RUS1	RUSBoost ($T = 10$)
Bagging-based	SBAG4	SMOTEBagging ($T = 40$)
Hybrids	EASY	EasyEnsemble

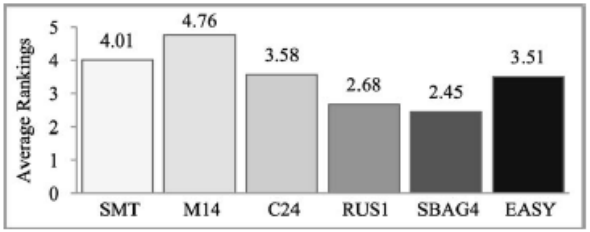


Fig. 9. Average rankings of the representatives of each family.

TABLE XVIII
SHAFFER TESTS FOR INTERFAMILY COMPARISON

	SMT	M14	C24	RUS1	SBAG4	EASY
SMT	×	=(0.24024)	=(1.0)	-(0.00858)	-(0.00095)	=(1.0)
M14	=(0.24024)	×	-(0.03047)	-(0.0)	-(0.0)	-(0.01725)
C24	=(1.0)	+(0.03047)	×	=(0.17082)	-(0.03356)	=(1.0)
RUS1	+(0.00858)	+(0.0)	=(0.17082)	×	=(1.0)	=(0.22527)
SBAG4	+(0.00095)	+(0.0)	+(0.03356)	=(1.0)	×	=(0.05641)
EASY	+(0.01725)	=(1.0)	=(1.0)	=(0.22527)	=(0.05641)	×

TABLE XVI
HOLM TABLE FOR BEST INTERFAMILY ANALYSIS

i	Algorithm (Rank)	Z	p-value	Holm	Hypothesis ($\alpha = 0.05$)
5	M14 (4.76)	5.78350	0.00000	0.01	**Rejected for SBAG4**
4	SMT (4.01)	3.90315	0.00009	0.0125	**Rejected for SBAG4**
3	C24 (3.58)	2.82052	0.00479	0.01667	**Rejected for SBAG4**
2	EASY (3.51)	2.64958	0.00806	0.025	**Rejected for SBAG4**
1	RUS1 (2.68)	0.56980	0.56881	0.05	Not Rejected

Control method : SBAG4, Rank :2.45.

TABLE XVII
WILCOXON TESTS TO SHOW DIFFERENCES BETWEEN SBAG4 AND RUS1

Comparison	R^+	R^-	Hypothesis($\alpha = 0.05$)	p-value
SBAG4 vs. RUS1	527.5	462.5	Not Rejected	0.71717

R^+ are ranks for SBAG4 and R^- for RUS1.

Emsembles to address class imbalance

Our proposal:

We develop a new ensemble construction algorithm (**EUSBoost**) based on RUSBoost, one of the simplest and most accurate ensemble, combining random undersampling with Boosting algorithm.

Our methodology aims to improve the existing proposals enhancing the performance of the base classifiers by the usage of the evolutionary undersampling approach.

Besides, we promote diversity favoring the usage of different subsets of majority class instances to train each base classifier.

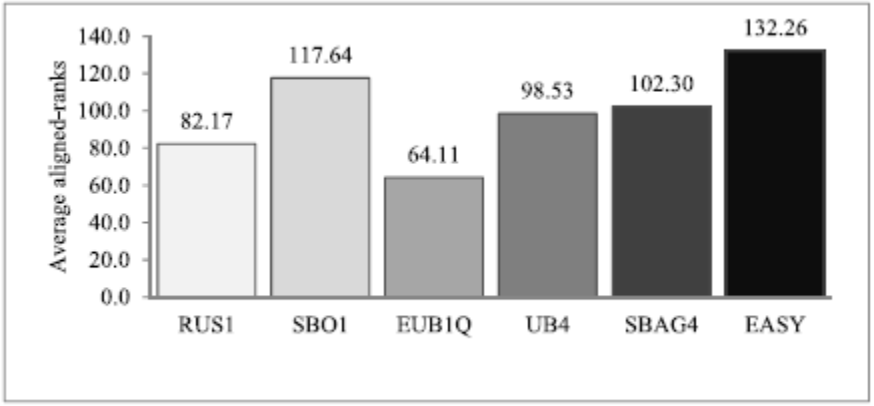


Figure: Average aligned-ranks of the comparison between EUSBoost and the state-of-the-art ensemble methods.

M. Galar, A. Fernandez, E. Barrenechea, F. Herrera, **EUSBoost: Enhancing Ensembles for Highly Imbalanced Data-sets by Evolutionary Undersampling**. *Pattern Recognition 46:12 (2013) 3460–3471*

Emsembles to address class imbalance

Final comments

- Ensemble-based algorithms are worthwhile, improving the results obtained by using data preprocessing techniques and training a single classifier.
- The use of more classifiers make them more complex, but this growth is justified by the better results that can be assessed.
- We have to remark the good performance of approaches such as RUSBoost or SmoteBagging, which despite of being simple approaches, achieve higher performance than many other more complex algorithms.
- We have shown the positive synergy between sampling techniques (e.g., undersampling or SMOTE) and Bagging ensemble learning algorithm.

Contents

I. Introduction to imbalanced data sets

II. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics

!Challenges on class distribution¡

I. Class imbalance: Data sets, implementations, …

II. Class imbalance: Trends and final comments

Why is difficult to learn in imbalanced domains?

- **Preprocessing and cost sensitive learning have a similar behavior.**
- **Performance can still be improved, but we must analyze in deep the nature of the imbalanced data-set problem:**
  - **Imbalance ratio is not a determinant factor**

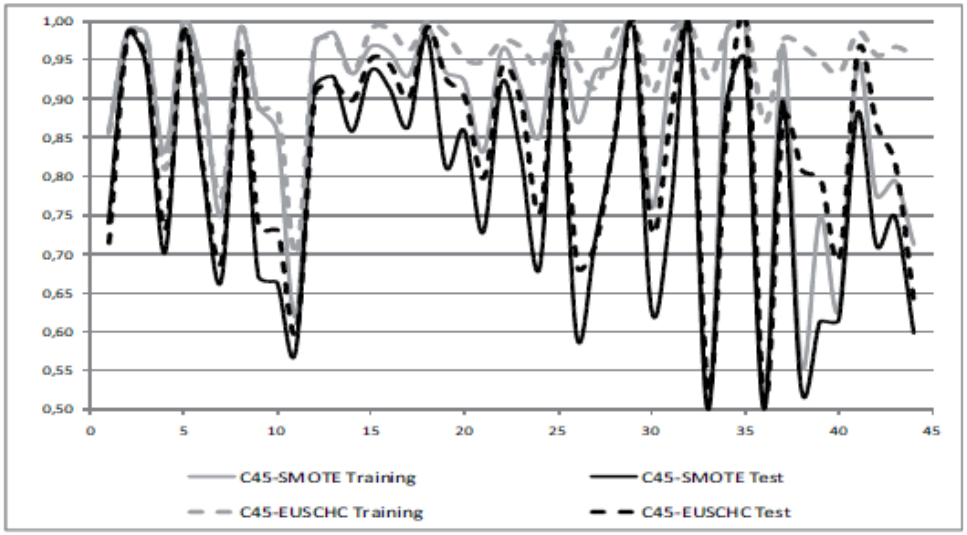
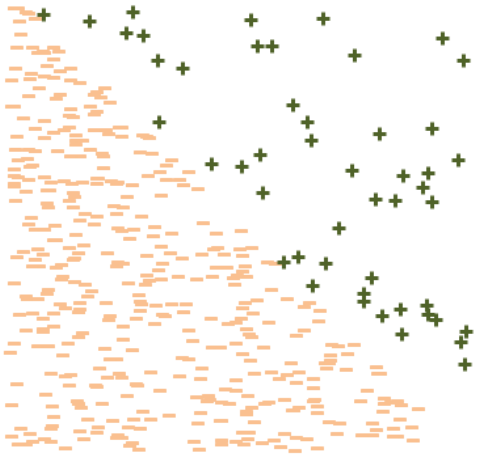


Fig. 4 C4.5 AUC in Training/Test sorted by IR

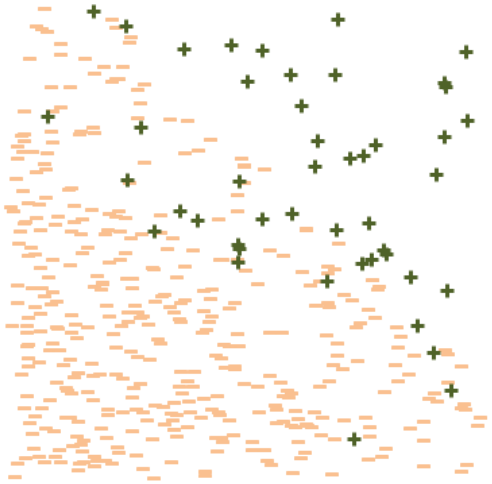
J. Luengo, A. Fernández, S. García, and F. Herrera. Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling. *Soft Computing 15 (2011) 1909-1936*, doi: 10.1007/s00500-010-0625-8.

Introduction to Imbalanced Data Sets

Why is difficult to learn in imbalanced domains?



Imbalance – why is it difficult?



An easier problem

More difficult one

Some of sources of difficulties:

- Overlapping,
- Small disjuncts,
- Lack of data,
- …

Majority classes overlaps the minority class:

- Ambiguous boundary between classes
- Influence of noisy examples
- Difficult border, …

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

Small disjuncts/rare data sets

Density: Lack of data

Bordeline and Noise data

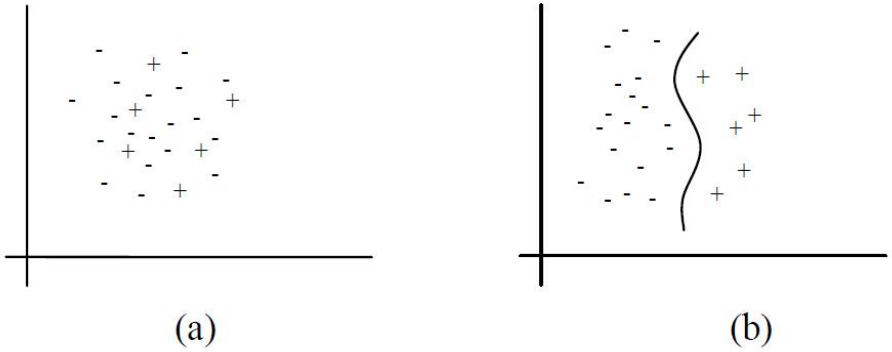
Dataset shift

V. López, A. Fernandez, S. García, V. Palade, F. Herrera, **An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics.** Information Sciences 250 (2013) 113-141.

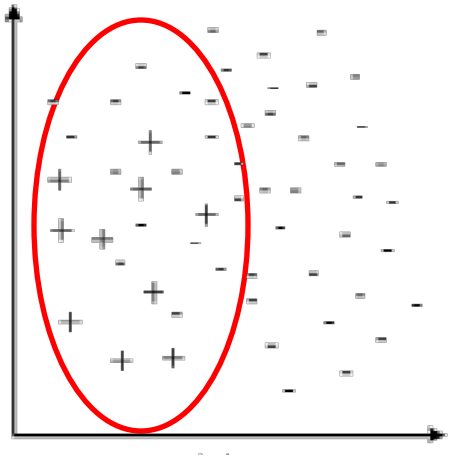
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Class imbalance is not the only responsible of the lack in accuracy of an algorithm.



The class overlapping also influences the behaviour of the algorithms, and it is very typical in these domains.

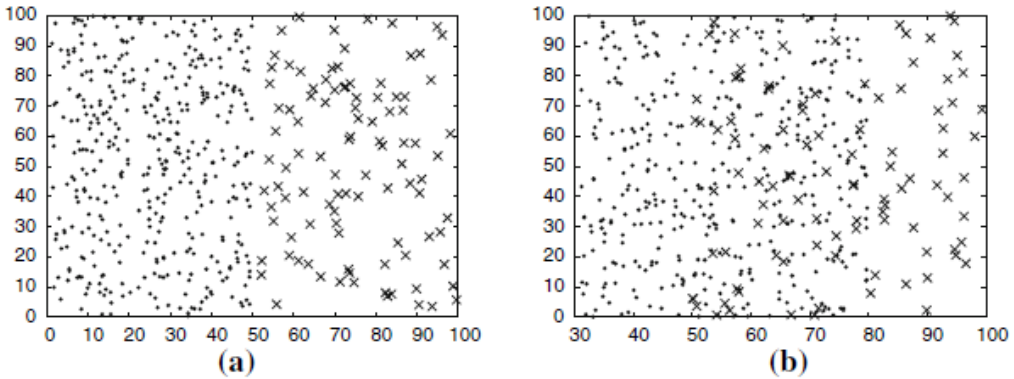


V. García, R.A. Mollineda, J.S. Sánchez. On the k-NN performance in a challenging scenario of imbalance and overlapping. Pattern Anal Applic (2008) 11: 269-280

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

- **There is an interesting relationship between imbalance and class overlapping:**



Two different levels of class overlapping: (a) 0% and (b) 60%

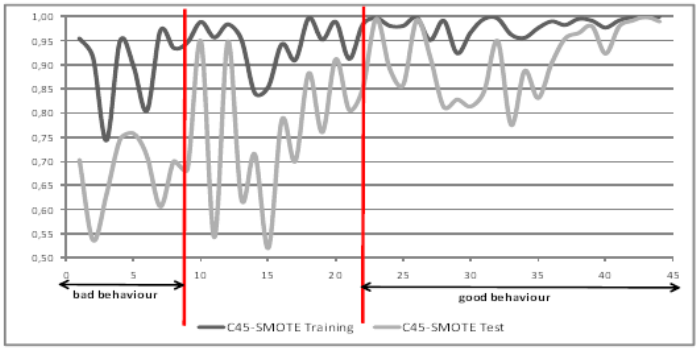


Fig. 6 C4.5 AUC with SMOTE in Training/Test sorted by F1

F1: maximum Fisher's discriminant ratio.

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

V. García, R.A. Mollineda, J.S. Sánchez. On the k-NN performance in a challenging scenario of imbalance and overlapping. Pattern Anal Applic (2008) 11: 269-280

J. Luengo, A. Fernandez, S. García, F. Herrera, Addressing Data Complexity for Imbalanced Data Sets: Analysis of SMOTE-based Oversampling and Evolutionary Undersampling. *Soft Computing, 15 (10) 1909-1936*

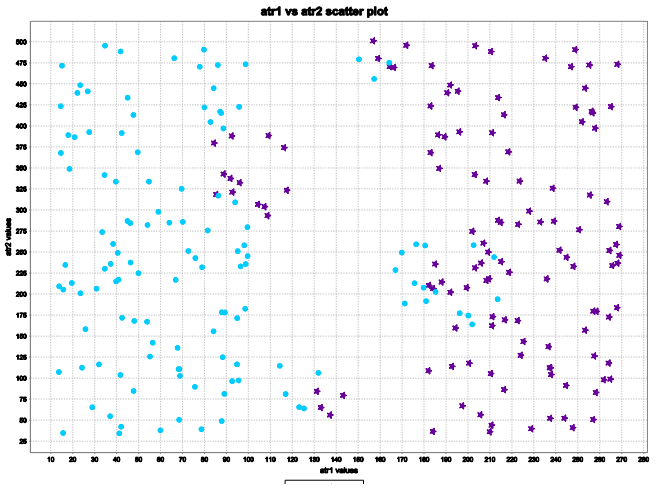
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

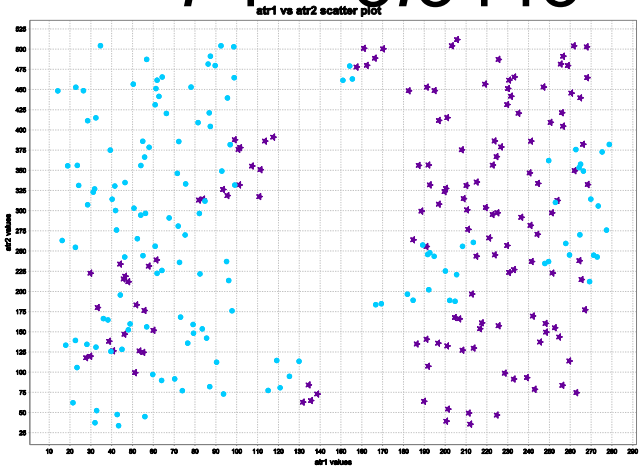
- **The degree of overlap for individual feature values is measured by the metric *F1* or *maximum Fisher's discriminant ratio***

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

- **We compute *f* for each feature and take the maximum over all dimensions as metric *F1*.**



*F*1 = 3.3443



*F*1 = 0.6094

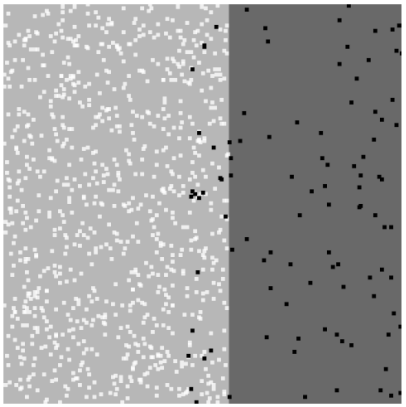
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

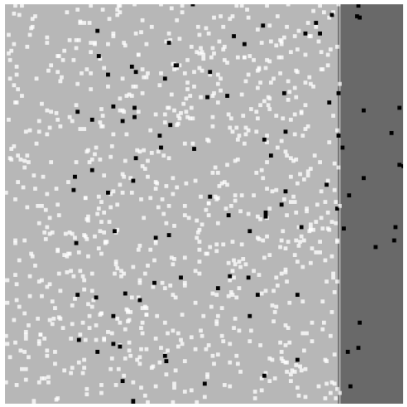
- **There is an interesting relationship between imbalance and class overlapping:**

Table 13 Performance obtained by C4.5 with different degrees of overlap

Overlap Degree	TP_{rate}	TN_{rate}	AUC
0 %	1.000	1.000	1.000
20 %	.79.00	1.000	.8950
40 %	.4900	1.000	.7450
50 %	.4700	1.000	.7350
60 %	.4200	1.000	.7100
80 %	.2100	.9989	.6044
100 %	.0000	1.000	.5000



(a) 20% of overlap



(b) 80% of overlap

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

V. García, R.A. Mollineda, J.S. Sánchez. On the k-NN performance in a challenging scenario of imbalance and overlapping. Pattern Anal Applic (2008) 11: 269-280

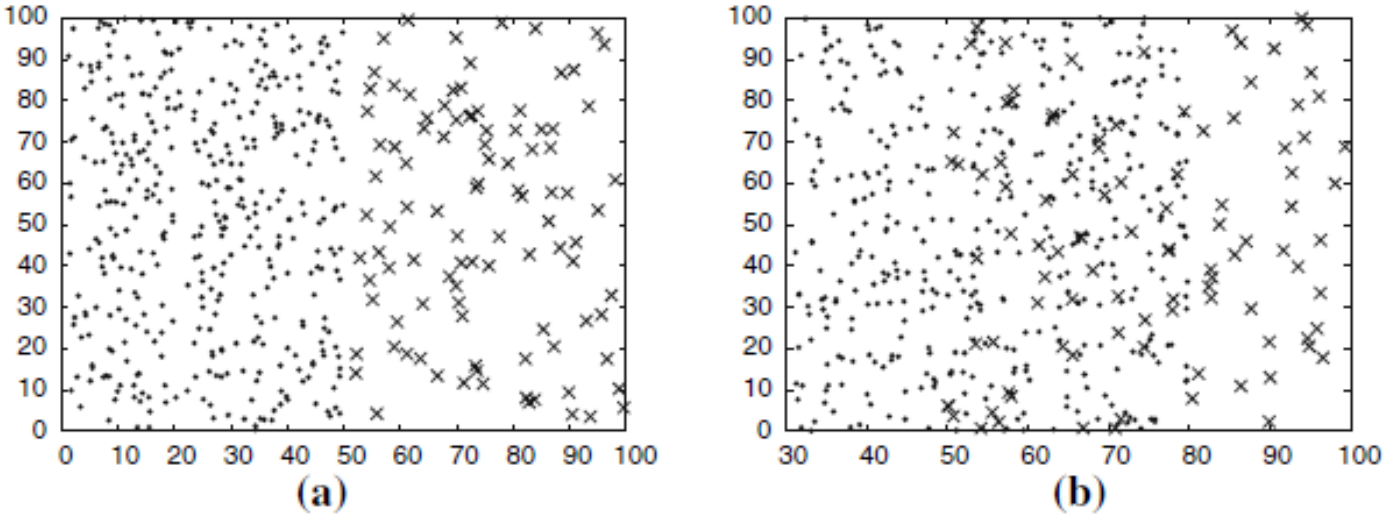


Fig. Two different levels of class overlapping: a 0% and b 60%

Experiment I: The positive examples are defined on the X-axis in the range [50–100], while those belonging to the majority class are generated in [0–50] for 0% of class overlap, [10–60] for 20%, [20–70] for 40%, [30–80] for 60%, [40–90] for 80%, and [50–100] for 100% of overlap.

The overall imbalance ratio matches the imbalance ratio corresponding to the overlap region, what could be accepted as a common case.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

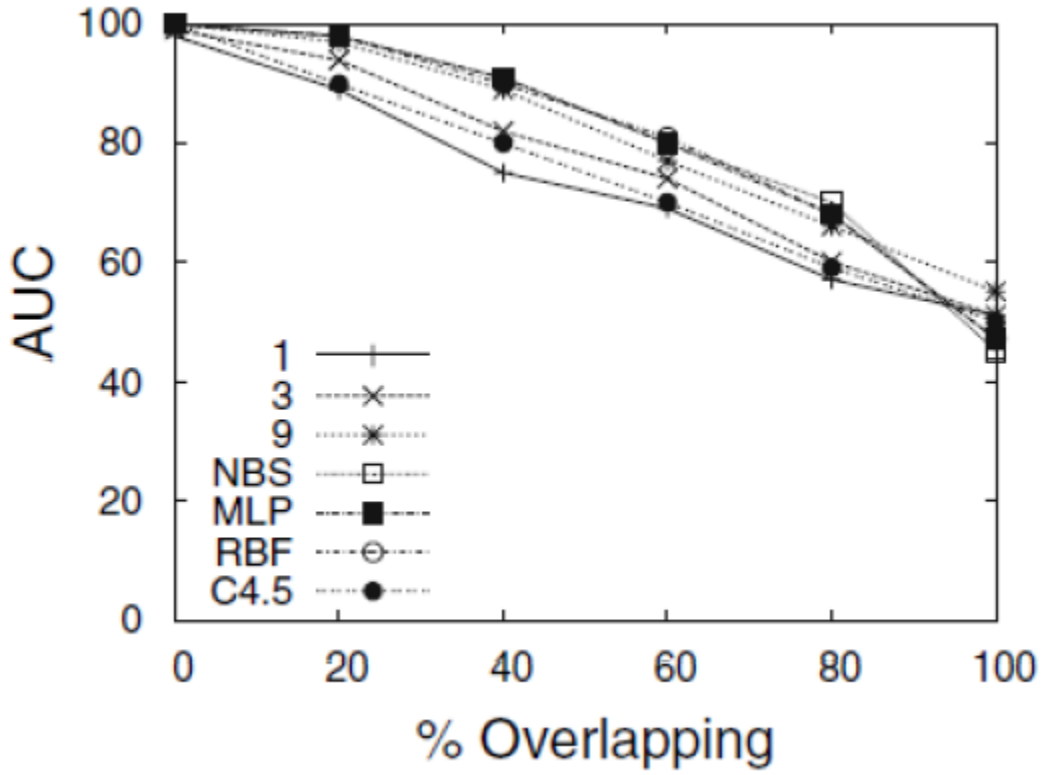


Fig. Performance metrics in k-NN rule and other learning algorithms for experiment I

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

V. García, R.A. Mollineda, J.S. Sánchez. On the k-NN performance in a challenging scenario of imbalance and overlapping. Pattern Anal Applic (2008) 11: 269-280

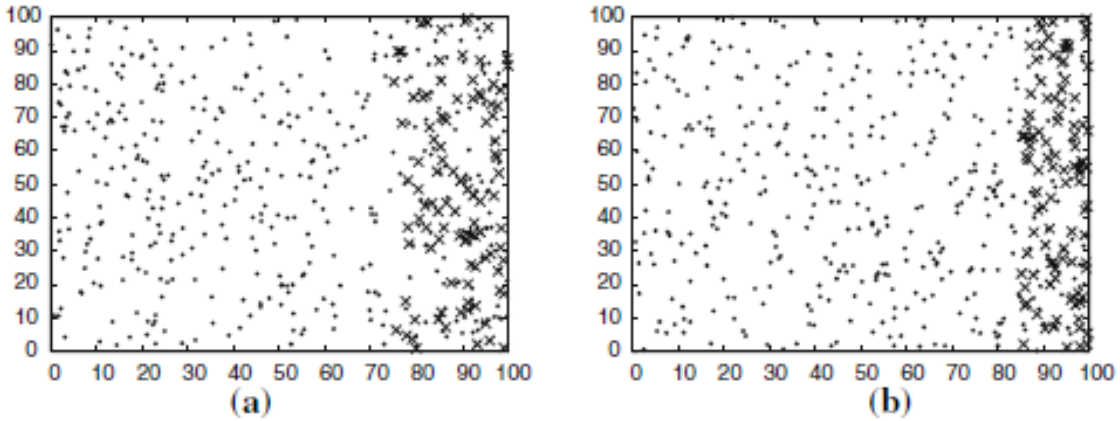


Fig. Two different cases in experiment II: [75-100] and [85-100]. For this latter case, note that in the overlap region, the majority class is under-represented in comparison to the minority class.

Experiment II: The second experiment has been carried out over a collection of five artificial imbalanced data sets in which the overall minority class becomes the majority in the overlap region. To this end, the 400 negative examples have been defined on the X-axis to be in the range [0–100] in all data sets, while the 100 positive cases have been generated in the ranges [75–100], [80–100], [85–100], [90–100], and [95–100]. The number of elements in the overlap region varies from no local imbalance in the first case, where both classes have the same (expected) number of patterns and density, to a critical inverse imbalance in the fifth case, where the 100 minority examples appears as majority in the overlap region along with about 20 expected negative examples.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

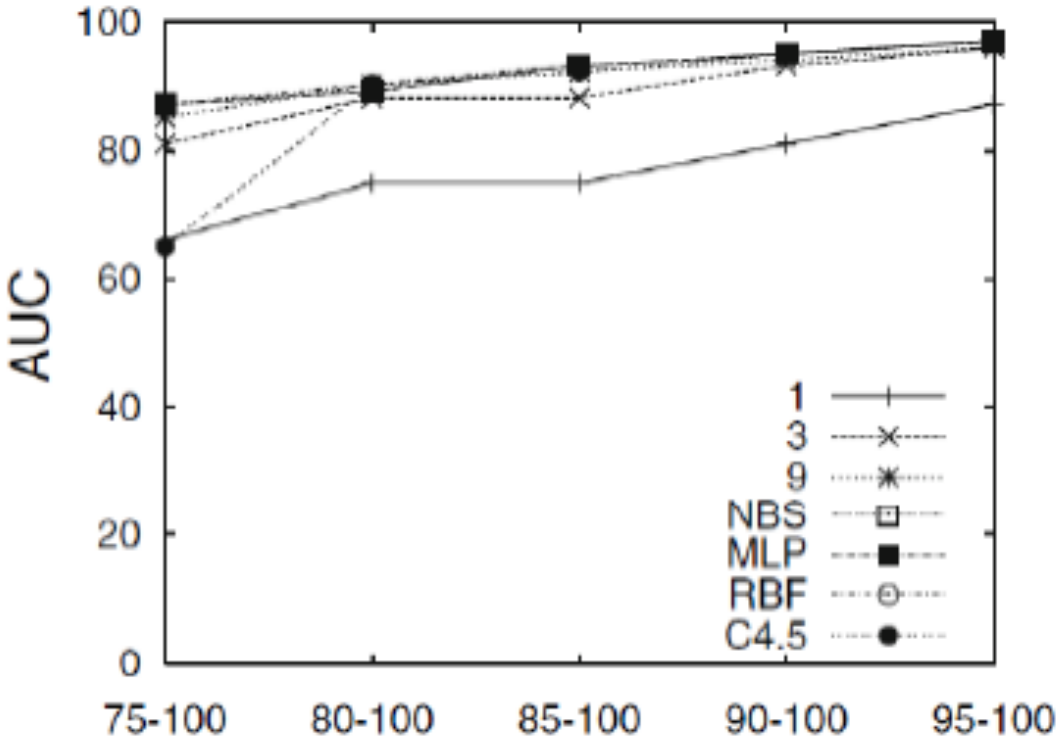
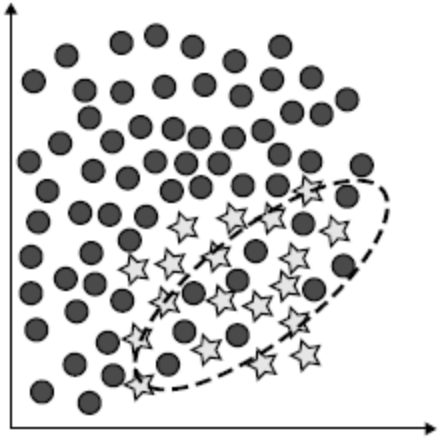


Fig. Performance metrics in k-NN rule and other learning algorithms for experiment II

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping



(a) Class overlapping

Conclusions: Results (in this paper) that the class more represented in ovel regions tends to be better classified by methods based on global learning, while the class less represented in such regions tends to be better classified by local methods.

In this sense, as the value of k of the k-NN rule increases, along with a weakening of its local nature, it was progressively approaching the behaviour of global models.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

Small disjuncts/rare data sets

Density: Lack of data

Bordeline and Noise data

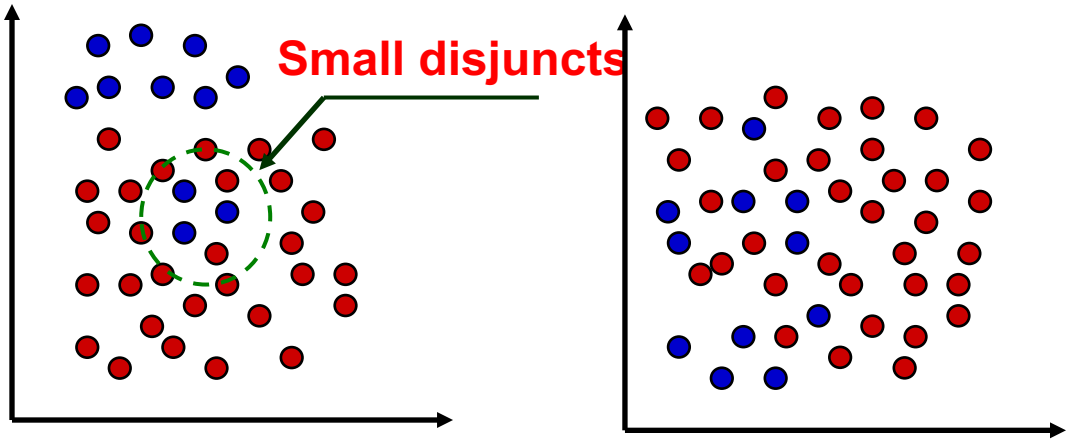
Dataset shift

V. López, A. Fernandez, S. García, V. Palade, F. Herrera, **An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics.** Information Sciences 250 (2013) 113-141.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Class imbalance is not the only responsible of the lack in accuracy of an algorithm.



Class imbalances may yield small disjuncts which, in turn, will cause degradation.

Rare cases or Small disjuncts are those disjuncts in the learned classifier that cover few training examples.

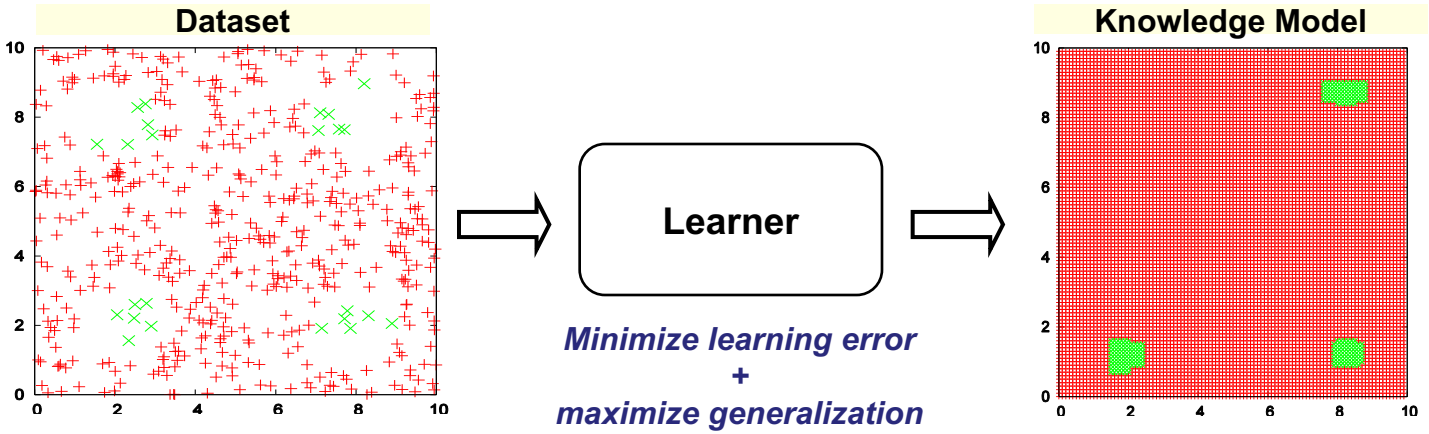
T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49
G.M. Weiss. Mining with Rarity: A Unifying Framework. SIGKDD Explorations 6:1 (2004) 7-19

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Rare or exceptional cases correspond to small numbers of training examples in particular areas of the feature space. When learning a concept, the presence of rare cases in the domain is an important consideration. The reason why rare cases are of interest is that they cause small disjuncts to occur, which are known to be more error prone than large disjuncts.

In the real world domains, rare cases are unknown since high dimensional data cannot be visualized to reveal areas of low coverage.



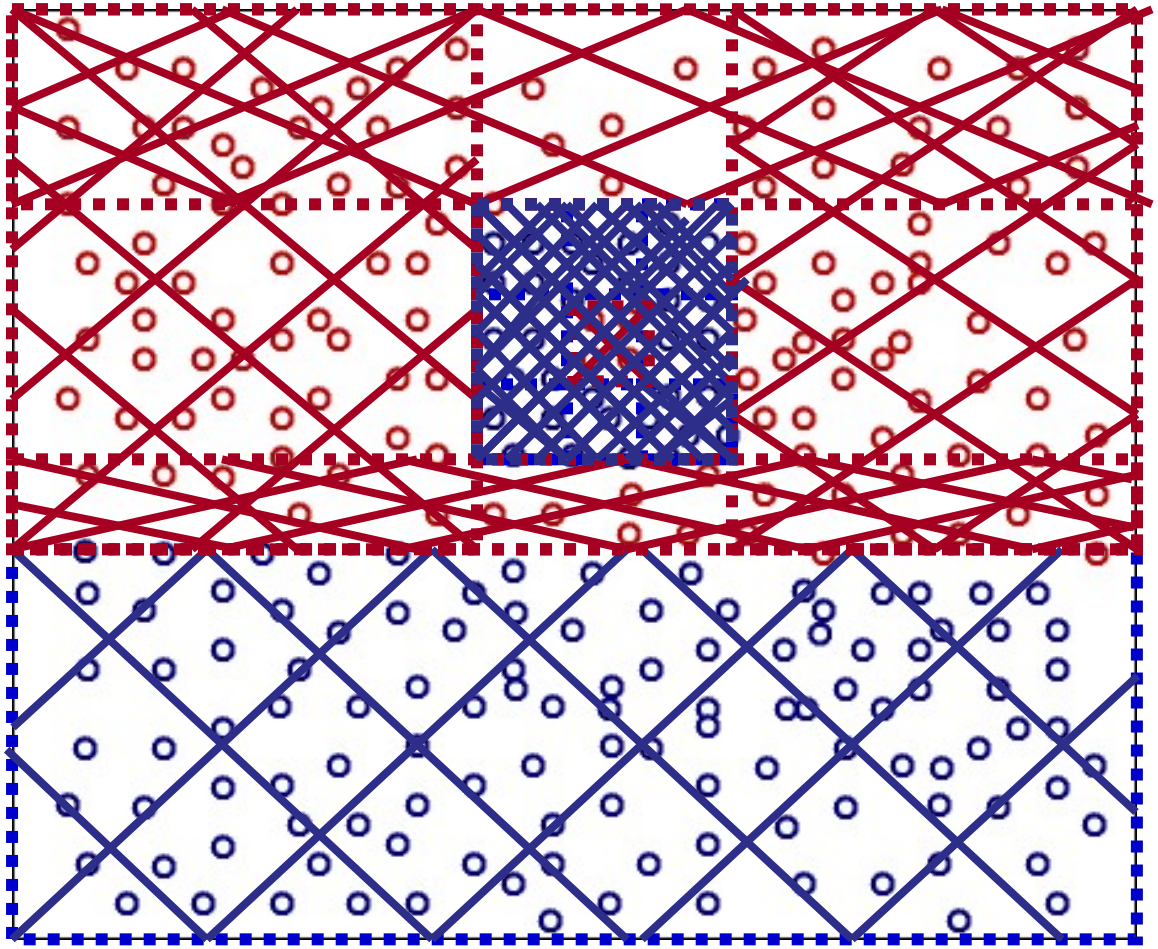
T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Rare or excepcional cases

Rare cases or Small disjunct: Focusing the problem



Small Disjunct **or** ***Starved niche***

Again ***more small disjuncts***

Overgeneral Classifier

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Rare or excepcional cases

Rarity: Rare Cases versus Rare Classes

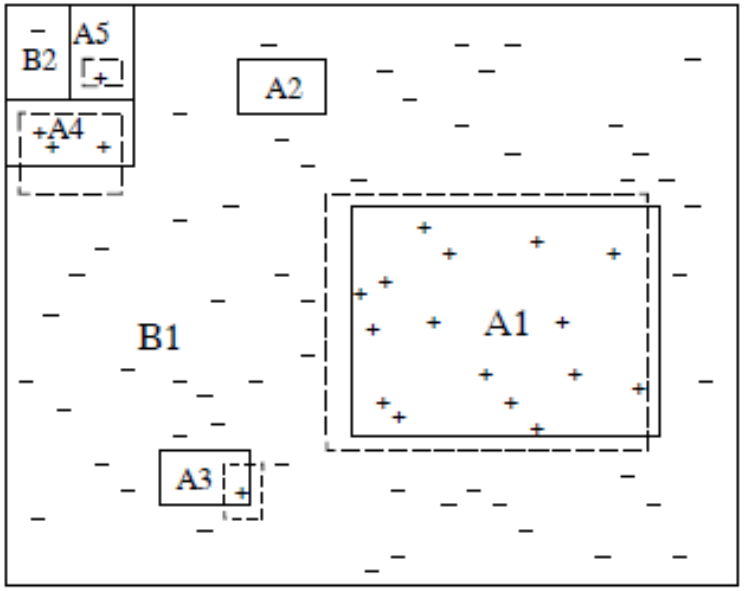


Figure 1: Graphical representation of a rare class and rare case

Class A is the rare (minority class and B is the common (majority class).

Subconcepts A2-A5 correspond to rare cases, whereas A1 corresponds to a fairly common case, covering a substantial portion of the instance space.

Subconcept B2 corresponds to a rare case, demonstrating that common classes may contain rare cases.

G.M. Weiss. Mining with Rarity: A Unifying Framework. SIGKDD Explorations 6:1 (2004) 7-19

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Small disjuncts/Rare or excepcional cases

In the real-word domains, rare cases are not easily identified. An approximation is to use a clustering algorithm on each class.

Jo and Japkowicz, 2004: Cluster-based oversampling: A method for inflating small disjuncts.

> Once the training examples of each class have been clustered, oversampling starts. In the majority class, all the clusters, except for the largest one, are randomly oversampled so as to get the same number of training examples as the largest cluster. Let *maxclasssize* be the overall size of the large class. In the minority class, each cluster is randomly oversampled until each cluster contains *maxclasssize/Nsmallclass* where *Nsmallclass* represents the number of subclusters in the small class.

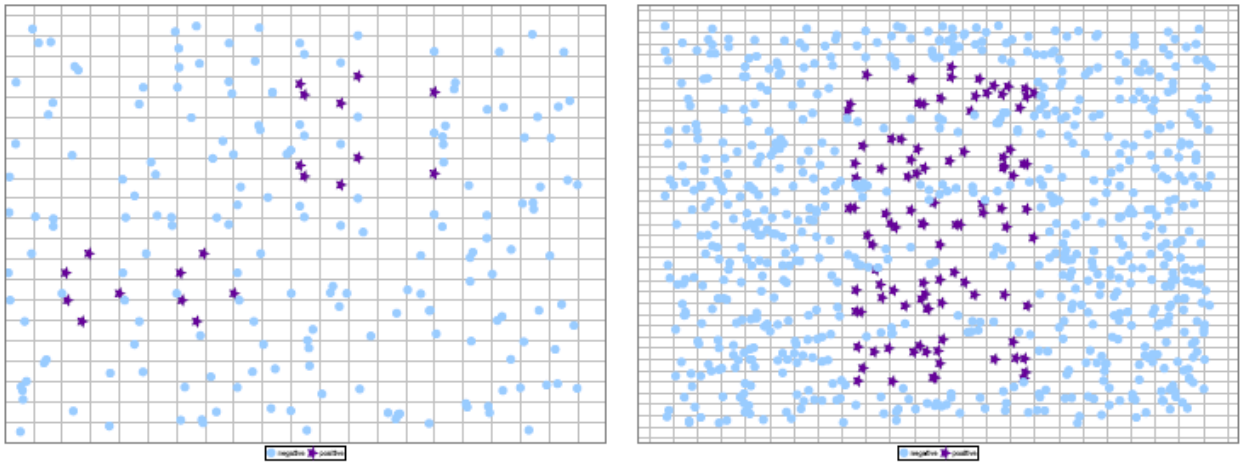
CBO method: Cluster-based resampling identifies rare cases and re-samples them individually, so as to avoid the creation of small disjuncts in the learned hypothesis.

T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Small disjuncts/Rare or excepcional cases



(a) Artificial dataset: small disjuncts for the minority class

(b) Subclus dataset: small disjuncts for both classes

Fig. 5 Example of small disjuncts on imbalanced data

Table 12 Performance obtained by C4.5 in datasets suffering from small disjuncts

Dataset	Original Data			Preprocessed Data with CBO		
	TP_{rate}	TN_{rate}	AUC	TP_{rate}	TN_{rate}	AUC
Artificial dataset	.0000	1.000	.5000	1.000	1.000	1.000
Subclus dataset	1.000	.9029	.9514	1.000	1.000	1.000

T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Rare or excepcional cases

Small disjuncts play a role in the performance loss of class imbalanced domains.

Jo and Japkowicz results show that it is the small disjuncts problem more than the class imbalance problem that is responsible for the this decrease in accuracy.

The performance of classifiers, though hindered by class imbalanced, is repaired as the training set size increases.

An open question: **Whether it is more effective to use solutions that address both the class imbalance and the small disjunct problem simultaneously than it is to use solutions that address the class imbalance problem or the small disjunct problem, alone.**

T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

Small disjuncts/rare data sets

Density: Lack of data

Bordeline and Noise data

Dataset shift

V. López, A. Fernandez, S. García, V. Palade, F. Herrera, **An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics.** Information Sciences 250 (2013) 113-141.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Density: Lack of data

Table 5. The Distribution of Training Examples in Pima Indian Diabetes

		Positive ('1')	Negative ('0')
1:9	40	4	36
	100	10	90
	200	20	180
1:3	40	10	30
	100	25	75
	200	50	150
1:1	40	20	20
	100	50	50
	200	100	100

Different levesl of imbalance and density

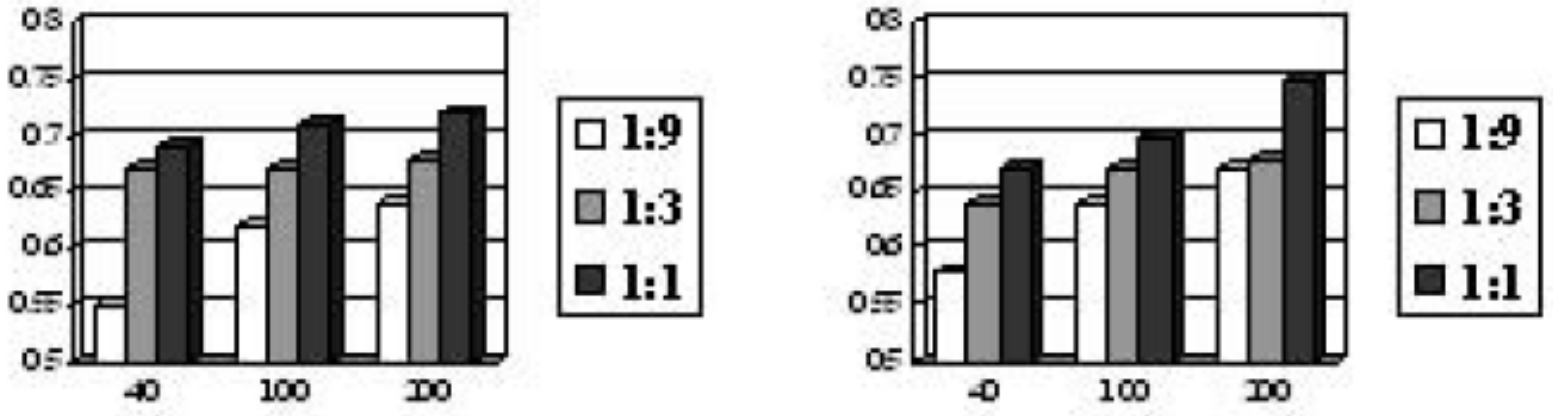
T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Density: Lack of data

Left-C4.5, right-Backpropagation: These resultas show that the performance of classifiers, though hindered by class imbalances, is repaired as the training set size incresses. This sugests that small disjuncts play a role in the performance loss of class imbalanced domains.



T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Density: Lack of data

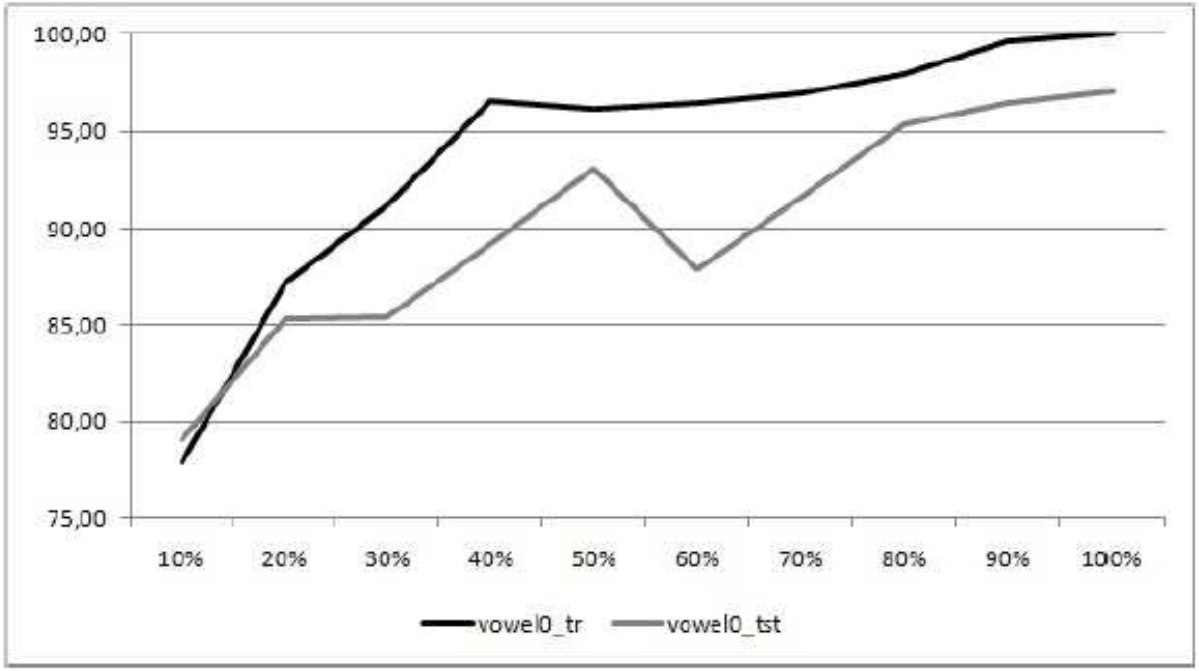


Fig. 8 AUC performance for the C4.5 classifier regarding the proportion of examples in the training set for the vowel0 problem

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

Small disjuncts/rare data sets

Density: Lack of data

Bordeline and Noise data

Dataset shift

V. López, A. Fernandez, S. García, V. Palade, F. Herrera, **An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics.** Information Sciences 250 (2013) 113-141.

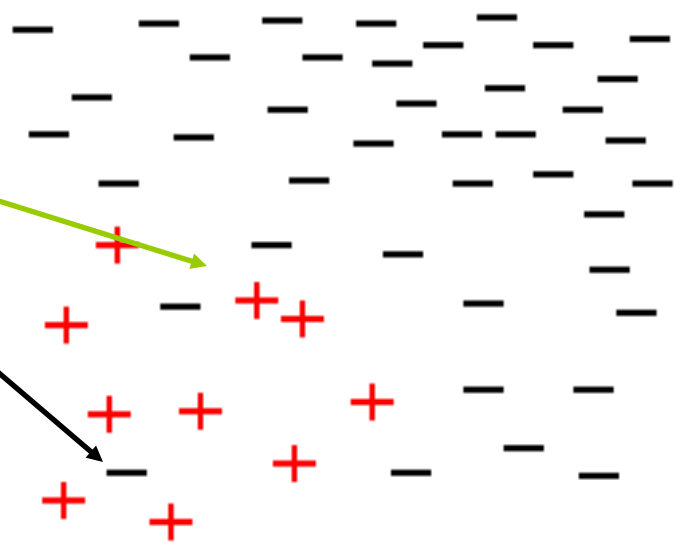
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Kind of examples: **The need of resampling or to manage the overlapping with other strategies**

- ❑ **Noise examples**
- ❑ **Borderline examples**

  Borderline examples are unsafe since a small amount of noise can make them fall on the wrong side of the decision border.
- ❑ **Redundant examples**



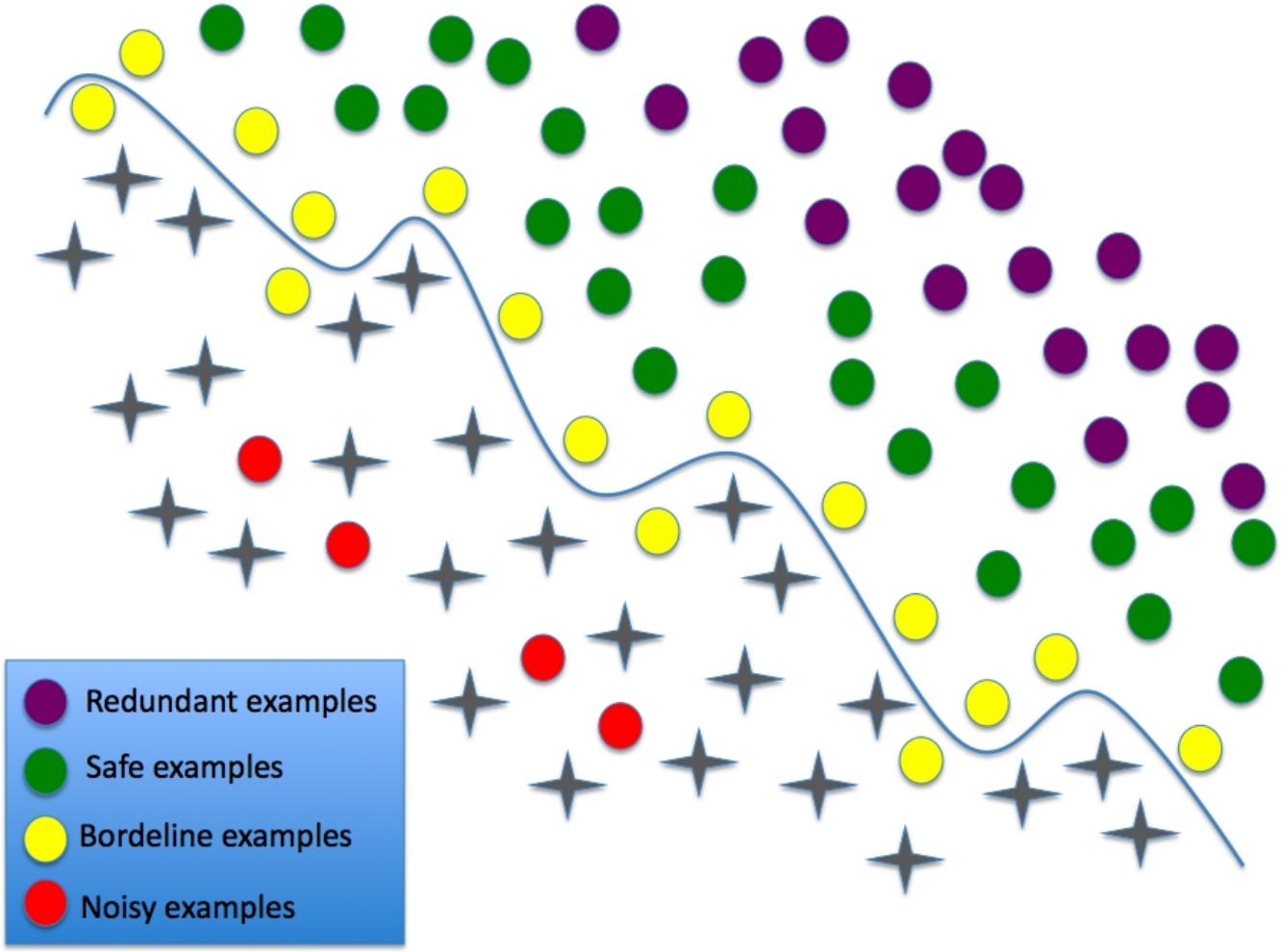
- ❑ **Safe examples**

An approach: Detect and remove such majority noisy and borderline examples in filtering before inducing the classifier.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data



Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

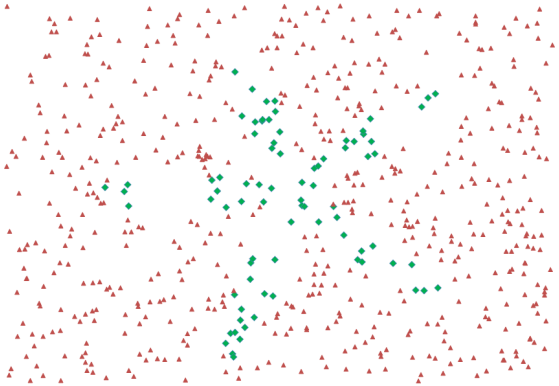
Bordeline and Noise data

3 kind of artificial problems:

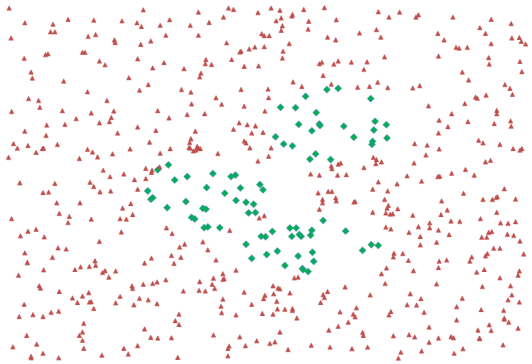
Subclus: examples from the minority class are located inside rectangles following related works on small disjuncts.

Clover: It represents a more difficult, non-linear setting, where the minority class resembles a flower with elliptic petals.

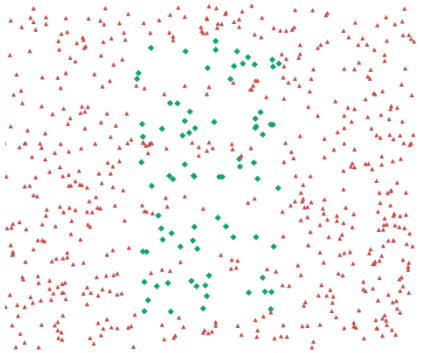
Paw: The minority class is decomposed into 3 elliptic sub-regions of varying cardinalities, where two subregions are located close to each other, and the remaining smaller sub-region is separated.



Clover data



Paw data

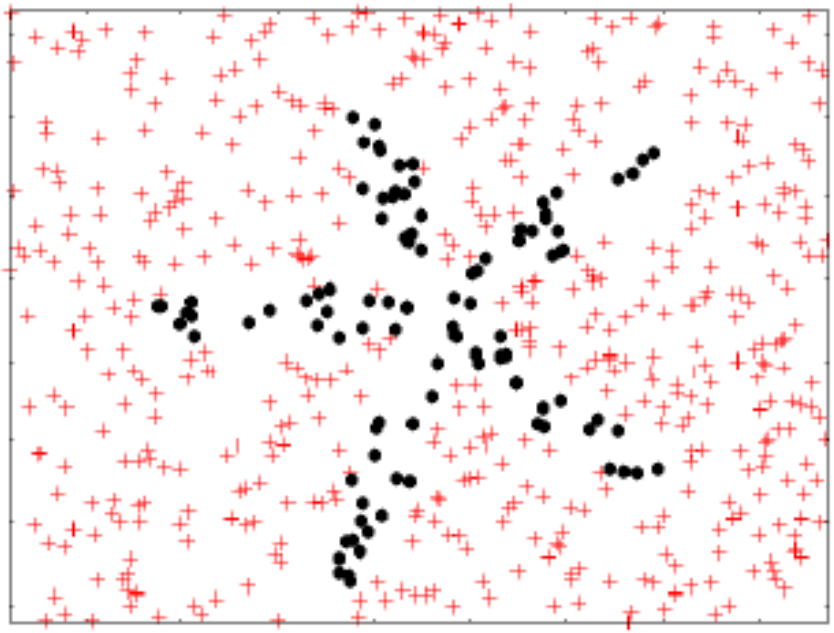


Subclus data

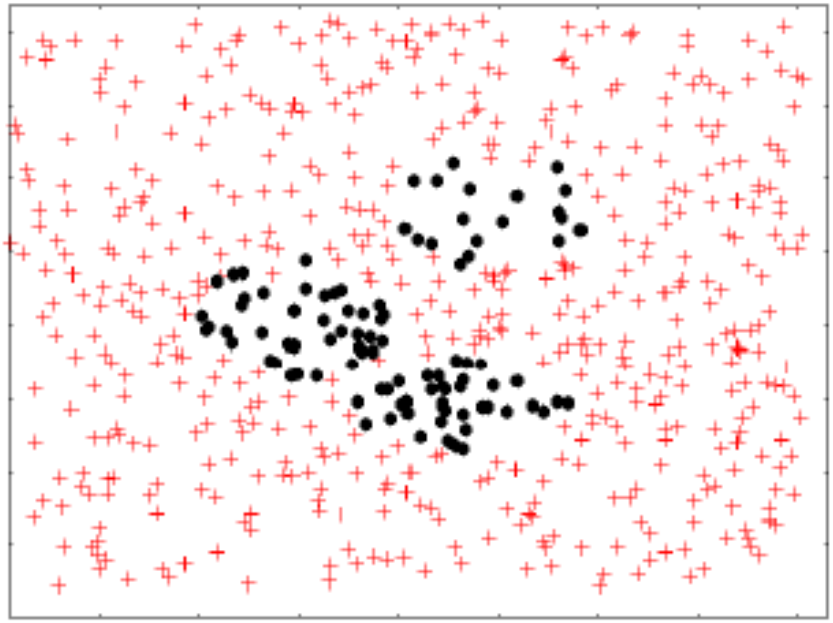
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data



Clover data

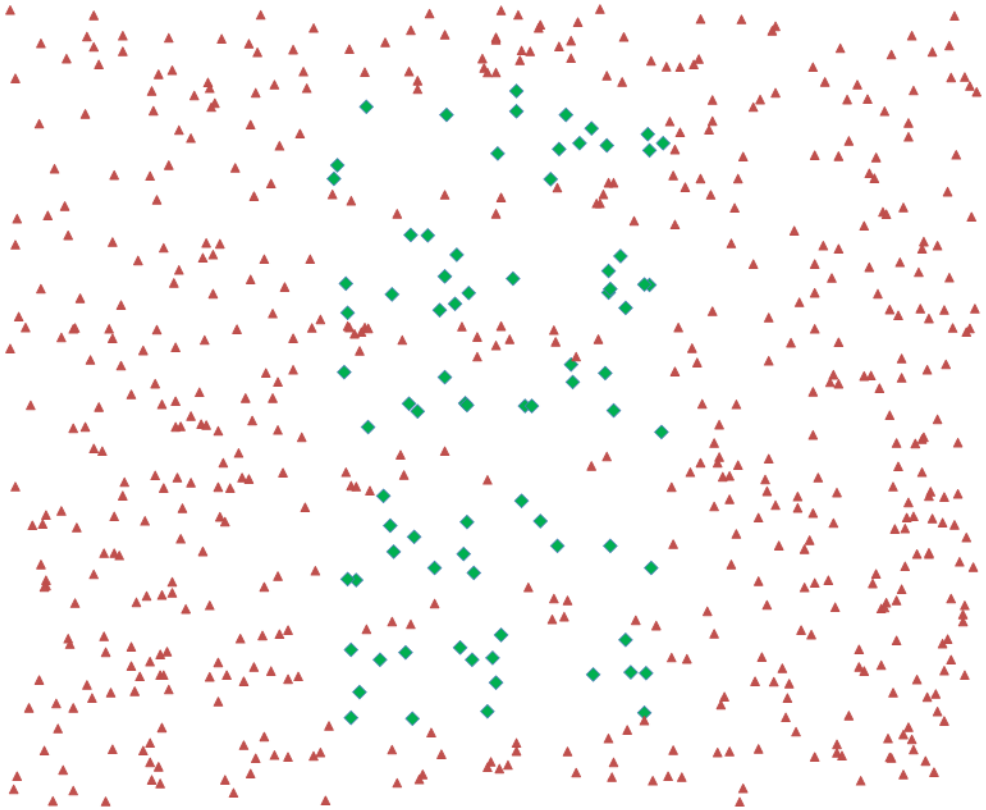


Paw data

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data

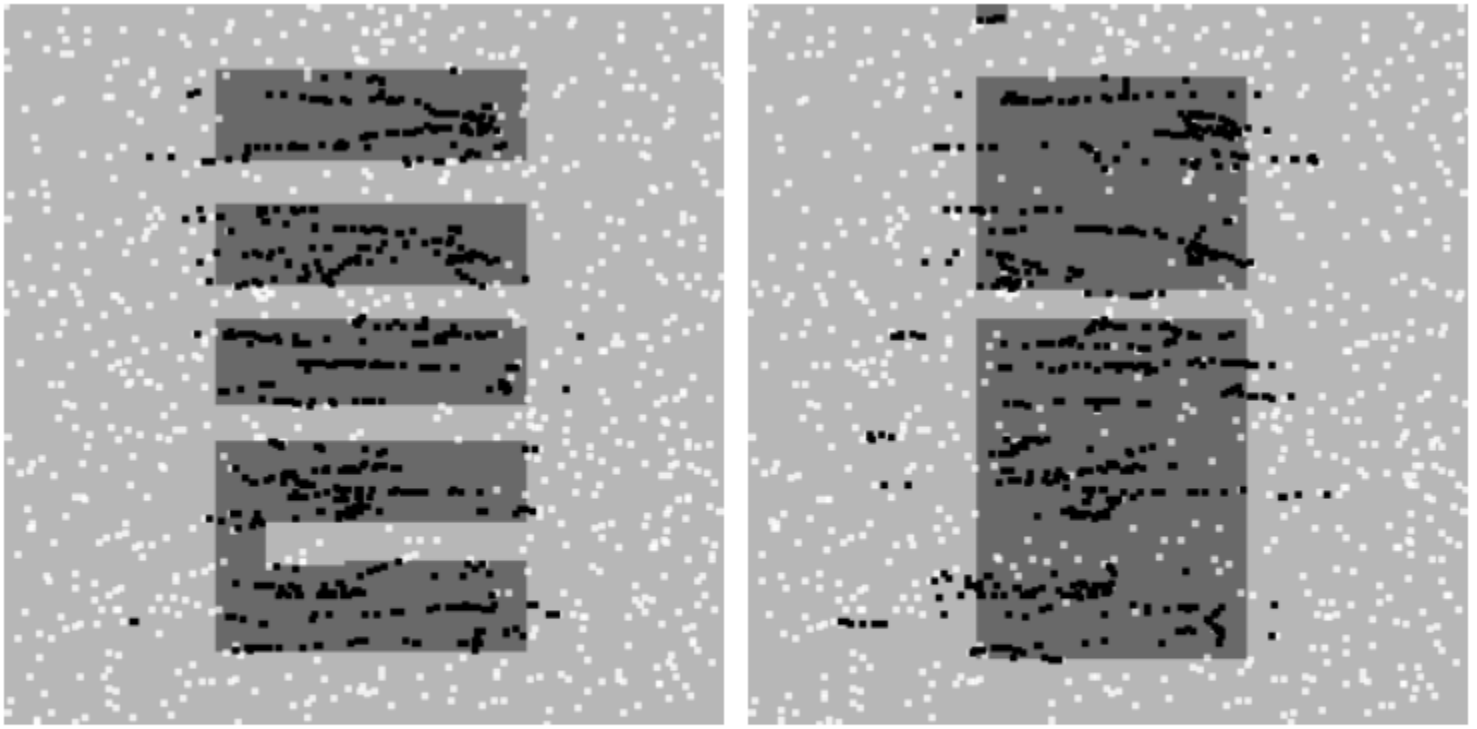


Subclus data

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data



(a) Original problem and decision functions (b) Noisy instances and new undesirable decision functions

Fig. 10 Example of the effect of noise in imbalanced datasets for SMOTE+C4.5 in the Subclus dataset

Subclus data

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data

SPIDER 2: Spider family (Selective Preprocessing of Imbalanced Data) rely on the local characteristics of examples discovered by analyzing their k-nearest neighbors.

J. Stefanowski, S. Wilk. Selective pre-processing of imbalanced data for improving classification performance. 10th International Conference in Data Warehousing and Knowledge Discovery (DaWaK2008). LNCS 5182, Springer 2008, Turin (Italy, 2008) 283-292.
K.Napierala, J. Stefanowski, and S. Wilk. **Learning from Imbalanced Data in Presence of Noisy and Borderline Examples**. **7th International Conference on Rough Sets and Current Trends in Computing , 7th International Conference on Rough Sets and Current Trends in Computing,** RSCTC 2010, LNAI 6086, pp. 158–167, 2010.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data

Data set	C4.5				
	Base	RO	CO	NCR	SP2
subclus-0	0.9540	0.9500	0.9500	0.9460	0.9640
subclus-30	0.4500	0.6840	0.6720	0.7160	0.7720
subclus-50	0.1740	0.6160	0.6000	0.7020	0.7700
subclus-70	0.0000	0.6380	0.7000	0.5700	0.8300
clover-0	0.4280	0.8340	0.8700	0.4300	0.4860
clover-30	0.1260	0.7180	0.7060	0.5820	0.7260
clover-50	0.0540	0.6560	0.6960	0.4460	0.7700
clover-70	0.0080	0.6340	0.6320	0.5460	0.8140
paw-0	0.5200	0.9140	0.9000	0.4900	0.5960
paw-30	0.2640	0.7920	0.7960	0.8540	0.8680
paw-50	0.1840	0.7480	0.7200	0.8040	0.8320
paw-70	0.0060	0.7120	0.6800	0.7460	0.8780

Noise data

Table 14 Performance obtained by C4.5 in the Subclus dataset with and without noisy instances

Dataset	Original Data			20% of Gaussian Noise		
	TP_{rate}	TN_{rate}	AUC	TP_{rate}	TN_{rate}	AUC
None	1.000	.9029	.9514	.0000	1.000	.5000
RandomUnderSampling	1.000	.7800	.8900	.9700	.7400	.8550
SMOTE	.9614	.9529	.9571	.8914	.8800	.8857
SMOTE+ENN	.9676	.9623	.9649	.9625	.9573	.9599
SPIDER2	1.000	1.000	1.000	.9480	.9033	.9256

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data

Small disjunct and Noise data

Bordeline and Noise data

Bordeline and Noise data

Data set	C4.5 Base	RO	CO	NCR	SP2
subclus-0	0.9540	0.9500	0.9500	0.9460	0.9640
subclus-30	0.4500	0.6840	0.6720	0.7160	0.7720
subclus-50	0.1740	0.6160	0.6000	0.7020	0.7700
subclus-70	0.0000	0.6380	0.7000	0.5700	0.8300
clover-0	0.4280	0.8340	0.8700	0.4300	0.4860
clover-30	0.1260	0.7180	0.7060	0.5820	0.7260
clover-50	0.0540	0.6560	0.6960	0.4460	0.7700
clover-70	0.0080	0.6340	0.6320	0.5460	0.8140
paw-0	0.5200	0.9140	0.9000	0.4900	0.5960
paw-30	0.2640	0.7920	0.7960	0.8540	0.8680
paw-50	0.1840	0.7480	0.7200	0.8040	0.8320
paw-70	0.0060	0.7120	0.6800	0.7460	0.8780

Table 14 Performance obtained by C4.5 in the Subclus dataset with and without noisy instances

Dataset	Original Data			20% of Gaussian Noise		
	TP_{rate}	TN_{rate}	AUC	TP_{rate}	TN_{rate}	AUC
None	1.000	.9029	.9514	.0000	1.000	.5000
RandomUnderSampling	1.000	.7800	.8900	.9700	.7400	.8550
SMOTE	.9614	.9529	.9571	.8914	.8800	.8857
SMOTE+ENN	.9676	.9623	.9649	.9625	.9573	.9599
SPIDER2	1.000	1.000	1.000	.9480	.9033	.9256

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data

- SPIDER 2: allows to get good results in comparison with classical ones.
- It has interest to analyze the use of noise filtering algorithms for these problems: IPF filtering algorithm shows good results.

José A. Sáez, J. Luengo, Jerzy Stefanowski, F. Herrera, **SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering**. *Information Sciences 291 (2015) 184-203, doi: 10.1016/j.ins.2014.08.051*.

- Specific methods for managing the noise and bordeline problems are necessary.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

Small disjuncts/rare data sets

Density: Lack of data

Bordeline and Noise data

Dataset shift

Three connected problems

V. López, A. Fernandez, S. García, V. Palade, F. Herrera, **An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics.** Information Sciences 250 (2013) 113-141.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Small disjuncts and density

Rare cases may be due to a lack of data. Relative lack of data. relative raritv.

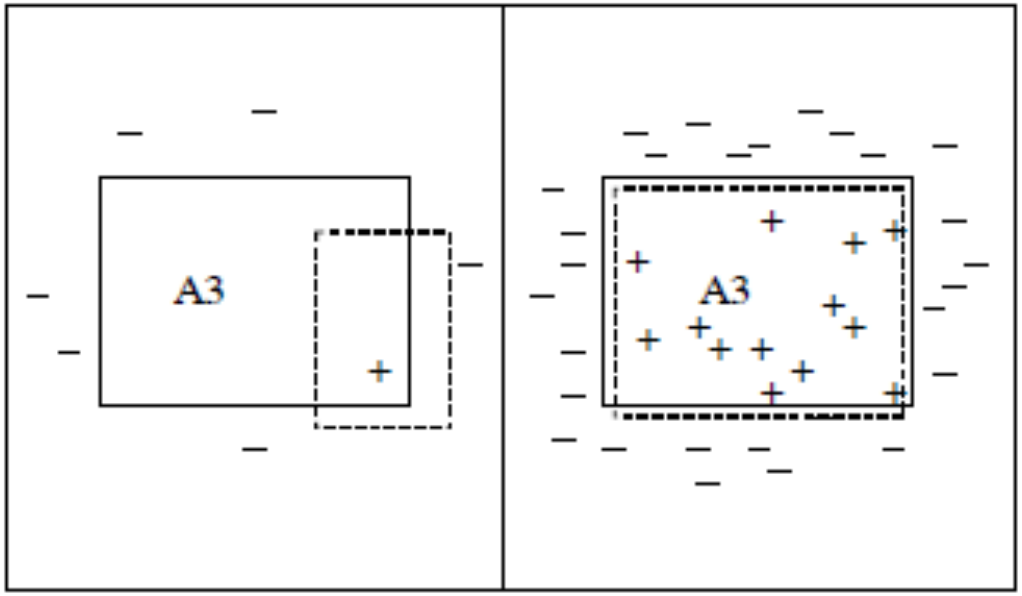


Figure 2: The impact of an "absolute" lack of data

G.M. Weiss. Mining with Rarity: A Unifying Framework. SIGKDD Explorations 6:1 (2004) 7-19

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Small disjuncts and Noise data

Noise data will affect the way any data mining system behaves. Noise has a greater impact on rare cases than on common cases.

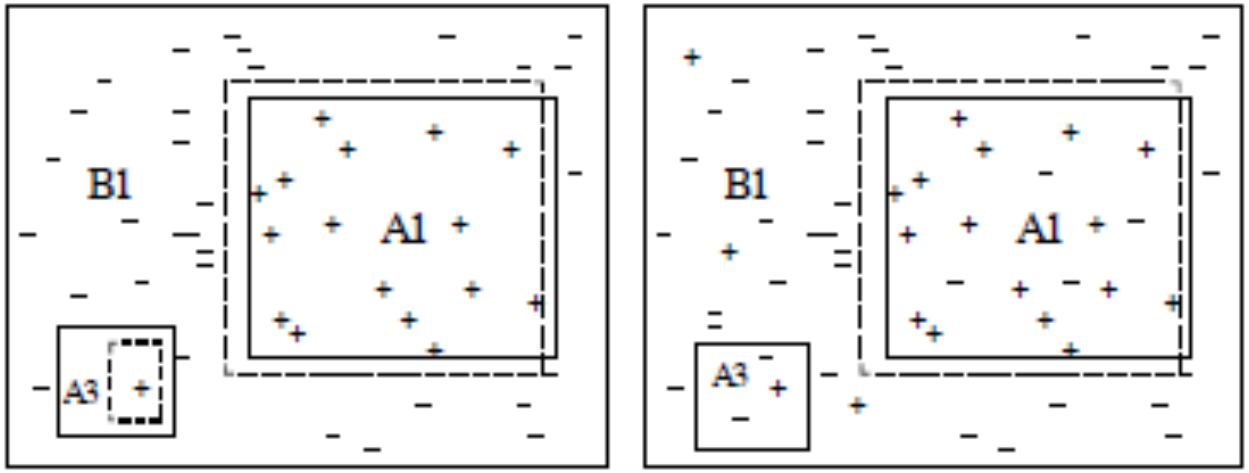


Figure 3: The effect of noise on rare cases

G.M. Weiss. Mining with Rarity: A Unifying Framework. SIGKDD Explorations 6:1 (2004) 7-19

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

Small disjuncts/rare data sets

Density: Lack of data

Bordeline and Noise data

Dataset shift

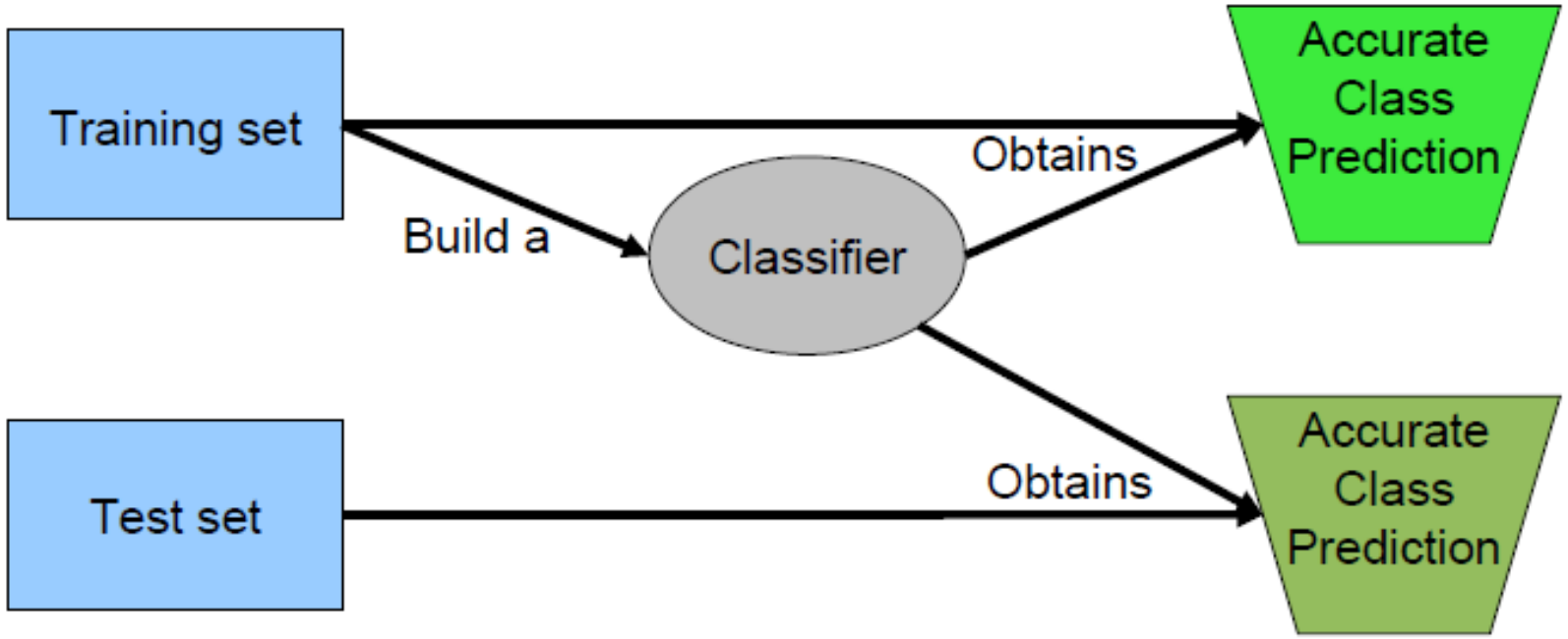
V. López, A. Fernandez, S. García, V. Palade, F. Herrera, **An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics.** Information Sciences 250 (2013) 113-141.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

- Basic assumption in classification:

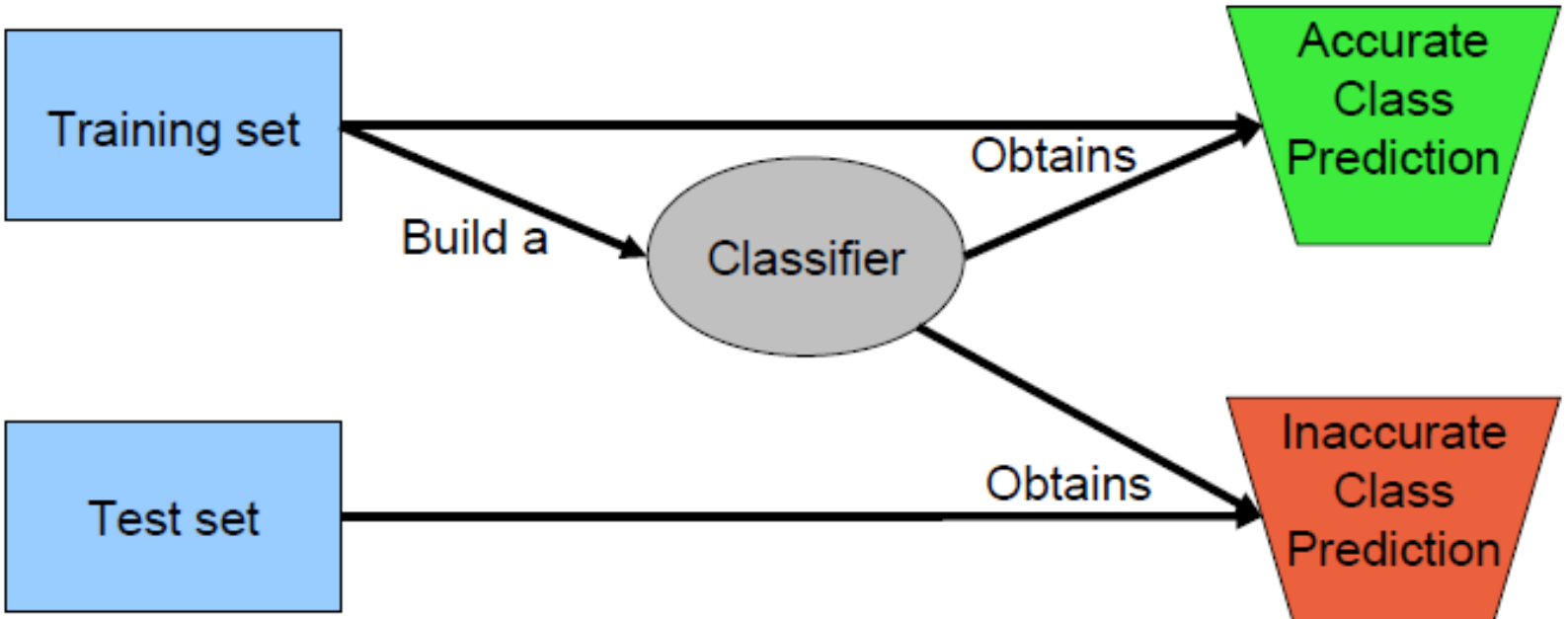


Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

- But sometimes....



- **The classifier has an overfitting problem.**
- **Is there a change in data distribution between training and test sets (Data fracture)?.**

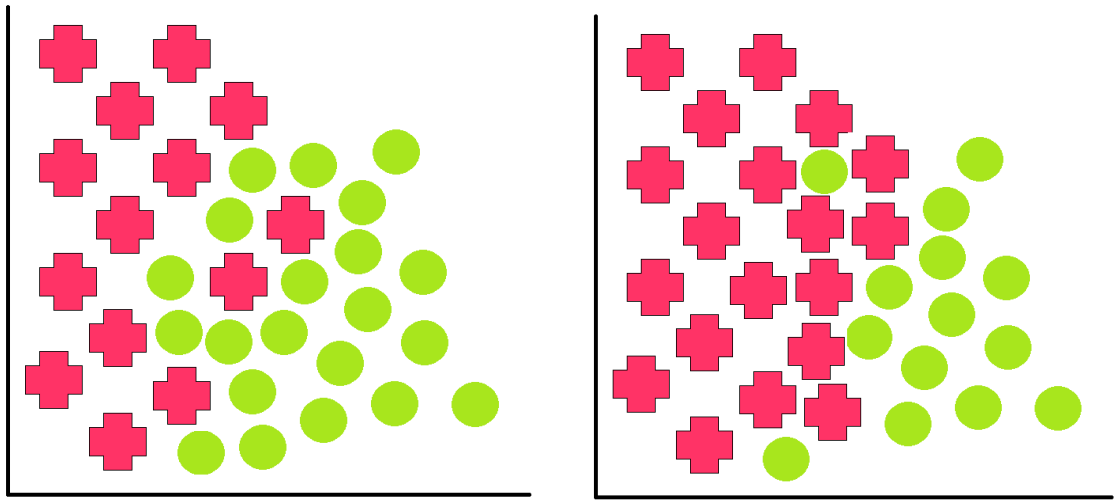
The Problem of Dataset Shift

- **The classifier has an overfitting problem.**
  - Change the parameters of the algorithm.
  - Use a more general learning method.

- **There is a change in data distribution between training and test sets (Dataset shift).**
  - Train a new classifier for the test set.
  - Adapt the classifier.
  - Modify the data in the test set …

The Problem of Dataset Shift

The problem of data-set shift is defined as the case where training and test data follow different distributions.



J. G. Moreno-Torres, T. R. Raeder, R. Aláiz-Rodríguez, N. V. Chawla, F. Herrera, A unifying view on dataset shift in classification. *Pattern Recognition 45:1 (2012) 521-530, doi:10.1016/j.patcog.2011.06.019*.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

This is a common problem that can affect all kind of classification problems, and it often appears due to sample selection bias issues.
However, **the data-set shift issue is specially relevant when dealing with imbalanced classification**, because in highly imbalanced domains, the minority class is particularly sensitive to singular classification errors, due to the typically low number of examples it presents.
In the most extreme cases, a single misclassified example of the minority class can create a significant drop in performance.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

Since dataset shift is a **highly relevant issue in imbalanced classification,** it is easy to see why it would be an interesting perspective to focus on future research regarding the topic.

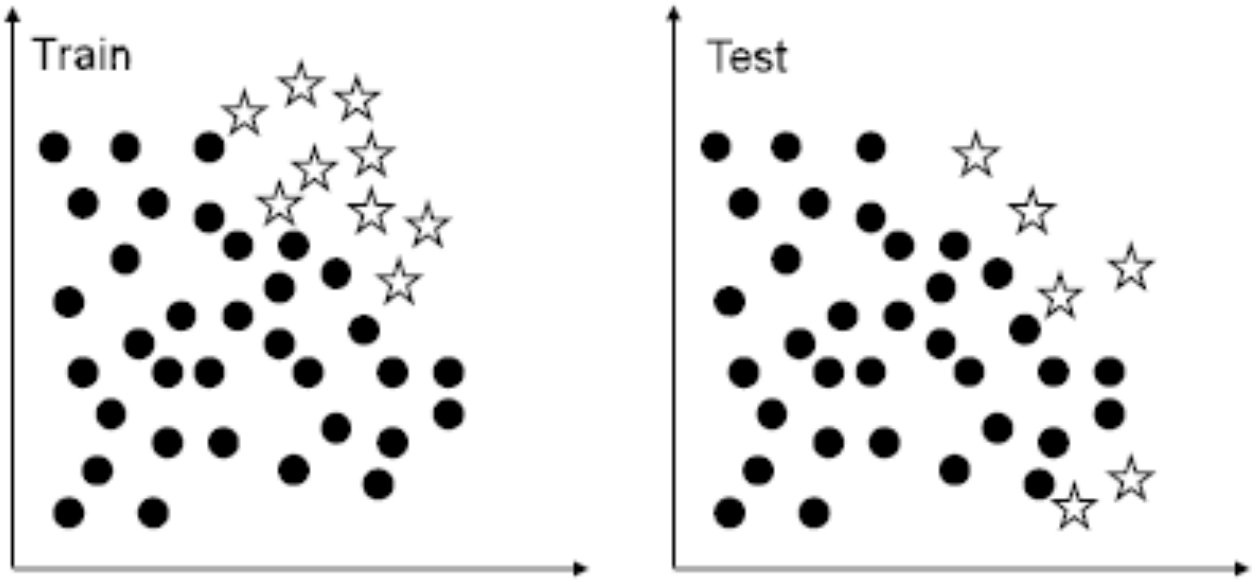


Figure 18: Example of the impact of data-set shift in imbalanced domains.

Causes of Dataset Shift

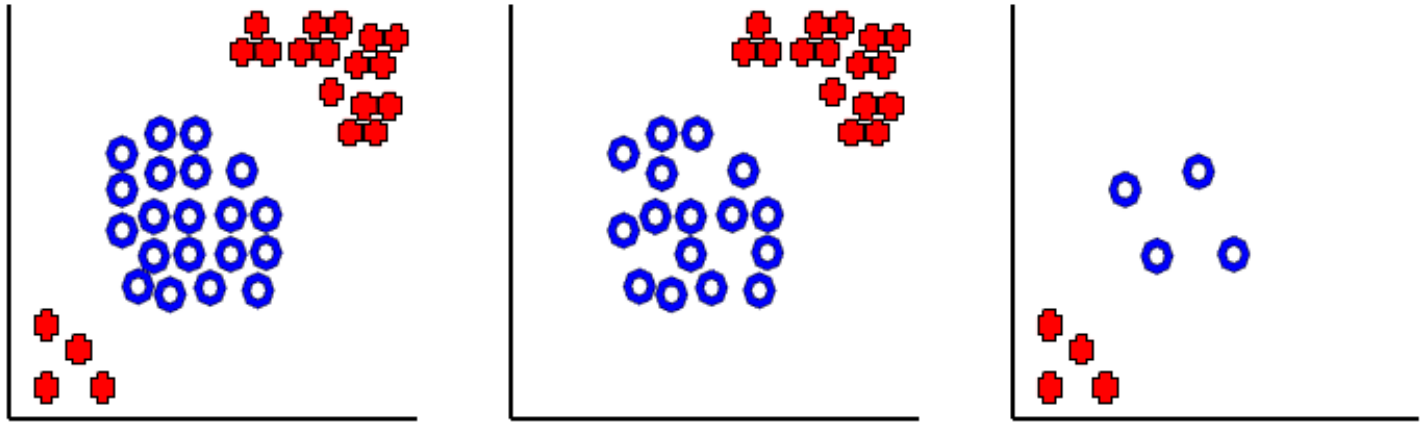
We comment on some of the most common causes of Dataset Shift:

Sample selection bias and non-stationary environments.

These concepts have created confusion at times, so it is important to remark **that these terms are factors that can lead to the appearance of some of the shifts explained, but they do not constitute Dataset Shift themselves.**

Causes of Dataset Shift

Sample selection bias:

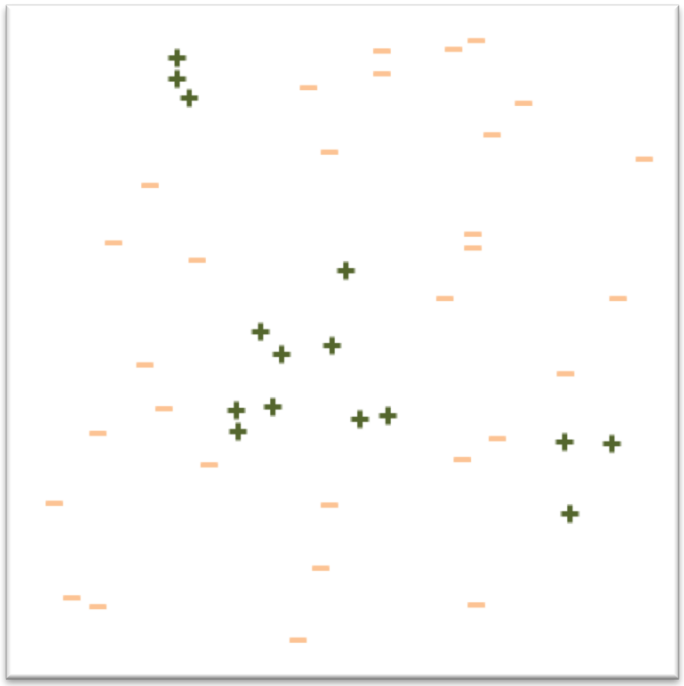


(a) Full dataset (b) Training set (c) Test set

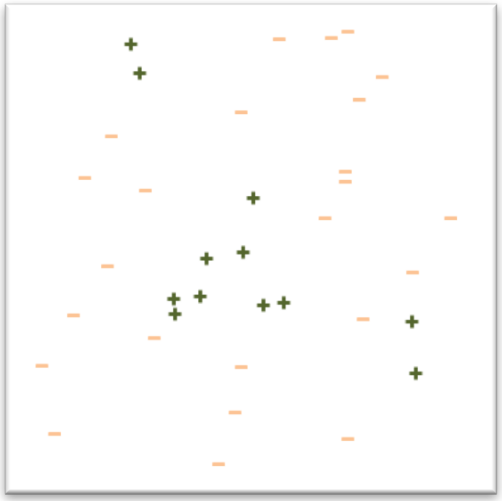
Fig. 1: Extreme example of partition-based covariate shift. Note how the examples on the bottom left of the "cross" class will be wrongly classified due to covariate shift.

Causes of Dataset Shift

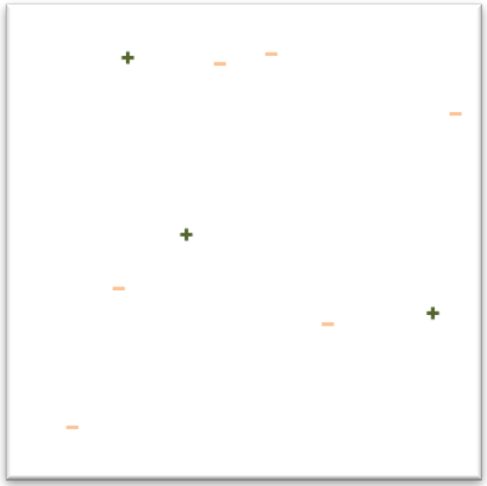
- **Training and test following the same data distribution**



Original Data



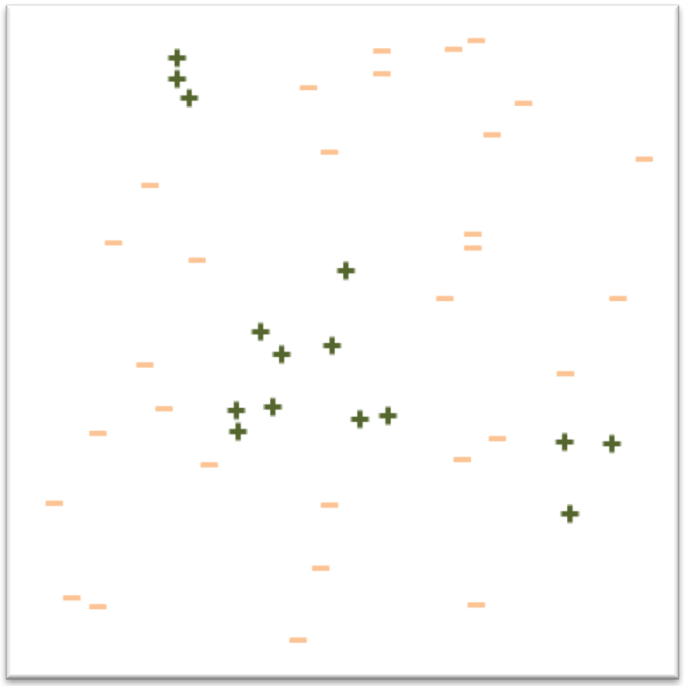
Training Data



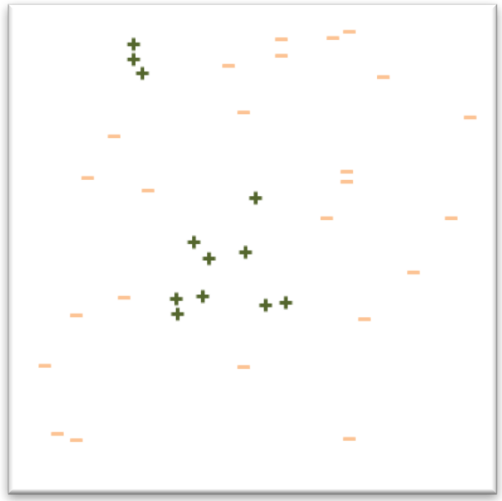
Test Data

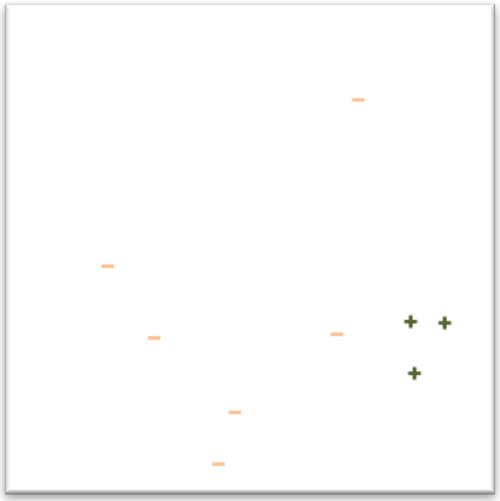
Causes of Dataset Shift

- **DATASET SHIT: Training and test following <u>different</u> data distribution**



Original Data





Training Data Test Data

Causes of Dataset Shift

Sample bias selection: Influence of partitioning on classifiers' performance

	Iteration 216		Iteration 459	
	C45	HDDT	C45	HDDT
breast-w	**0.9784**	0.9753	0.9768	**0.9820**
bupa	**0.6936**	0.6913	0.6521	**0.6531**
credit-a	**0.8996**	0.8967	**0.9044**	0.8967
crx	**0.8993**	0.8877	**0.9021**	0.8898
heart-c	**0.8431**	0.8181	0.8161	**0.8333**
heart-h	**0.8756**	0.8290	0.8376	**0.8404**
horse-colic	0.8646	**0.8848**	0.8742	**0.8928**
ion	**0.9353**	0.9301	0.9247	**0.9371**
krkp	0.9992	**0.9993**	0.9988	**0.9991**
pima	**0.7781**	0.7717	0.7661	**0.7696**
promoters	**0.8654**	0.8514	0.8676	**0.8774**
ringnorm	**0.8699**	0.8533	0.8669	**0.8727**
sonar	**0.8053**	0.7929	0.8076	**0.8127**
threenorm	**0.7964**	0.7575	**0.7419**	0.7311
tic-tac-toe	**0.9354**	0.9254	**0.9342**	0.9273
twonorm	**0.8051**	0.8023	0.7722	**0.7962**
vote	**0.9843**	0.9824	0.9828	**0.9835**
vote1	**0.9451**	0.9343	**0.9497**	0.9426
avg. rank	**1.11**	1.89	1.72	**1.28**
$\alpha = 0.10$	✓			✓
$\alpha = 0.05$	✓			✓

- **Classifier performance results over two separate iterations of random 10-fold cross-validation.**
- **A consistent random number seed was used across al datasets within an iteration.**

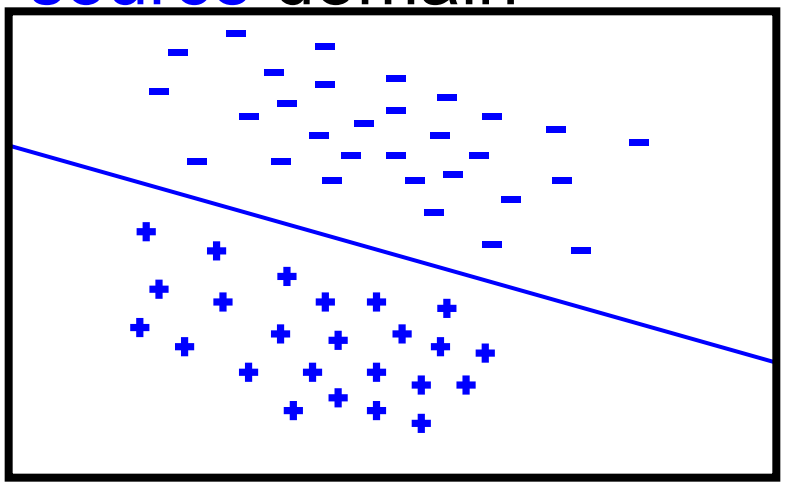
T. Raeder, T. R. Hoens, and N. V. Chawla, "Consequences of variability in classifier performance estimates," Proceedings of the 2010 IEEE International Conference on Data Mining, 2010, pp. 421–430.

Wilcoxon test: Clear differences for both algorithms

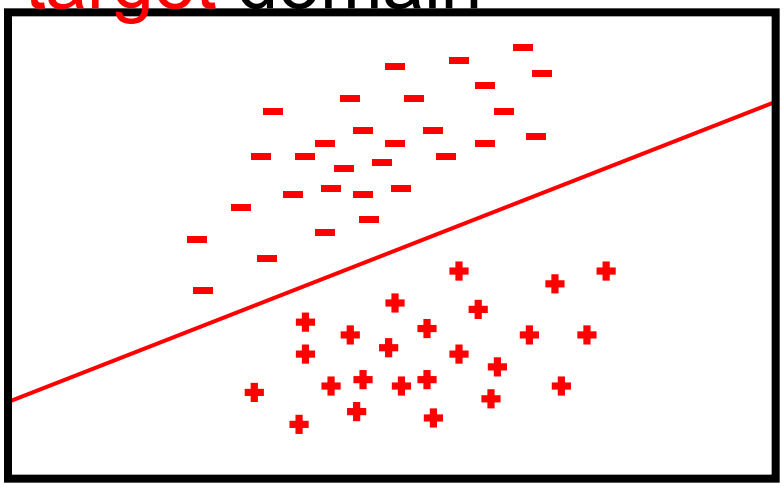
Causes of Dataset Shift

Challenges in correcting the dataset shift generated by the sample selection bias

source domain



target domain

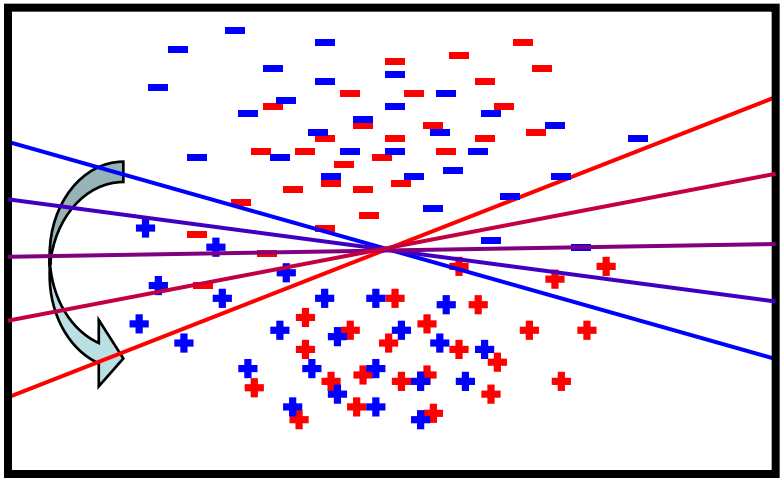


Causes of Dataset Shift

Challenges in correcting the dataset shift generated by the sample selection bias

source domain

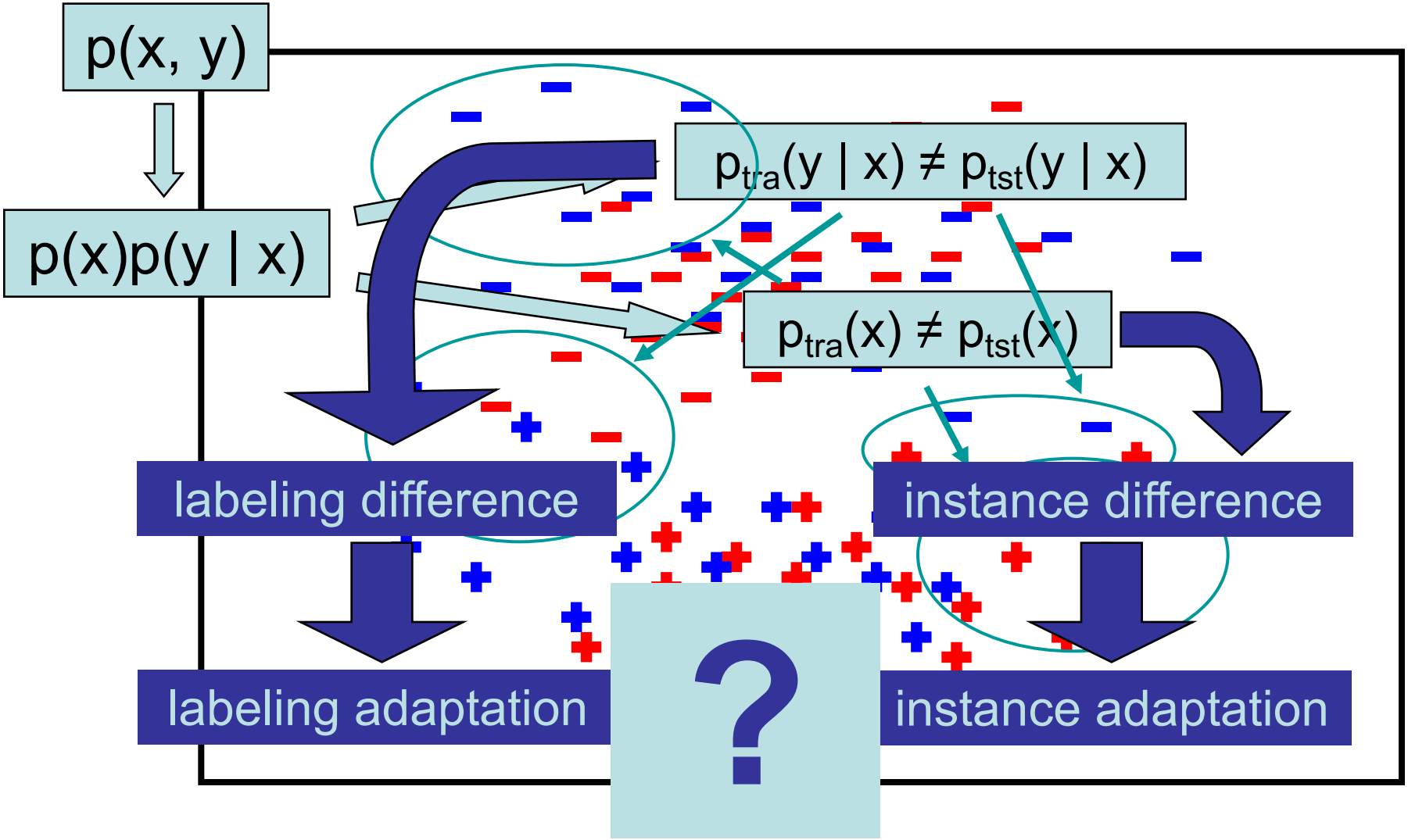
target domain



Causes of Dataset Shift

Challenges in correcting the dataset shift generated by the sample selection bias

Where Does the Difference Come from?



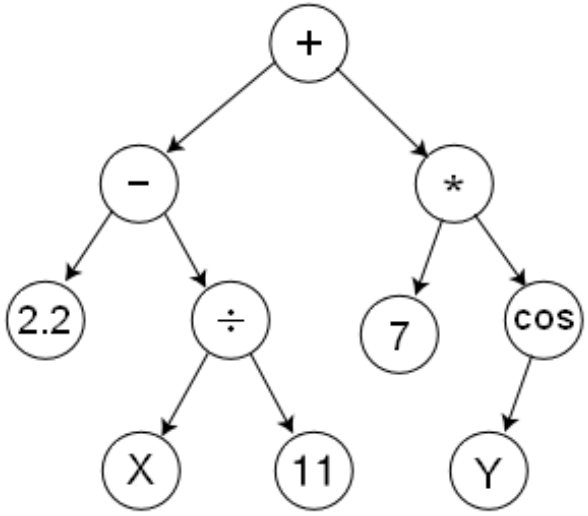
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

GP-RST: From N dimensions to 2

- Goal: obtain a 2-dimensional representation of a given dataset that is as separable as possible.
- Genetic Programming based: evolves 2 trees simultaneously as arithmetic functions of the previous N-dimensions.
- Evaluation of an individual dependant on Rough Set Theory measures.

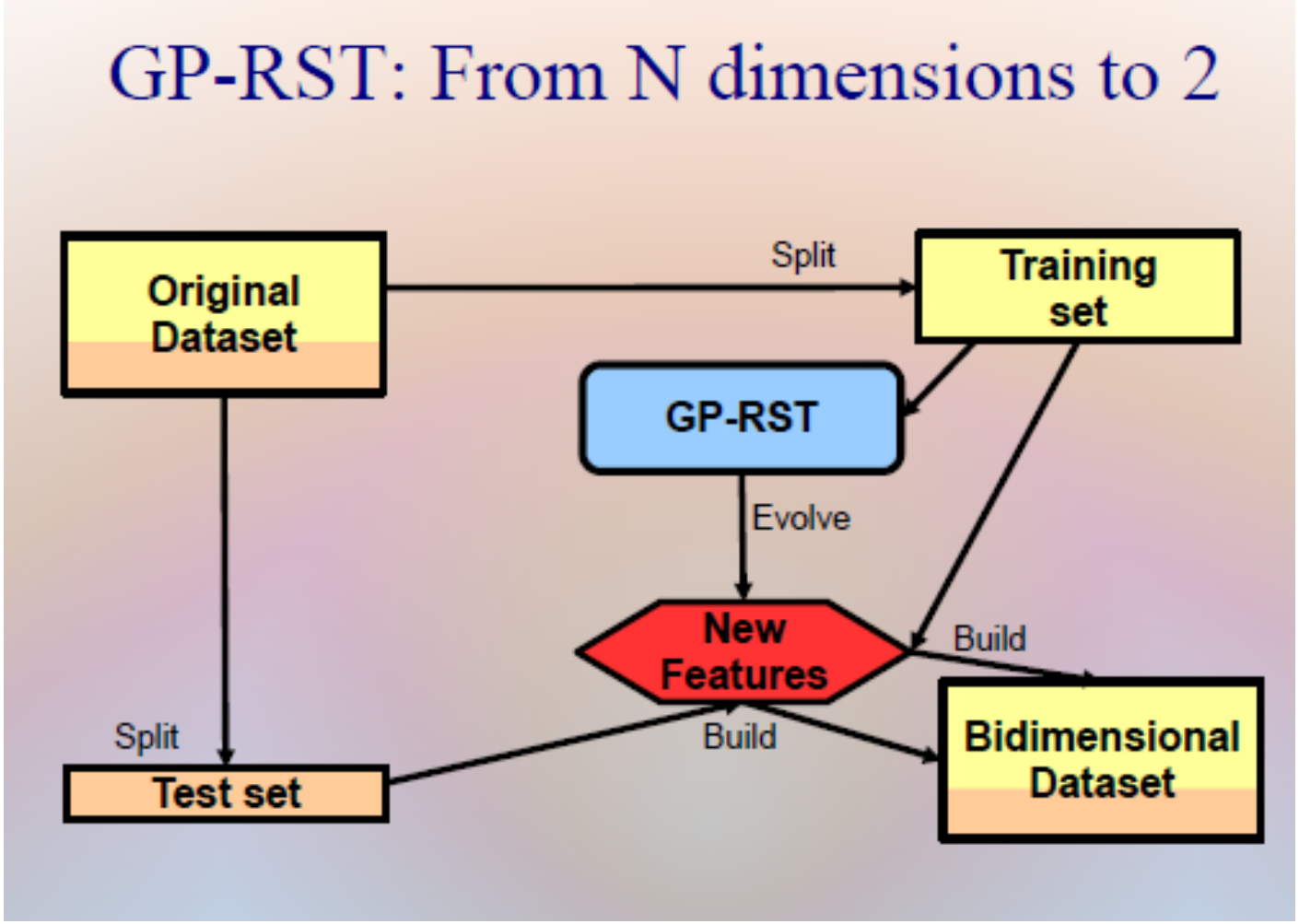


Moreno-Torres, J. G., & Herrera, F. (2010). A preliminary study on overlapping and data fracture in imbalanced domains by means of genetic programming-based feature extraction. In *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA 2010) (pp. 501–506).*

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Data-set shift



A Genetic-Programming based Feature Selection and RST for Visualization of Fracture between Data

The quality of approximation $\gamma(x)$ is the proportion of the elements of a rough set that belong to its lower approximation.

$$B_*(X) = \{x \in X : R'(x) \subseteq X\}$$

$$\gamma(x) = \frac{|B_*(X)|}{|X|}$$

Algorithm 1 Fitness evaluation procedure

1. Obtain $E' = \{e'^h = (f_1(e^h), f_2(e^h), C^h)/h = 1, ..., n_e\}$, where f_1 and f_2 are the expressions encoded on each of the trees of the individual being evaluated.
2. For each class label $C_i \in C : i = 1, ..., n_c$,
   - 2.1 Build a rough set X_i containing all the elements of class C_i.
   - 2.2 Calculate the lower approximation of X_i, $B_*(X_i)$.
   - 2.3 The fitness of the chromosome for class C_i is estimated as the quality of the approximation over X_i, $\gamma(X_i)$.
3. The fitness of the chromosome is the geometric mean of the ones obtained for each class: $fitness = \sqrt[n_c]{\prod_{i=1}^{n_c} \gamma(X_i)}$.

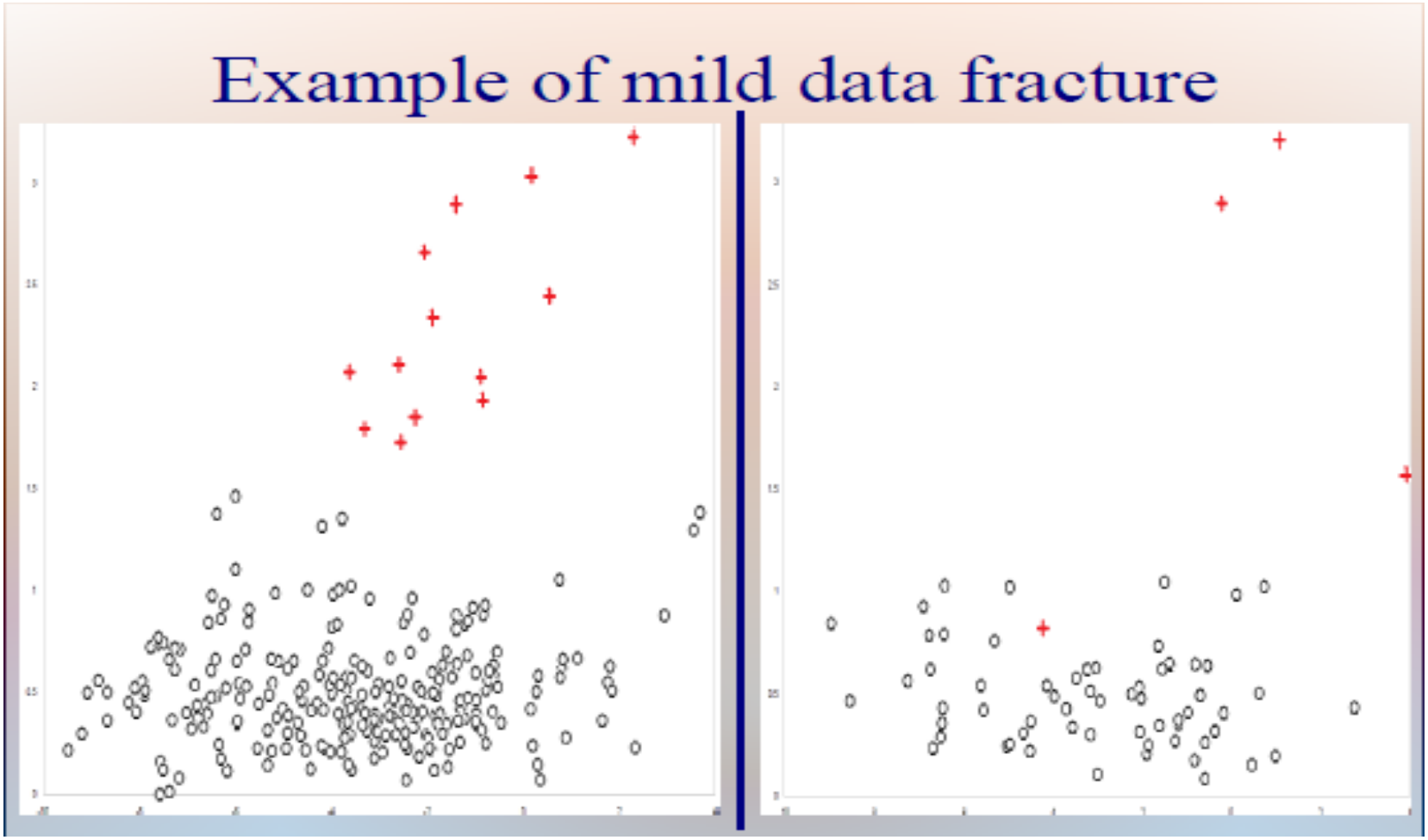
A Genetic-Programming based Feature Selection and RST for Visualization of Fracture between Data

Good behaviour. pageblocks 13v4, 1st partition.

(a) Training set (1.0000) (b) Test set (1.0000)

A Genetic-Programming based Feature Selection and RST for Visualization of Fracture between Data

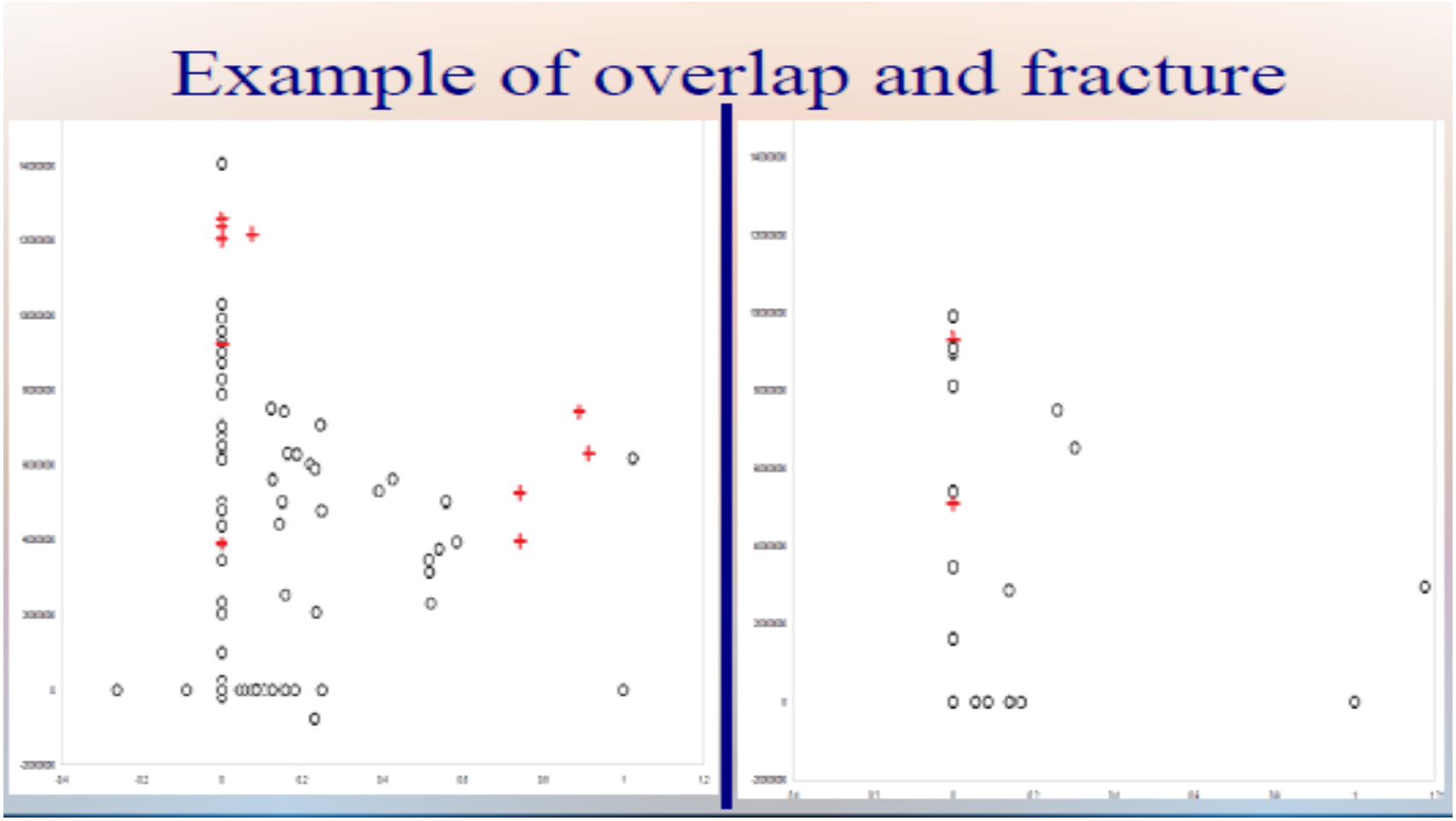
Dataset shift. ecoli 4, 1st partition.



(a) Training set (0.9663) (b) Test set (0.8660)

A Genetic-Programming based Feature Selection and RST for Visualization of Fracture between Data

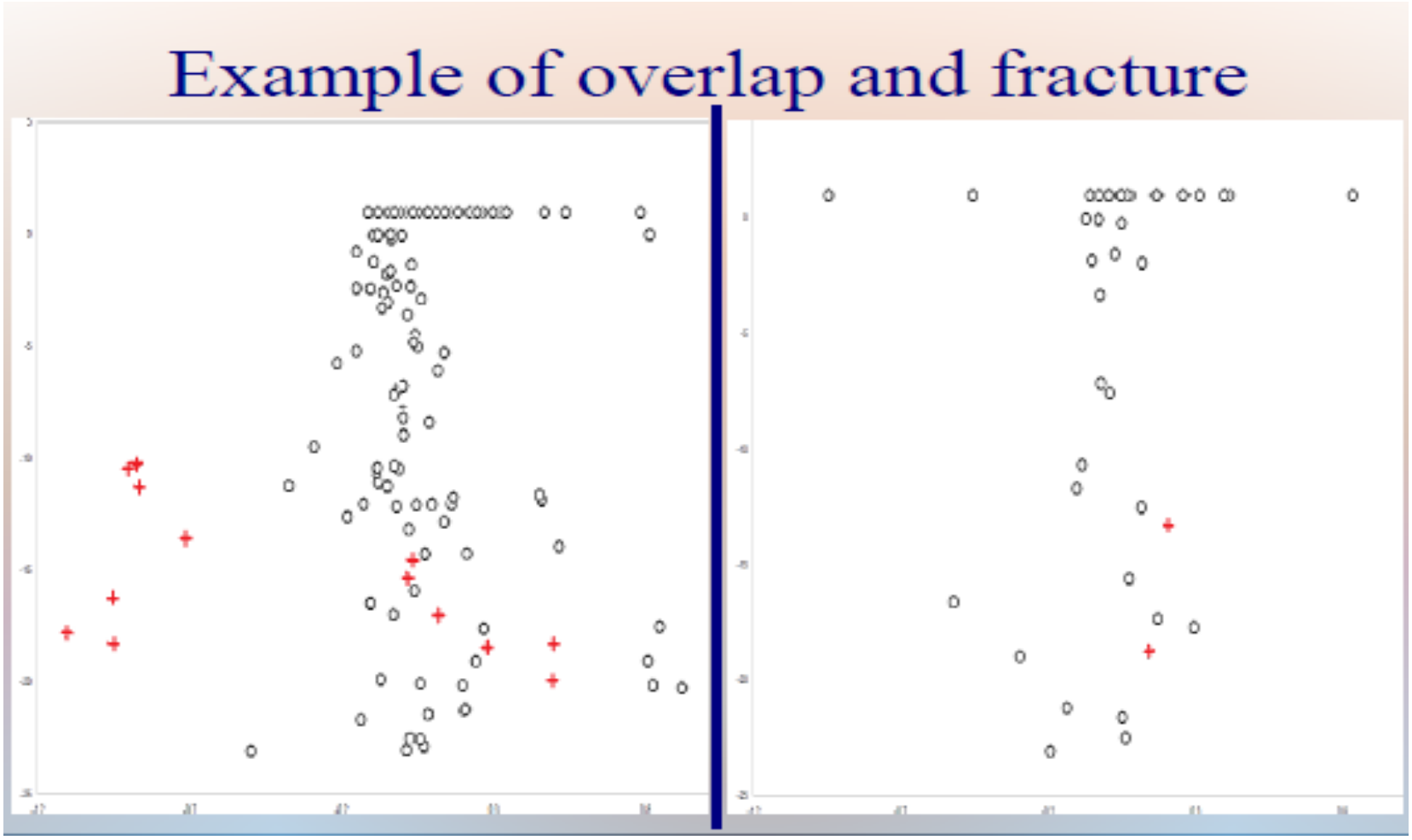
Overlap and dataset shift. glass 016v2, 4th partition.



(a) Training set (0.3779) (b) Test set (0.0000)

A Genetic-Programming based Feature Selection and RST for Visualization of Fracture between Data

Overlap and dataset shift. glass 2, 2nd partition



(a) Training set (0.6794)

(b) Test set (0.0000)

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

There are two different potential approaches in the study of the effect and solution of data-set shift in imbalanced domains.

- The first one focuses on intrinsic data-set shift, that is, the data of interest includes some degree of shift that is producing a relevant drop in performance. In this case, we need to:
  - Develop techniques to discover and measure the presence of data-set shift adapting them to minority classes.
  - Design algorithms that are capable of working under data-set shift conditions. These could be either preprocessing techniques or algorithms that are designed to have the capability to adapt and deal with dataset shift without the need for a preprocessing step.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

- The second branch in terms of data-set shift in imbalanced classification is related to induced data-set shift. Most current state of the art research is validated through stratified cross-validation techniques, which are another potential source of shift in the machine learning process.

A more suitable validation technique needs to be developed in order to avoid introducing data-set shift issues artificially.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

- Imbalanced classification problems are difficult when overlap and/or data fracture are present.
- Single outliers can have a great influence on classifier performance.
- This is a novel problem in imbalanced classification that need a lot of studies.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

What domain characteristics aggravate the problem?

- Overlapping
- Rare sets/ Small disjuncts: The class imbalance problem may not be a problem in itself. Rather, the small disjunct problem it causes is responsible for the decay.
- The overall size of the training set
  large training sets yield low sensitivity to class imbalances
- Noise and border data provokes additional problems.
- An increase in the degree of class imbalance. The data partition provokes data fracture: Dataset shift.

Contents

I. Introduction to imbalanced data sets

II. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics

III. Class imbalance: Data sets, implementations, …

IV. Class imbalance: Trends and final comments

Class Imbalance: Data sets, implementations, …

KEEL Data Mining Tool:
It includes algorithms and data set partitions

http://www.keel.es

KEEL-dataset
Data set repository

KEEL

KNOWLEDGE EXTRACTION based on EVOLUTIONARY LEARNING

Class Imbalance: Data sets, implementations, …

- KEEL is an open source (GPLv3) Java software tool to assess evolutionary algorithms for Data Mining problems including regression, classification, clustering, pattern mining and so on.

- It contains a big collection of classical knowledge extraction algorithms, preprocessing techniques.
- It includes a large list of algorithms for imbalanced data.

Imbalanced Classification (42)	Resampling Data Space (20)	Over-sampling Methods (12)
		Under-sampling Methods (8)
	Cost-Sensitive Classification (3)	
	Ensembles for Class Imbalance (19)	

Class Imbalance: Data sets, implementations, …

- We include 111 data sets:
66 for 2 classes,
15 for multiple classes and
30 for noise and bordeline.

Imbalanced data sets

We divide our Imbalanced data sets into the following sections:

- Imbalance ratio between 1.5 and 9
- Imbalance ratio higher than 9 - Part I
- Imbalance ratio higher than 9 - Part II
- Multiple class imbalanced problems
- Noisy and Borderline Examples

We also include the preprocessed data sets.

Contents

I. Introduction to imbalanced data sets

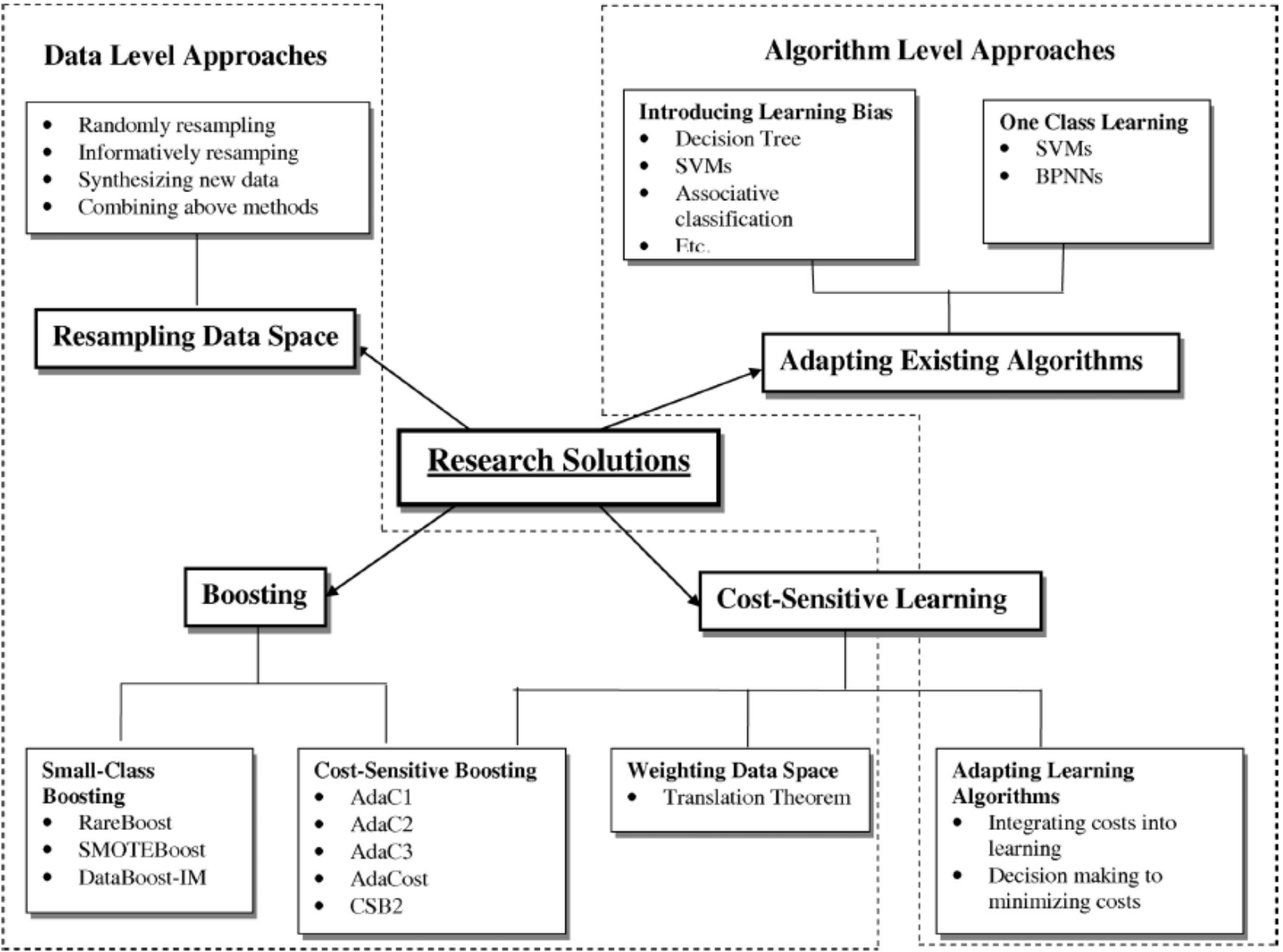
II. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics

III. Class imbalance: Data sets, implementations, …

IV. Class imbalance: Trends and final comments

Class Imbalance: Trends and final comments

Data level vs algorithm Level



Y. Sun, A. K. C. Wong and M. S. Kamel. Classification of imbalanced data: A review. International Journal of Pattern Recognition 23:4 (2009) 687-719.

Class Imbalance: Trends and final comments New studies, trends and challenges

- Improvements on resampling – specialized resampling
  - New approches for creating artificial instances
  - How to choose the amount to sample?
  - New hybrid approaches oversampling vs undersampling
- Cooperation between resampling/cost sensitive/boosting
- Cooperation between feature selection and resampling
- Scalability: high number of features and sparse data
- Intrinsic data characteristics. To analyze the challenges on the class distribution.

Class Imbalance: Trends and final comments New studies, trends and challenges

In short, it is necessary to do work for:

- **Establishing some fundamental results regarding:**
  a) **the nature of the problem,**
  b) **the behaviour of different types of classifiers, and**
  c) **the relative performance of various previously proposed schemes for dealing with the problem.**

- **Designing new methods addressing the problem.**
  Tackling data preprocessing and changing rule classification strategy.

Class Imbalance: Trends and final comments Final comments

- Class imbalance is a challenging and critical problem in the knowledge discovery field, the classification with imbalanced data sets.

- Due to the intriguing topics and tremendous potential applications, the classification of imbalanced data will continue to receive more and more attention along next years. Class of interest is often much smaller or rarer (minority class).

Sistemas Inteligentes para la Gestión de la Empresa

TEMA 3. Análisis Predictivo para la Empresa

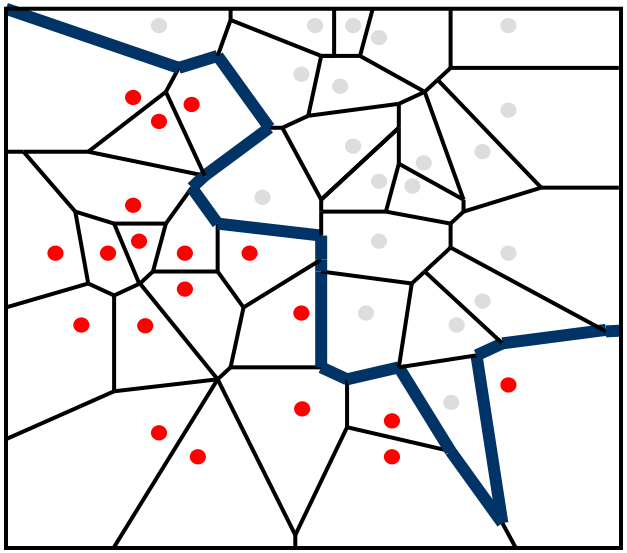
(Modelos predictivos avanzados de clasificación)

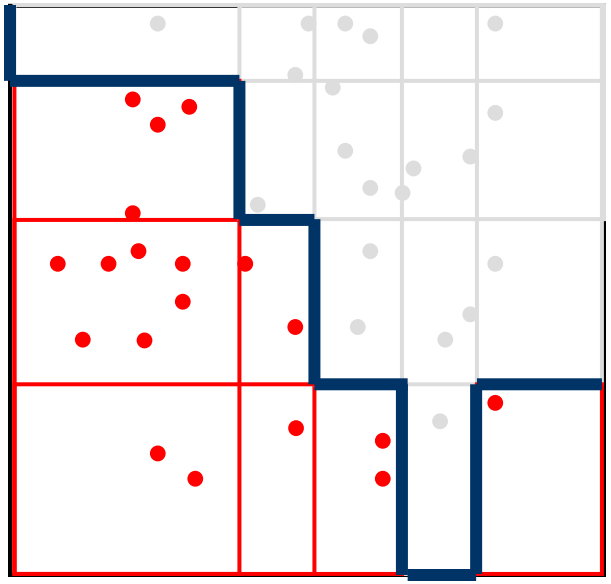
1. Clasificación no balanceada
2. Multiclasificadores: Bagging y Boosting
3. Multiples clases: Descomposición binaria
4. Redes Neuronales y Maquinas de soporte Vectorial

2. Multiclasificadores

Algunos ejemplos de clasificadores sencillos

Clasificador Del Vecino Más Cercano (k-NN): Basado en distancias (muy diferente a los basados en particiones)

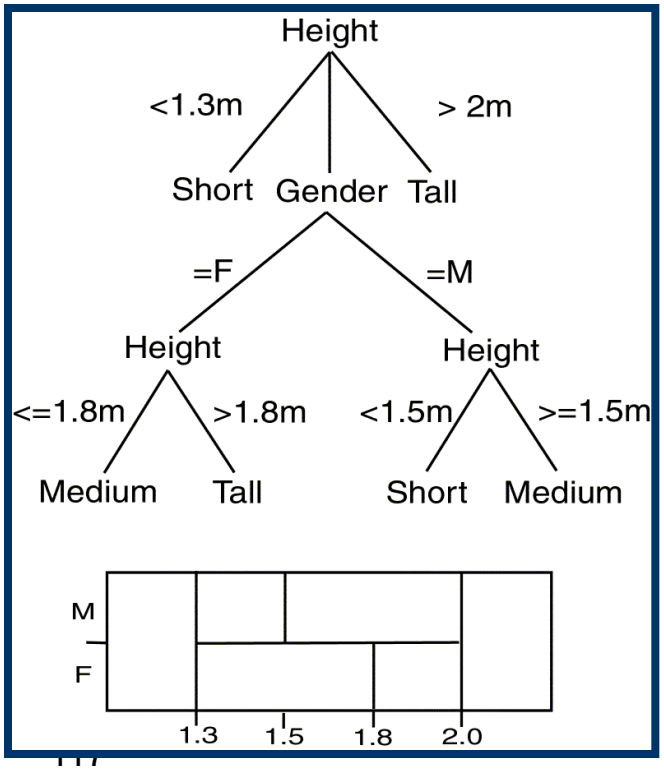


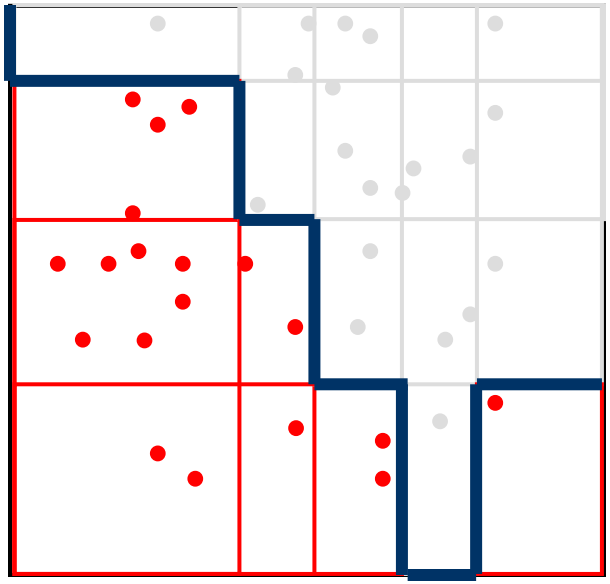


2. Multiclasificadores

Algunos ejemplos de clasificadores sencillos

Clasificador basado en particiones y reglas – Arboles de decisión

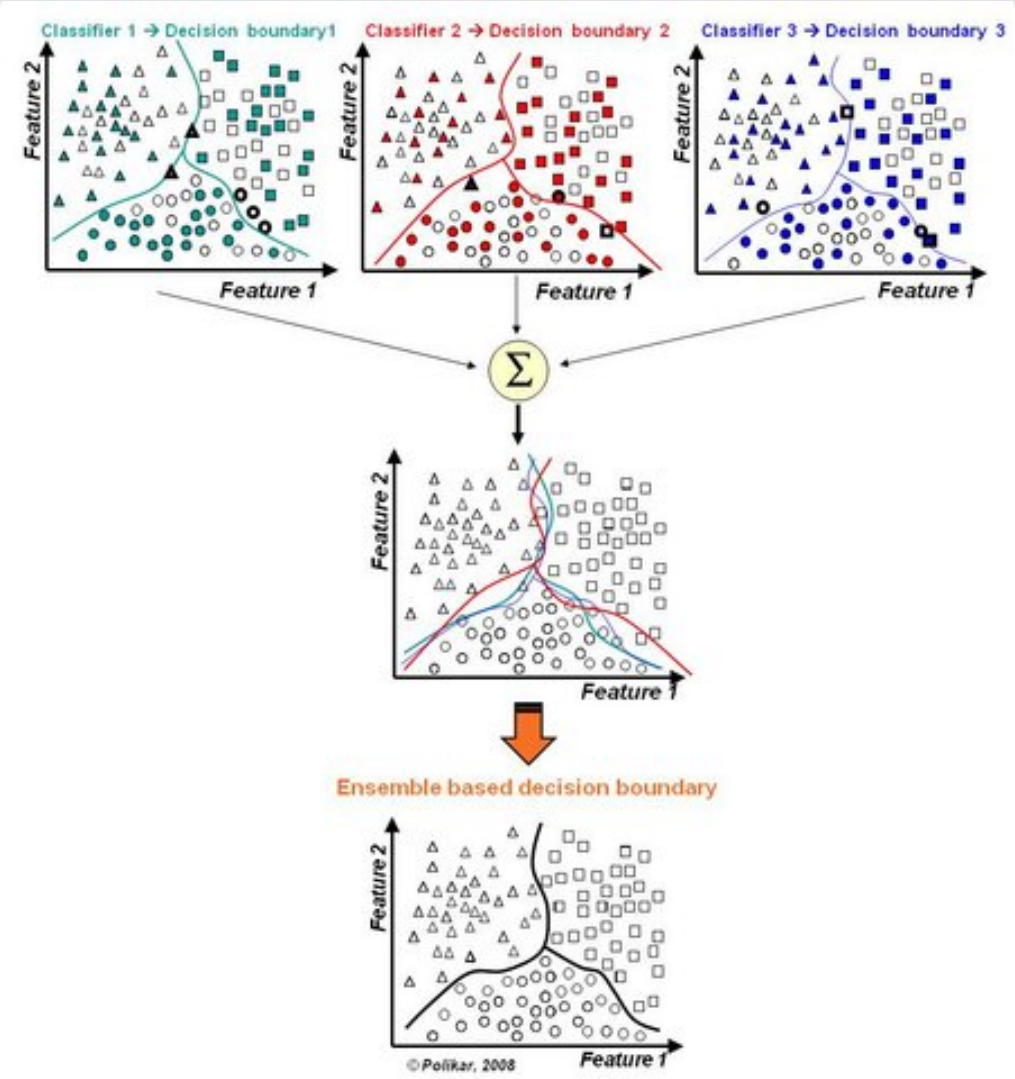




2. Multiclasificadores

- La idea es inducir *n* clasificadores en lugar de uno solo
- Para clasificar se utilizará una combinación de la salida que proporciona cada clasificador
- Los clasificadores pueden estar basados en distintas técnicas (p.e. árboles, reglas, instancias,…)
- Se puede aplicar sobre el mismo clasificador o con diferentes.
- Modelos específicos:
  - *Bagging*: cada clasificador se induce independientemente incluyendo una fase de diversificación sobre los datos
  - *Boosting*: cada clasificador tiene en cuenta los fallos del anterior

2. Multiclasificadores



2. Multiclasificadores: Bagging

- Bagging definido a partir de "**b**ootstrap **agg**regat**ing**".

- Un método para combinar múltiples predictores/clasificadores.

- Bagging funciona bien para los algoritmos de aprendizaje "inestables", aquellos para los que un pequeño cambio en el conjunto de entrenamiento puede provocar grandes cambios en la predicción (redes neuronales, árboles de decisión, redes neuronales, …) (No es el caso de k-NN que es un algoritmo estable)

2. Multiclasificadores: Bagging

Bagging.[Breiman,94] Repeat for $t = 1, \ldots, T$:

- Select, at random *with replacement*, N training examples.
- Train learner on selected samples to generate h_t

Final hypothesis is simple vote:

$$H(x) = MAJ(h_1(\mathbf{x}), \ldots, h_T(\mathbf{x}))$$

2. Multiclasificadores: Bagging

- Fase 1: Generación de modelos
  1. Sea *n* el número de ejemplos en la BD y *m* el número de los modelos a utilizar
  2. Para *i*=1,…,*m* hacer
     - Muestrear con reemplazo *n* ejemplos de la BD (Boosting)
     - Aprender un modelo con ese conjunto de entrenamiento
     - Almacenarlo en modelos[*i*]

- Fase 2: Clasificación
  1. Para i=1,…,m hacer
     - Predecir la clase utilizando modelos[i]
  2. Devolver la clase predicha con mayor frecuencia

2. Multiclasificadores: Bagging

Algorithm 1 Bagging

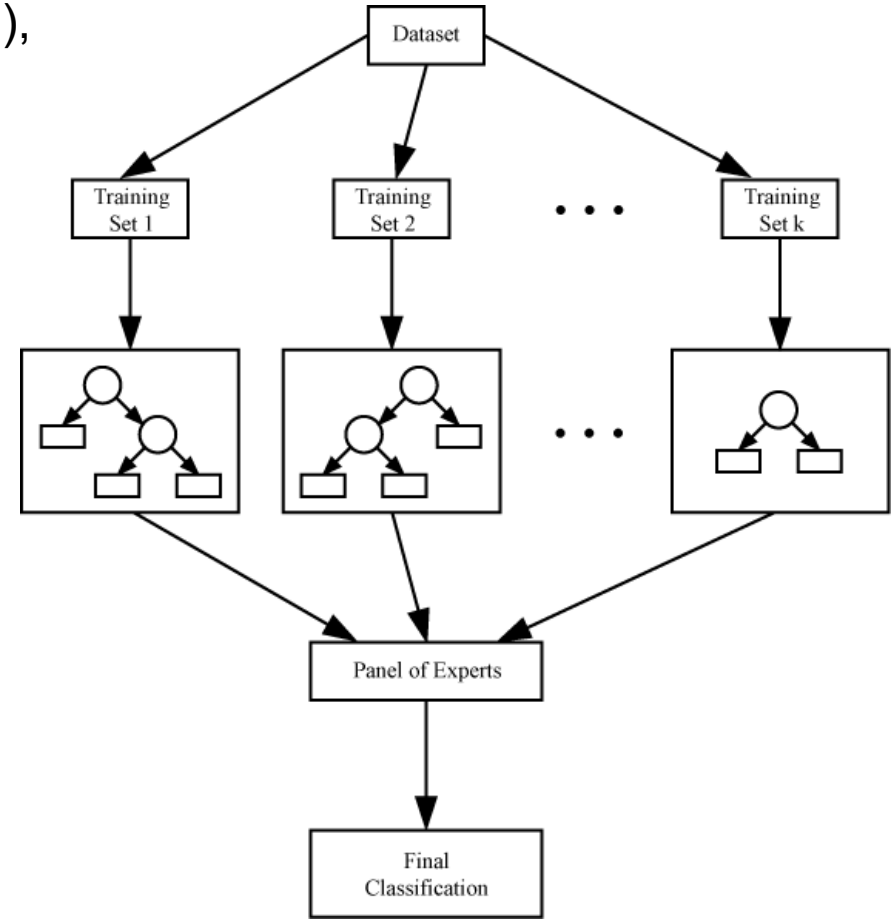
Input: S: Training set; T: Number of iterations;
n: Bootstrap size; I: Weak learner

Output: Bagged classifier: $H(x) = sign\left(\sum_{t=1}^{T} h_t(x)\right)$ where $h_t \in [-1, 1]$ are the induced classifiers

1: **for** $t = 1$ to T **do**
2: $S_t \leftarrow$ RandomSampleReplacement(n,S)
3: $h_t \leftarrow$ I(S_t)
4: **end for**

2. Multiclasificadores: Bagging

Random Forest (Leo Breiman, Adele Cutler): Boostraping (muestreo con reemplazamiento, 66%), Selección aleatoria de un conjunto muy pequeño de variables (m<<M) (ej. log M + 1) para elegir entre ellas el atributo que construye el árbol (medida de Gini), sin poda (combinación de clasificadores débiles)



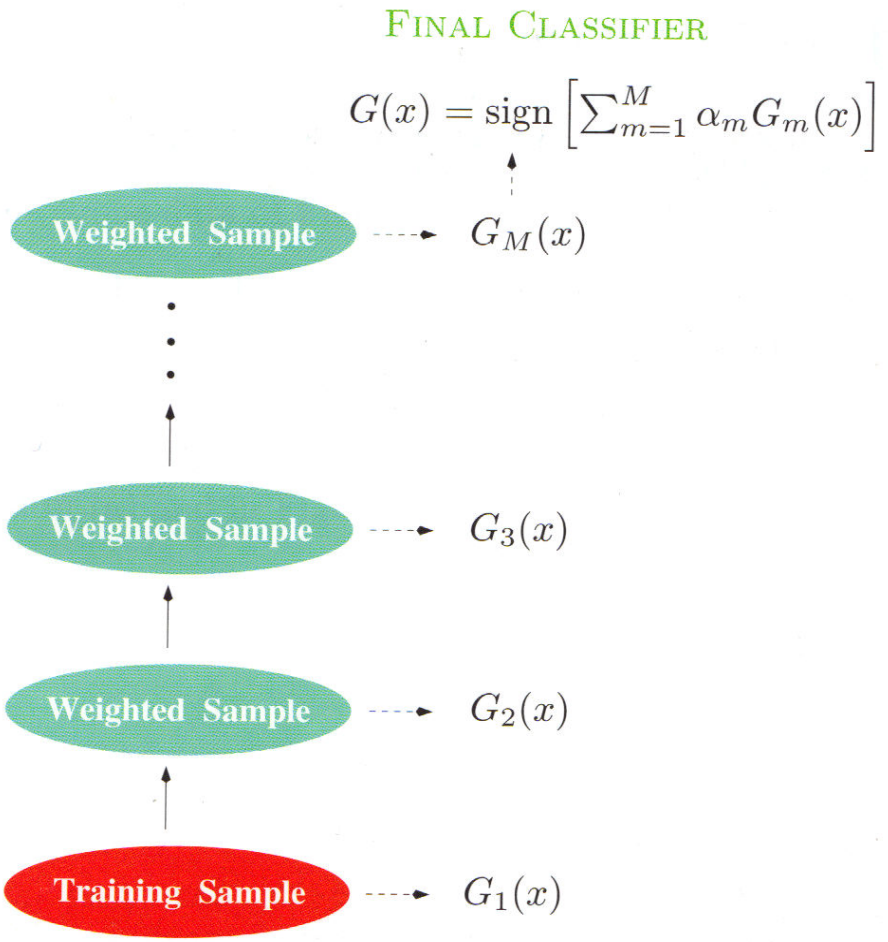
Breiman, Leo (2001). «Random Forests». *Machine Learning* **45** (1): pp. 5–32. doi:10.1023/A:1010933404324

3. Multiclasificadores: Boosting

- **Muestreo ponderado (ejemplos):**
  - En lugar de hacer un muestreo aleatorio de los datos de entrenamiento, se ponderan las muestras para concentrar el aprendizaje en los ejemplos más difíciles.
  - Intuitivamente, los ejemplos más cercanos a la frontera de decisión son más difíciles de clasificar, y recibirán pesos más altos.
- **Votos ponderados (clasificadores):**
  - En lugar de combinar los clasificadores con el mismo peso en el voto, se usa un voto ponderado.
  - Esta es la regla de combinación para el conjunto de clasificadores débiles.
  - En conjunción con la estrategia de muestreo anterior, esto produce un clasificador más fuerte.

3. Multiclasificadores: Boosting

Idea



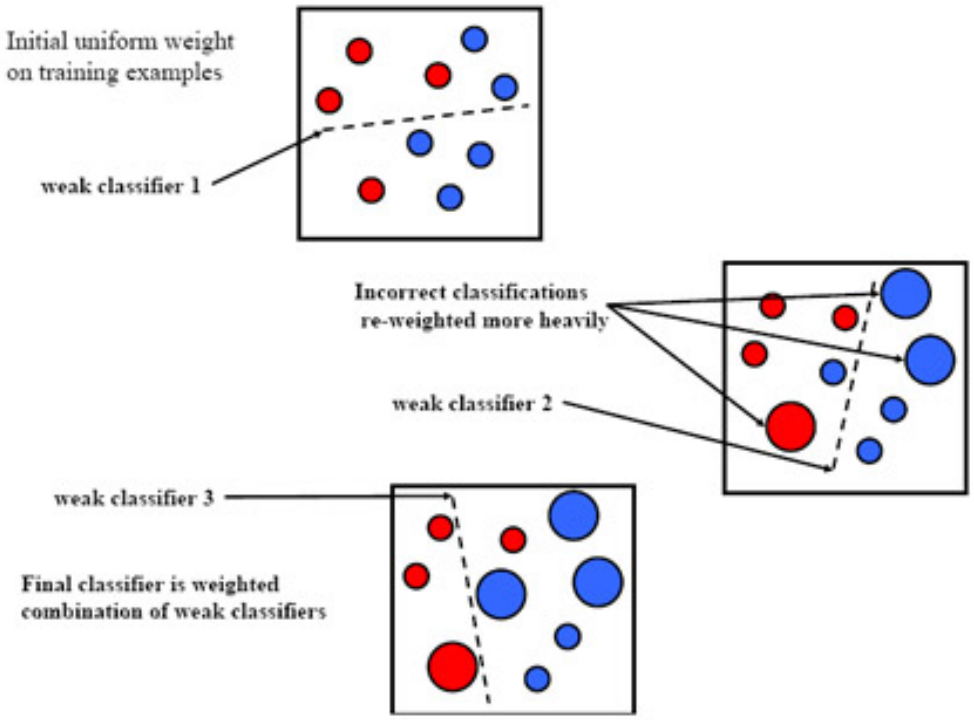
3. Multiclasificadores: Boosting

- Los modelos no se construyen independientemente, sino que el modelo i-ésimo está influenciado por los anteriores
- La idea es prestar más atención a los ejemplos mal clasificados por los modelos anteriores
- Fase 1: Generación de modelos
  1. Asignar a cada ejemplo el mismo peso (*1/n*)
  2. Para *i=1,…,m* hacer
     - Aprender un modelo a partir de la BD con pesos
     - Almacenarlo en modelos[*i*]
     - Calcular el error ε_i sobre el conjunto de ejemplos
     - Si $\varepsilon_i=0$ o $\varepsilon_i>=0.5$ terminar
     - Para cada ejemplo bien clasificado multiplicar su peso por $\varepsilon_i/(1-\varepsilon_i)$
     - Normalizar los pesos de todos los ejemplos
- Fase 2: Clasificación
  1. Asignar peso cero a todas las categorías de la variable clase
  2. Para *i=1,…,m* hacer
     - Sumar $-log(\varepsilon_i/(1-\varepsilon_i))$ a la categoría predicha por modelos[*i*]
  3. Devolver la categoría con mayor peso

3. Multiclasificadores: Boosting

AdaBoost, abreviatura de "Adaptive Boosting", es un algoritmo de aprendizaje meta-algoritmo formulado por Yoav Freund y Robert Schapire que ganó el prestigioso "Premio Gödel" en 2003 por su trabajo. Se puede utilizar en conjunción con muchos otros tipos de algoritmos de aprendizaje para mejorar su rendimiento.

La salida de los otros algoritmos de aprendizaje ("algoritmos débiles") se combina en una suma ponderada que representa la salida final del clasificador impulsado.



$$H(x) = sign(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x))$$

3. Multiclasificadores: Boosting

Algorithm 1: Adaboost Algorithm (Freund and Schapire)

Input : A weak learning algorithm $WeakLearn$, an integer T specifying number of iterations, and N labelled training data $\{(x_1, y_1), \ldots, (x_N, y_N)\}$.

Output : A strong classifier F.

Initialize the weight vector $w_i^1 = \frac{1}{N}$, *for* $i = 1, \ldots, N$.

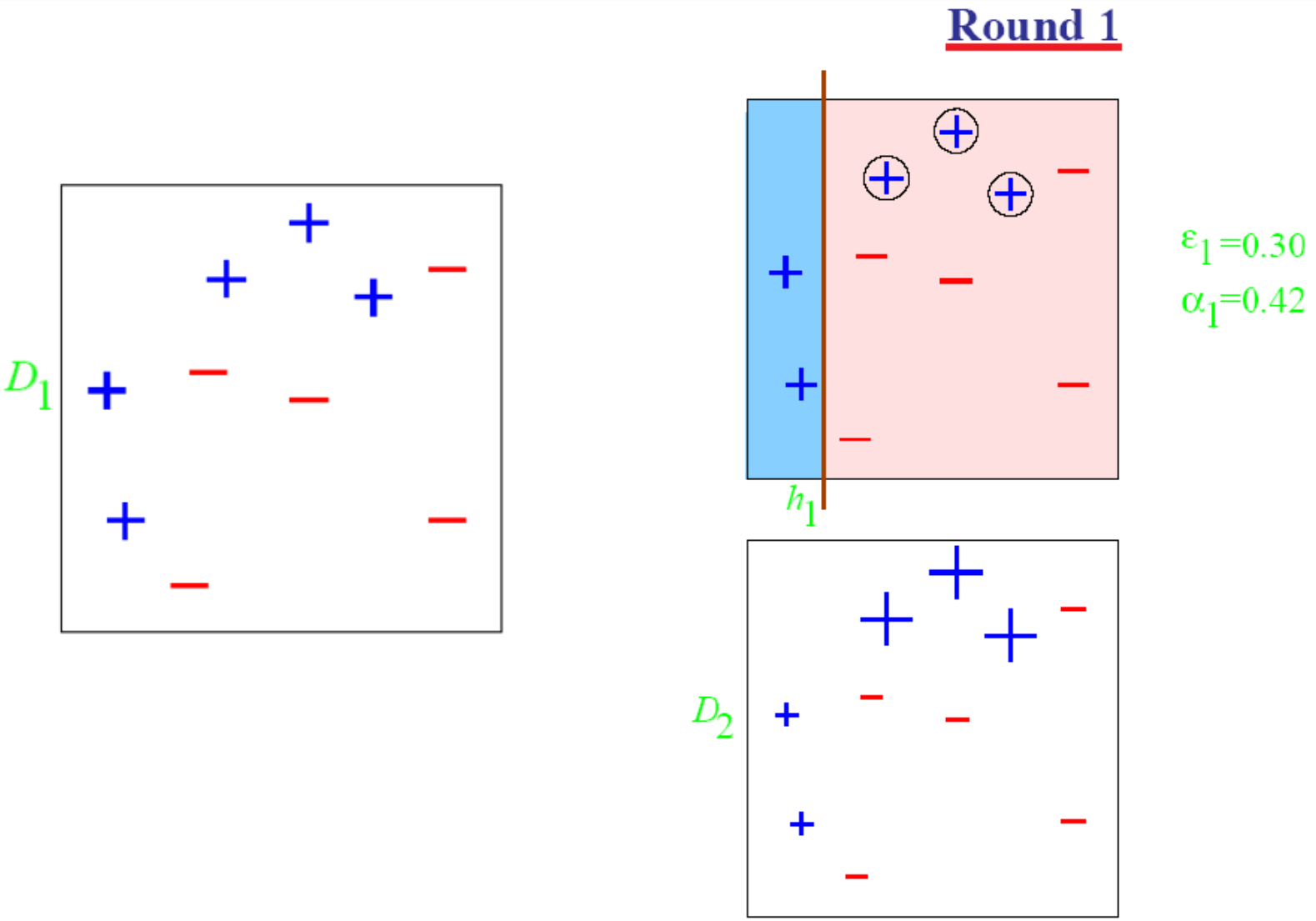
for $t \leftarrow 1, 2, \ldots, T$ **do**

1. $\mathbf{p}^t \leftarrow \mathbf{w}^t / \sum_{i=1}^{N} w_i^t$.
2. Call $WeakLearn$, providing it with the distribution on $\mathbf{p}^t$; get back a weak learner $h_t : X \rightarrow \pm 1$.
3. Calculate the weight error of h_t: $\epsilon_t = \sum_{i=1}^{N} p_i^t \frac{1}{2} |h_t(x_i) - y_i|$.
4. $\alpha_t \leftarrow \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
5. $w_i^{t+1} \leftarrow w_i^t \exp\left(\alpha_t \frac{1}{2} |h_t(x_i) - y_i|\right)$, for $i = 1, 2, \ldots, T$.

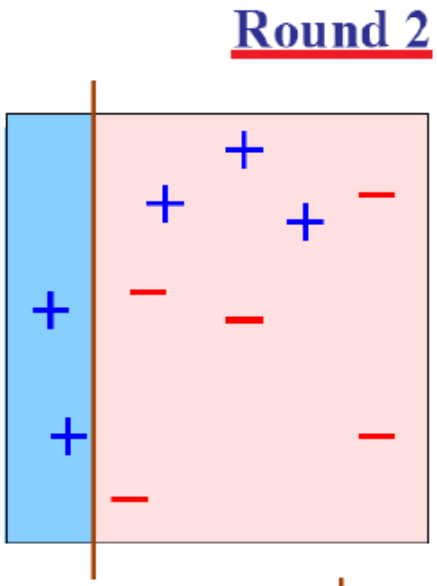
Output the final strong classifier:

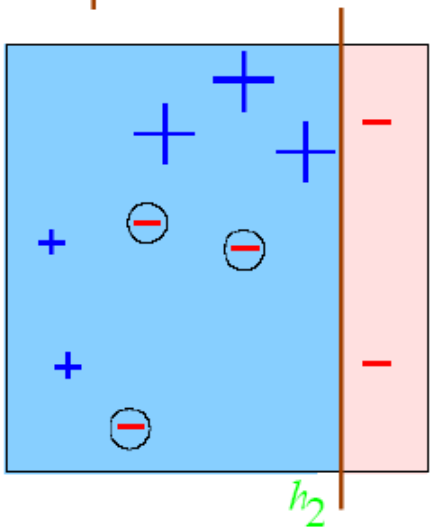
$$F(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{T} \alpha_t h_t(x) \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

3. Boosting. Ejemplo



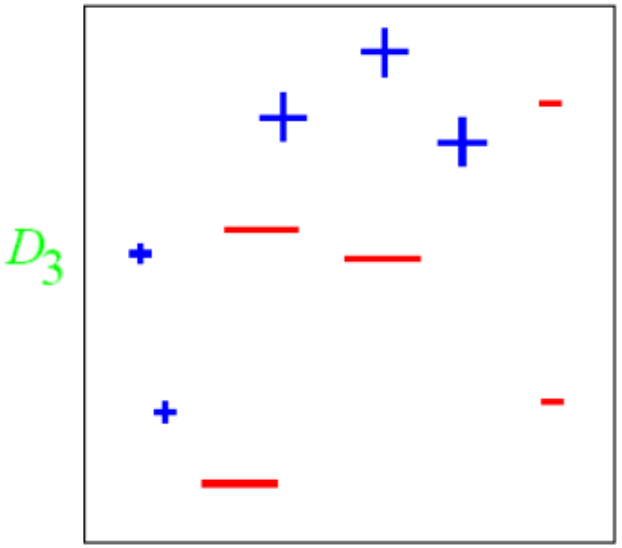
3. Boosting. Ejemplo



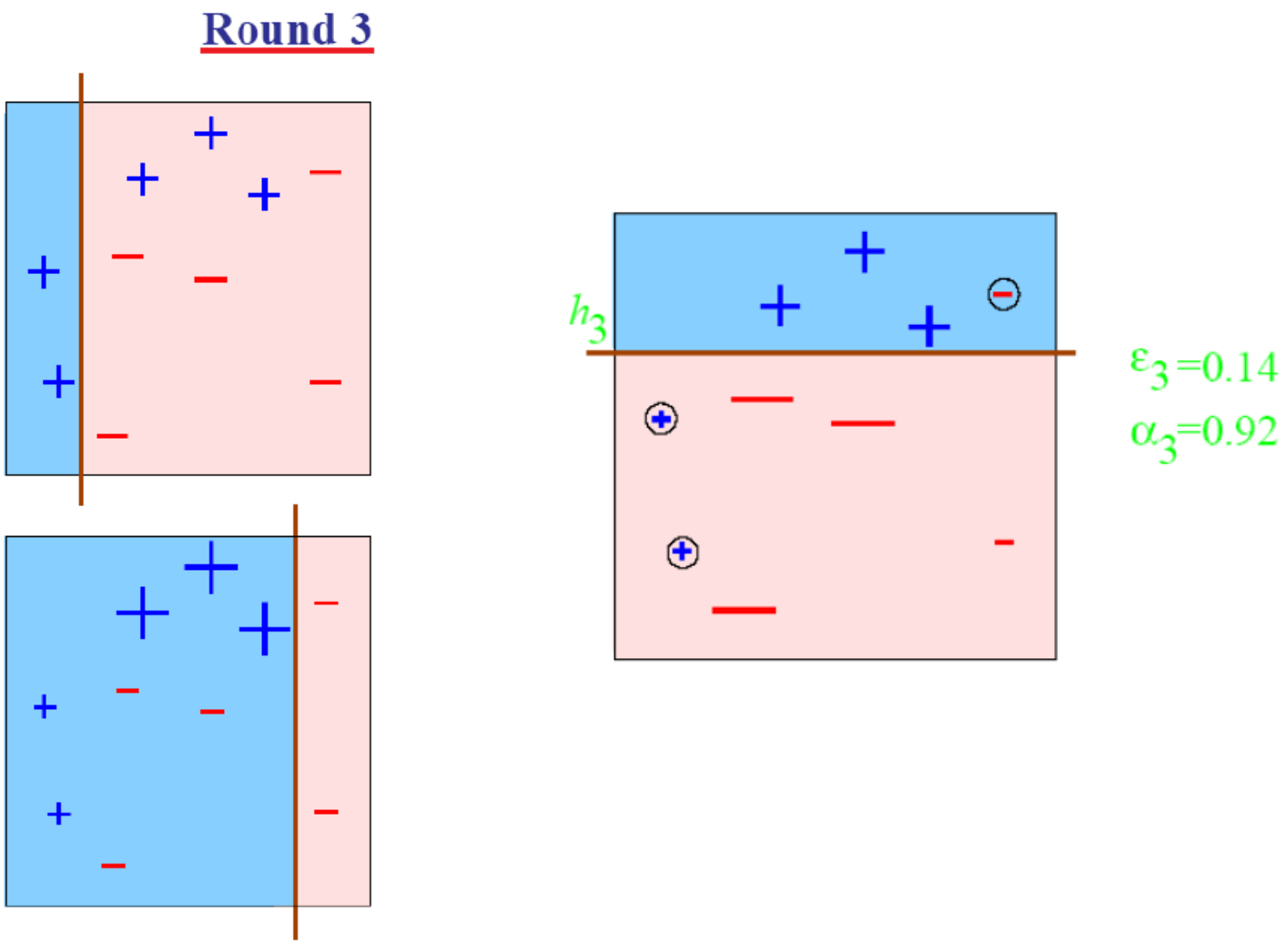


$\varepsilon_2=0.21$

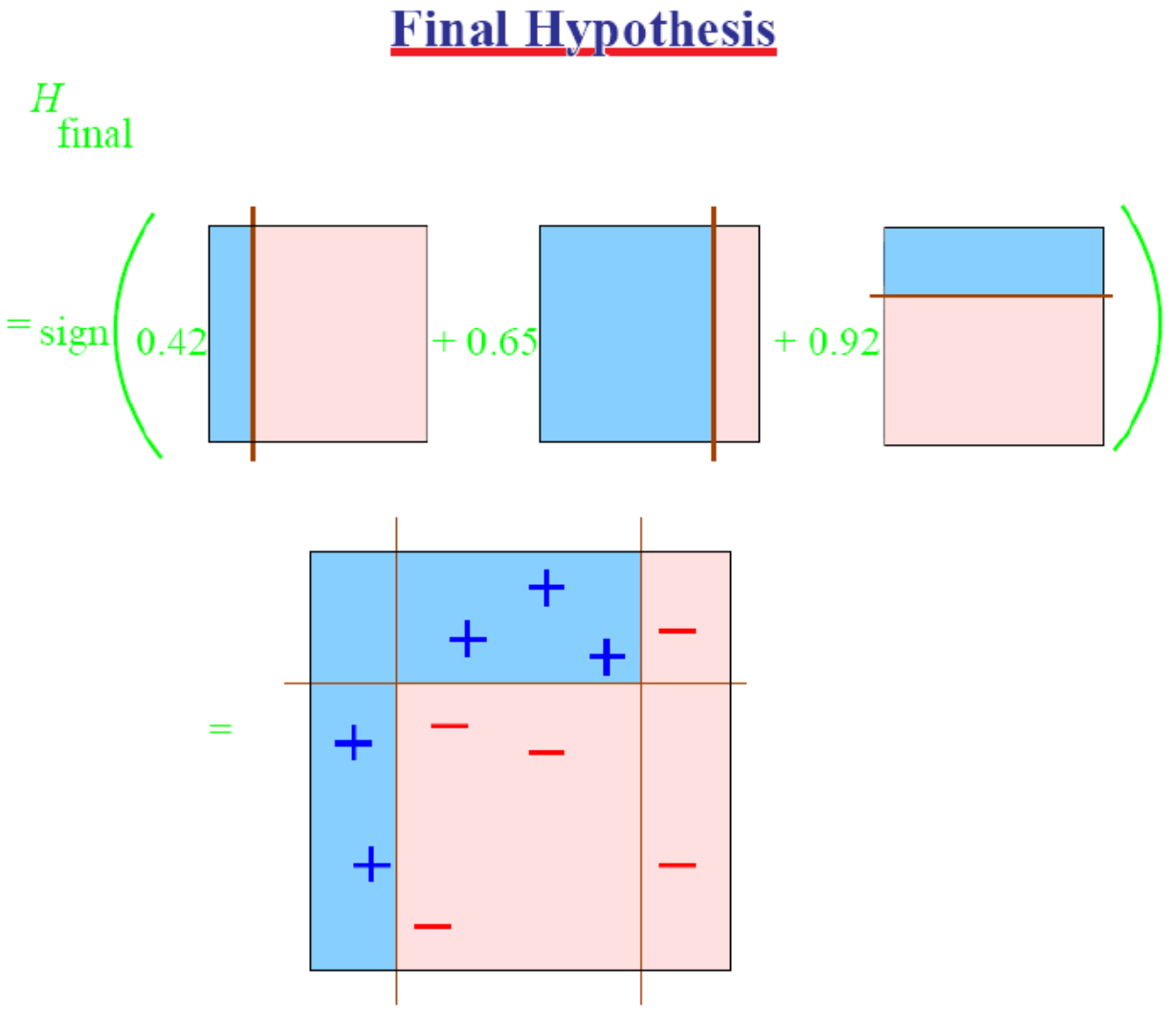
$\alpha_2=0.65$



3. Boosting. Ejemplo

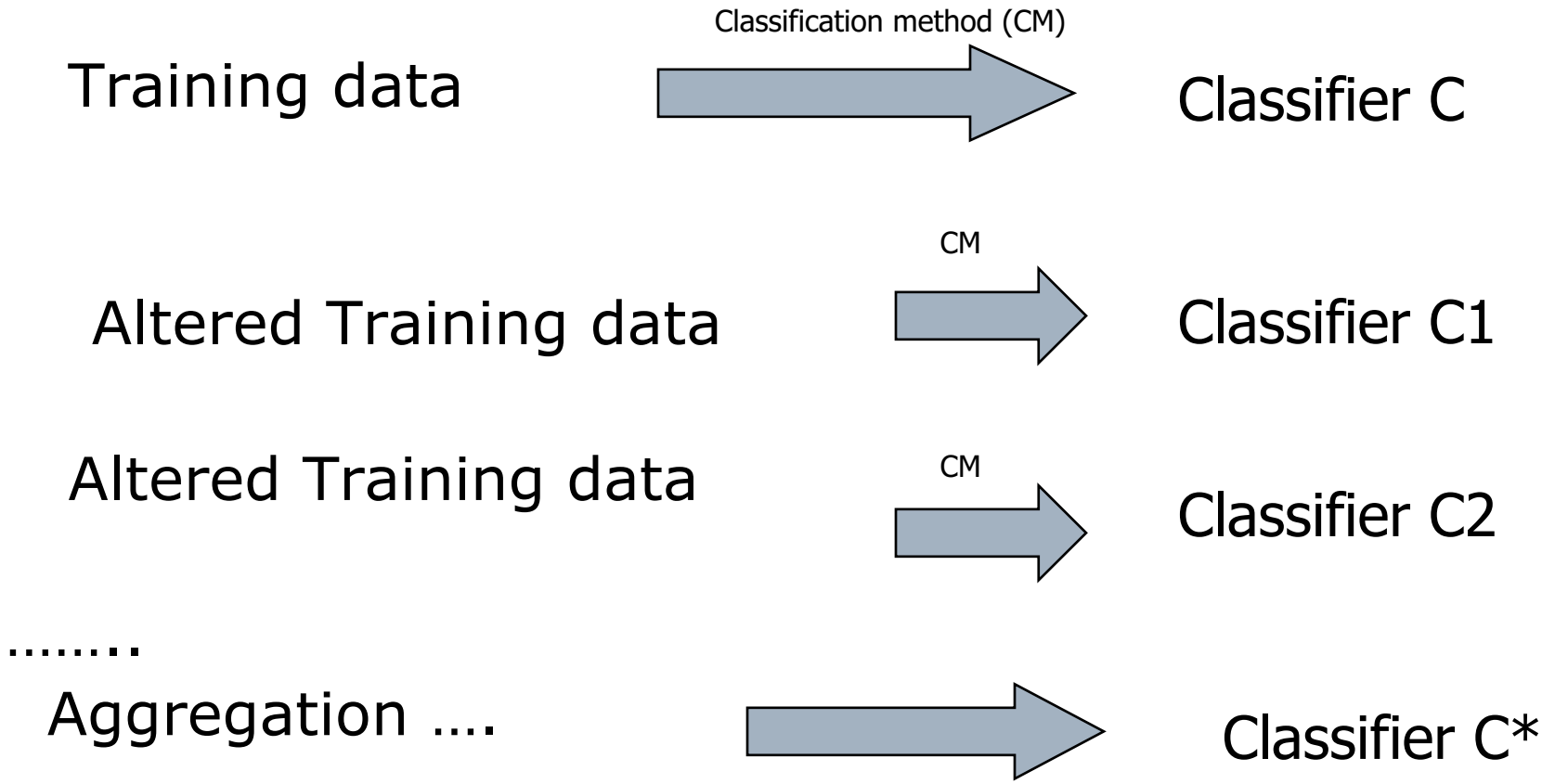


3. Boosting. Ejemplo



Bagging and Boosting

Resumen: Idea General

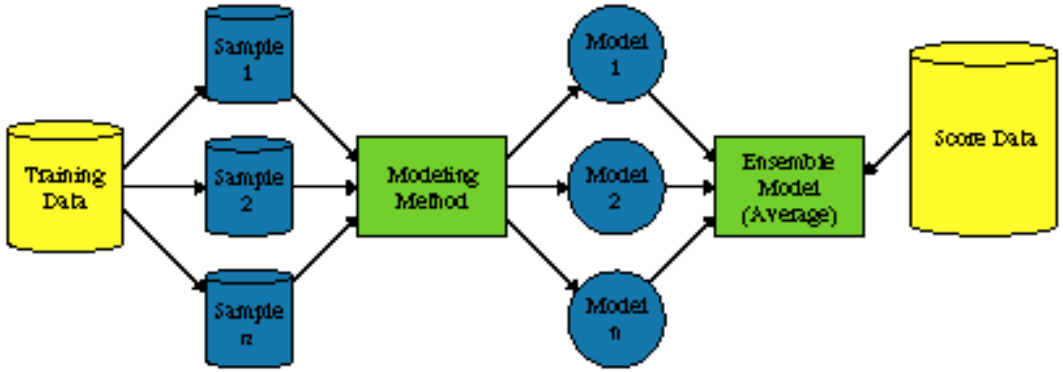


Bagging and Boosting

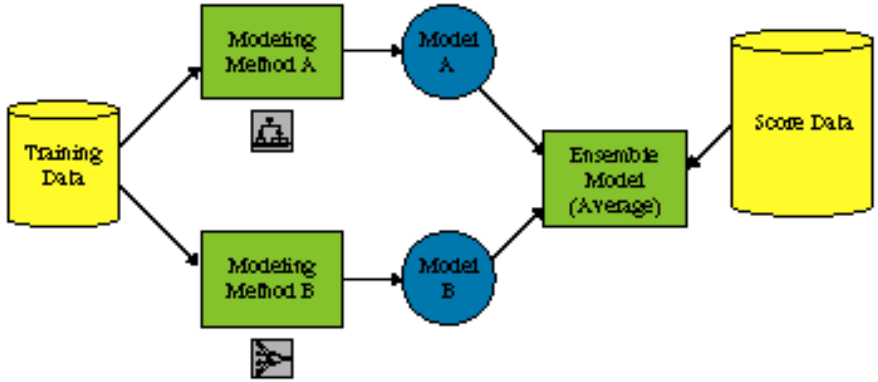
Bagging and Boosting

Resumen:

- Bagging= *Manipulation with data set*



- Boosting = *Manipulation with model*



Bagging and Boosting

Bagging and Boosting, **Poda de los clasificadores**

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 31, NO. 2, FEBRUARY 2009 245

An Analysis of Ensemble Pruning Techniques Based on Ordered Aggregation

Gonzalo Martínez-Muñoz, Daniel Hernández-Lobato, and Alberto Suárez, *Member, IEEE*

La idea general consiste en crear un pool de clasificadores más grande de lo habitual y luego reducirlo para quedarnos con los que forman el ensemble más preciso. Esta forma de selección de clasificadores se denomina "ensemble pruning".
Los modelos basados en ordenamiento generalmente parten de añadir 1 clasificador al modelo final. Posteriormente, en cada iteración añaden un nuevo clasificador del pool de los no seleccionados en base a una medida establecida. Y generalmente lo hacen hasta llegar a un número de clasificadores establecido. Según este artículo, en Bagging, teniendo un pool de 100 clasificadores, es suficiente con usar 21.
Boosting-based (**BB**): Hace un boosting a posteriori (muy interesante!). Coge el clasificador que minimiza más el coste respecto a boosting, pero no los entrena respecto a esos costes porque los clasificadores ya están en el pool.

Sistemas Inteligentes para la Gestión de la Empresa

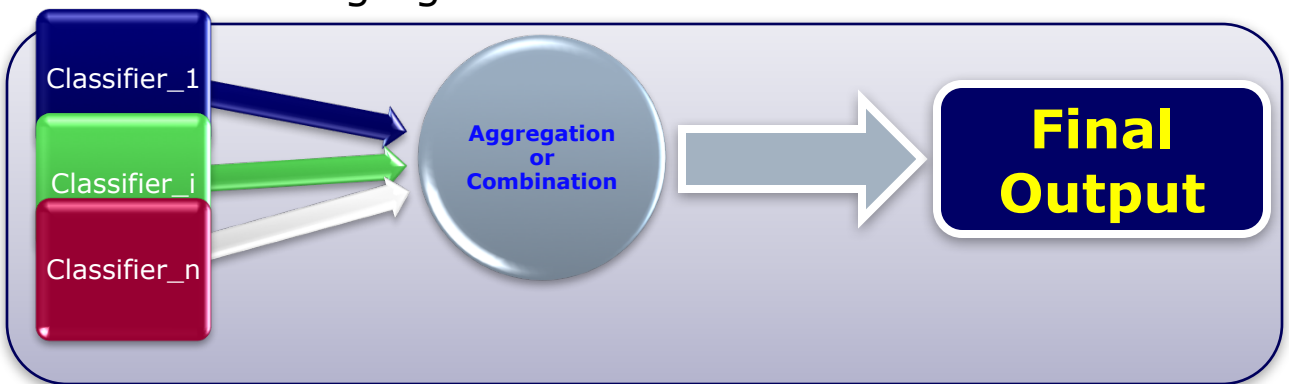
TEMA 3. Análisis Predictivo para la Empresa

(Modelos predictivos avanzados de clasificación)

1. Clasificación no balanceada
2. Multiclasificadores: Bagging y Boosting
3. Multiples clases: Descomposición binaria
4. Redes Neuronales y Maquinas de soporte Vectorial

3. Descomposición de problemas multiclase. Binarización

- Descomposición de un problema con múltiples clases
  - Estrategia Divide y Vencerás
  - Multi-clase → Es más fácil resolver problemas binarios
    - Para cada problema binario
      - 1 clasificador binario = clasificador base
    - Problem
      - ¿Cómo hacer la descomposición?
      - ¿Cómo agregar las salidas?



3. Descomposición de problemas multiclase. Binarización

Una aplicación real en KAGGLE de Problema Multiclase

otto group

$10,000 • 1,987 teams

Otto Group Product Classification Challenge

Enter/Merge by

Tue 17 Mar 2015 — Mon 18 May 2015 (39 days to go)

Classify products into the correct category

The Otto Group is one of the world's biggest e-commerce companies, with subsidiaries in more than 20 countries, including Crate & Barrel (USA), Otto.de (Germany) and 3 Suisses (France). We are selling millions of products worldwide every day, with several thousand products being added to our product line.

A consistent analysis of the performance of our products is crucial. However, due to our diverse global infrastructure, many identical products get classified differently. Therefore, the quality of our product analysis depends heavily on the ability to accurately cluster similar products. The better the classification, the more insights we can generate about our product range.

3. Descomposición de problemas multiclase. Binarización

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.

3. Descomposición de problemas multiclase. Binarización

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

3. Descomposición de problemas multiclase. Binarización

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,

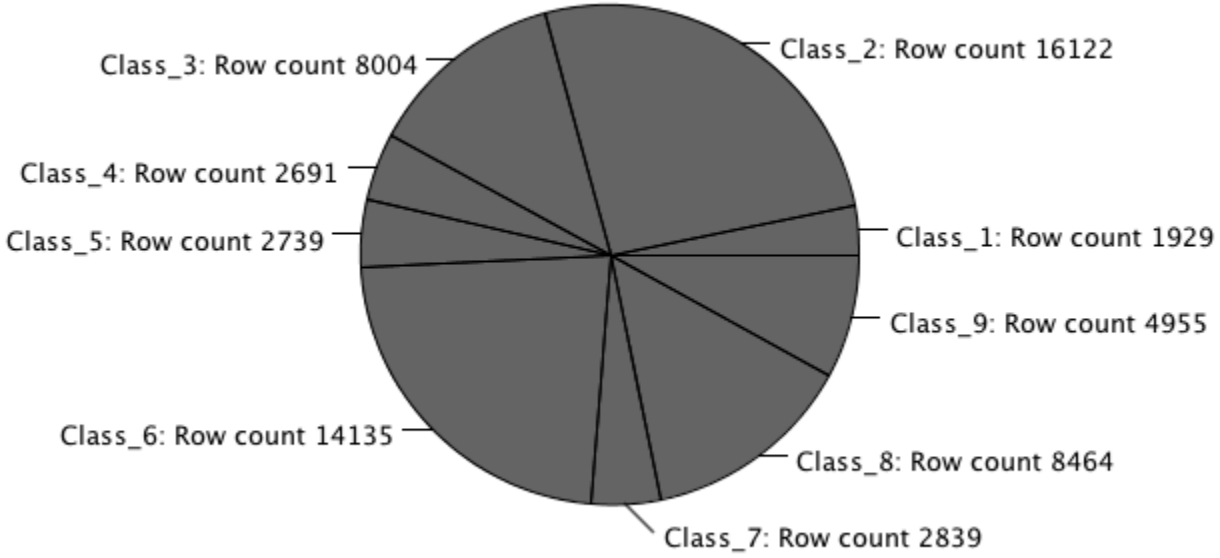
$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

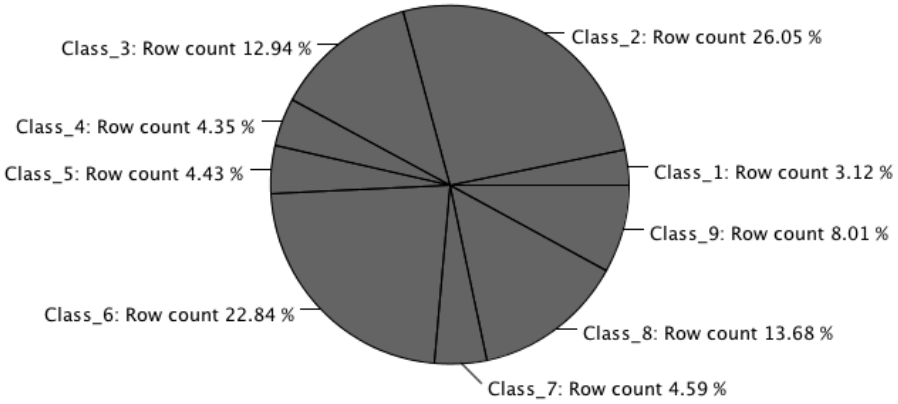
where N is the number of products in the test set, M is the number of class labels, log is the natural logarithm, y_{ij} is 1 if observation i is in class j and 0 otherwise, and p_{ij} is the predicted probability that observation i belongs to class j.

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

3. Descomposición de problemas multiclase. Binarización

Una aplicación real en KAGGLE de Problema Multiclase





3. Descomposición de problemas multiclase. Binarización

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.

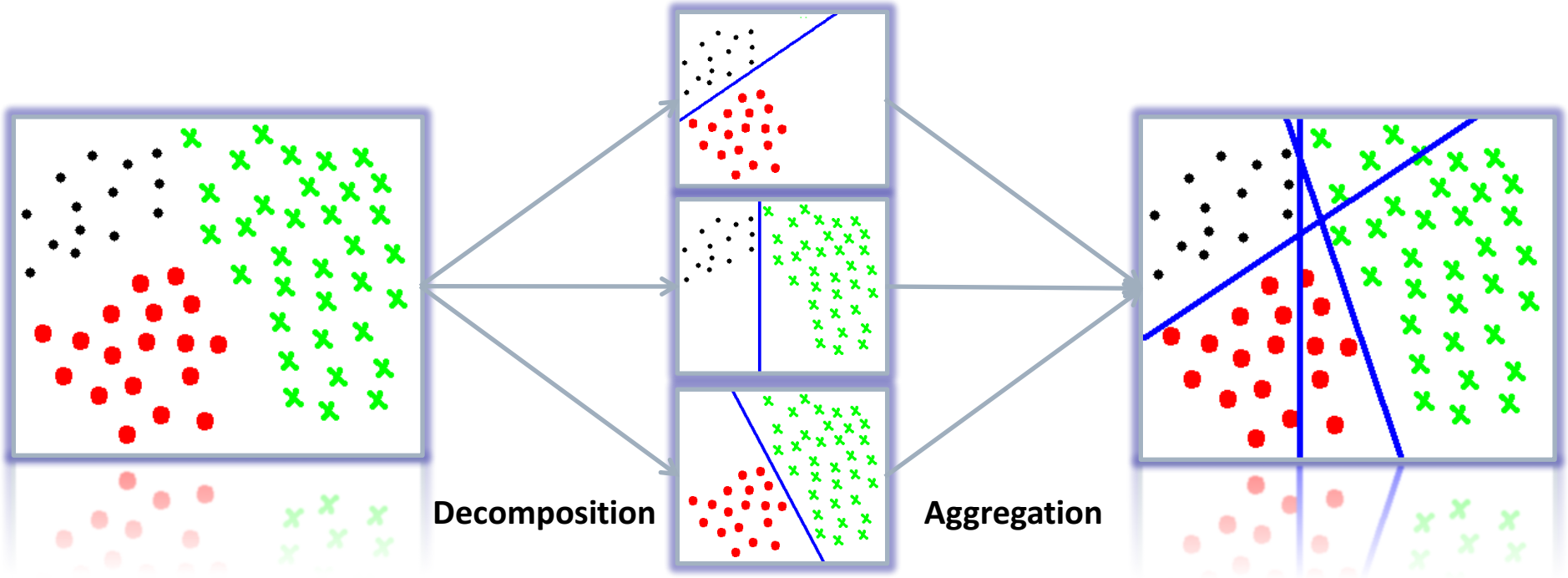
See someone using multiple accou
Let us kn

#	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)

3. Descomposición de problemas multiclase. Binarización

Estrategias de descomposición: "One-vs-One" (OVO)

- 1 problema binario para cada par de clases
  - Nombres en la literatura: Pairwise Learning, Round Robin, All-vs-All…
  - *Total = m(m-1) / 2 classificadores*



3. Descomposición de problemas multiclase. Binarización

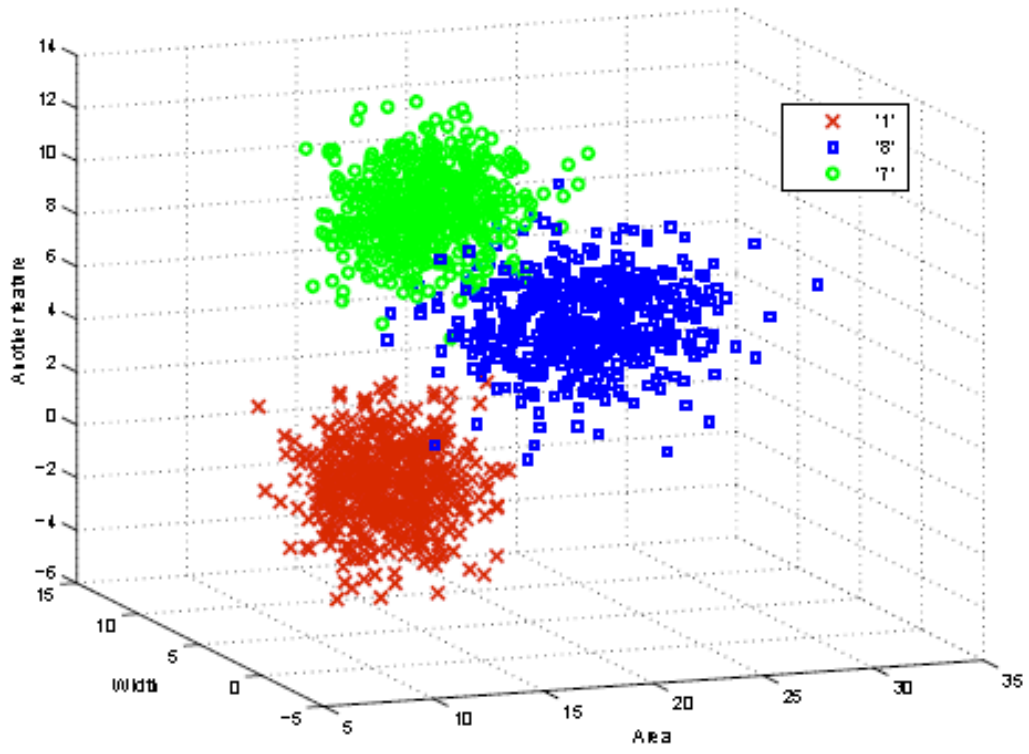


FIGURE : A multi-class problem, a new feature is needed

3. Descomposición de problemas multiclase. Binarización

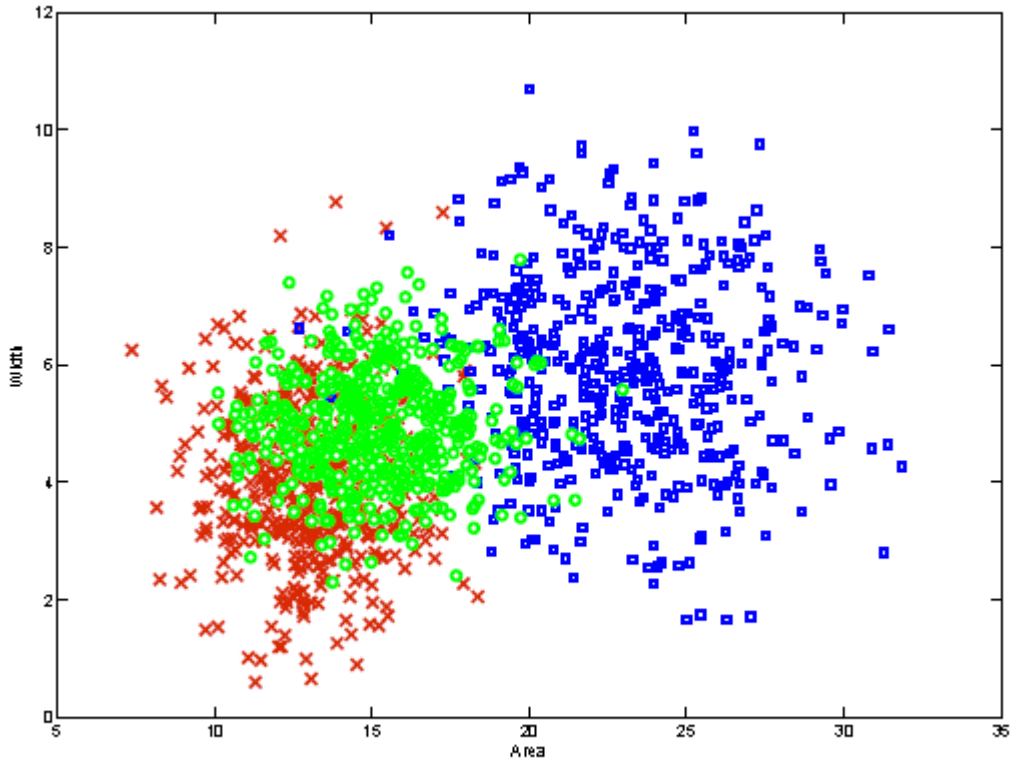
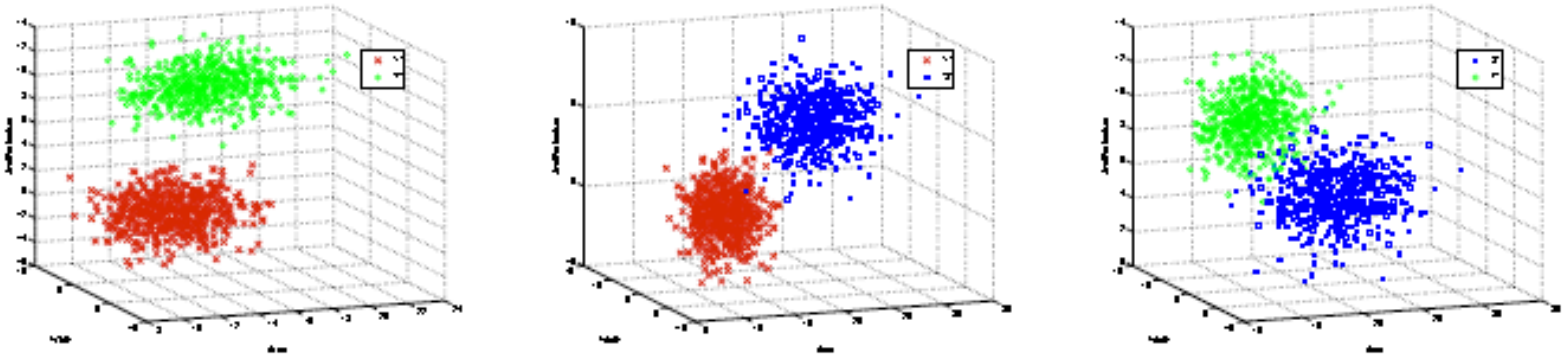


FIGURE : A multi-class problem

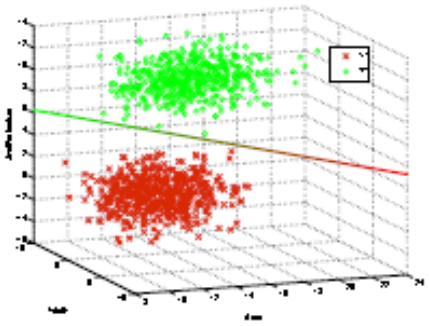
3. Descomposición de problemas multiclase. Binarización



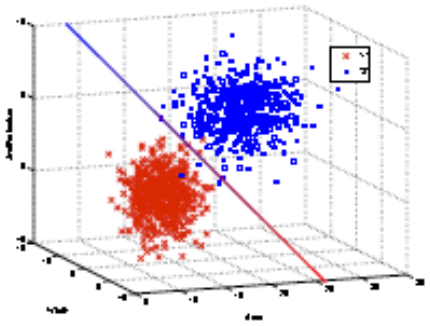
(a) '1' vs. '7' (b) '1' vs. '8' (c) '8' vs. '7'

FIGURE : One-vs-One scheme

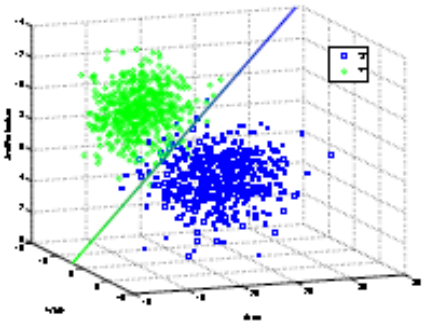
3. Descomposición de problemas multiclase. Binarización



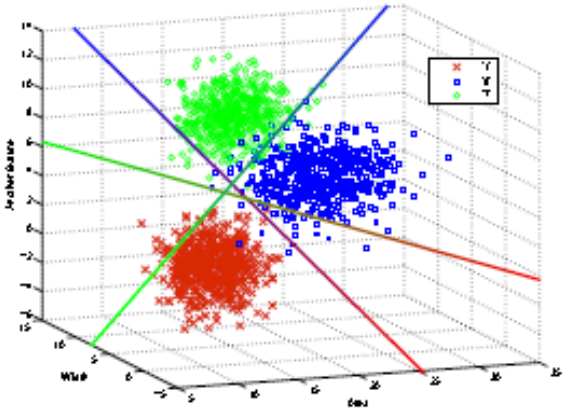
(a) '1' vs. '7'



(b) '1' vs. '8'



(c) '8' vs. '7'



(d) Aggregation

FIGURE : One-vs-One scheme

3. Descomposición de problemas multiclase. Binarización

Pairwise Learning: Combinación de las salidas

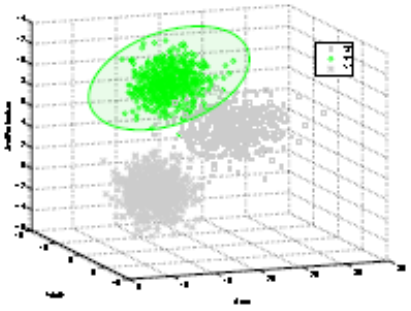
- Fase de agregación
  - *Se proponen diferentes vías para combinar los clasificadores base y obtener la salida final (clase).*
- Se comienza por una matriz de ratios de clasif.

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

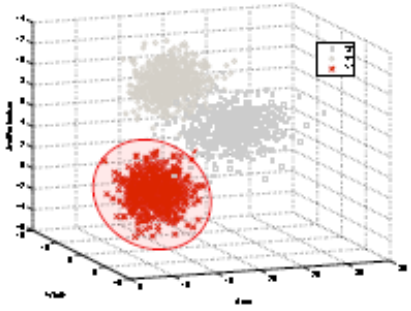
  - r_{ij} = confianza del clasificador en favor de la clase i
  - r_{ji} = confianza del clasificador en favor de la clase j
    - Usualmente: $r_{ji} = 1 - r_{ij}$
  - Se pueden plantear diferentes formas de combinar estos ratios.

3. Descomposición de problemas multiclase. Binarización

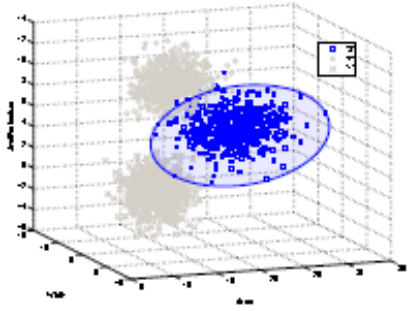
Otra estrategia de descomposición: One vs All (OVA)



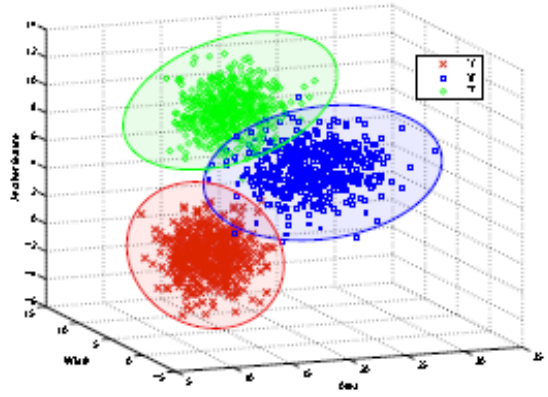
(a) '7' vs. '1' and '8'



(b) '1' vs. '7' and '8'



(c) '8' vs. '1' and '7'

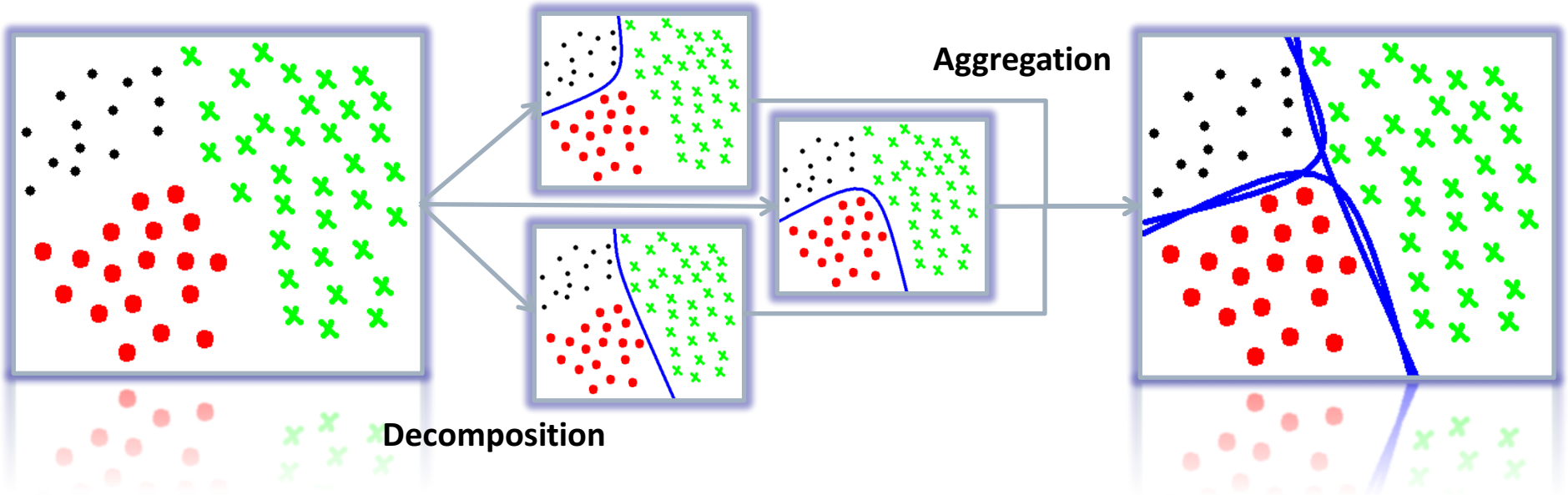


(d) Aggregation

FIGURE : One-vs-All scheme

3. Descomposición de problemas multiclase. Binarización

Otra estrategia de descomposición: One vs All (OVA)



3. Descomposición de problemas multiclase. Binarización

- 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.

3. Descomposición de problemas multiclase. Binarización

Ventajas:

- Clasificadores más pequeños (menor número de instancias)
- Fronteras de decisión más simples

Un algoritmo que está en el "estado del arte" en comportamiento y utiliza la estrategia OVO para múltiples clases: SVM (Support Vector Machine).

Estudios sobre la descomposición binaria:

M. Galar, A. Fernandez, 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

Anderson Rocha and Siome Goldenstein. Multiclass from Binary: Expanding One-vs-All, One-vs-One and ECOC-based Approaches. IEEE Transactions on Neural Networks and Learning Systems Vol. 25, Num. 2, pgs. 289–302, 2014 doi:10.1109/TNNLS.2013.2274735

Sistemas Inteligentes para la Gestión de la Empresa

TEMA 3. Análisis Predictivo para la Empresa

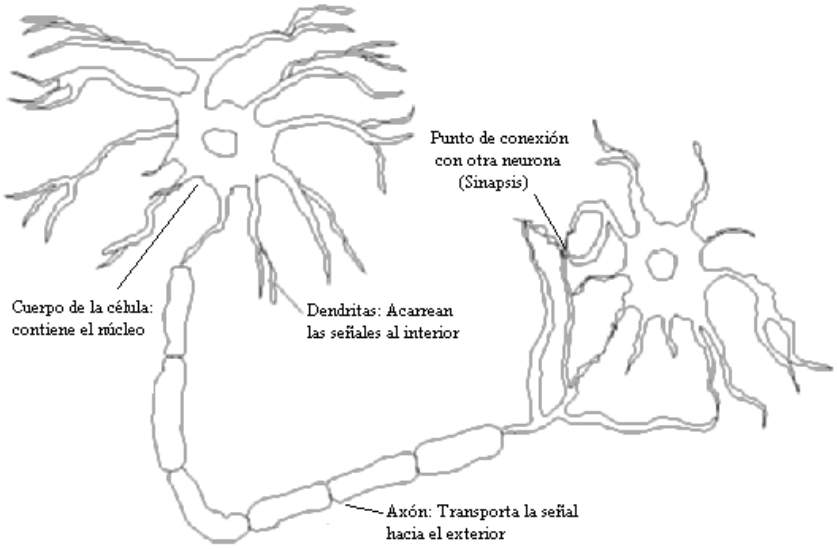
(Modelos predictivos avanzados de clasificación)

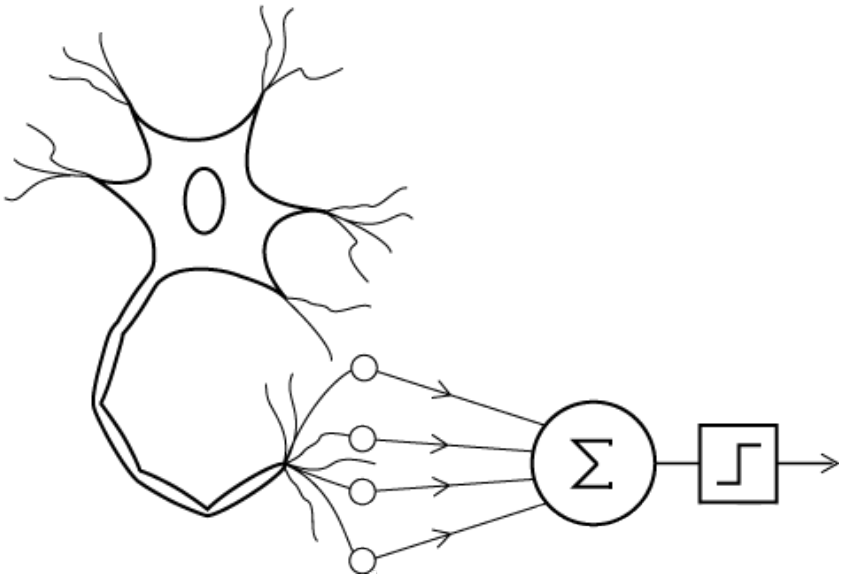
1. Clasificación no balanceada
2. Multiclasificadores: Bagging y Boosting
3. Multiples clases: Descomposición binaria
4. Redes Neuronales y Maquinas de soporte Vectorial

4. Clasificadores basados en redes neuronales

Redes neuronales

- Surgieron como un intento de emulación de los sistemas nerviosos biológicos
- Actualmente: computación modular distribuida mediante la interconexión de una serie de procesadores (neuronas) elementales



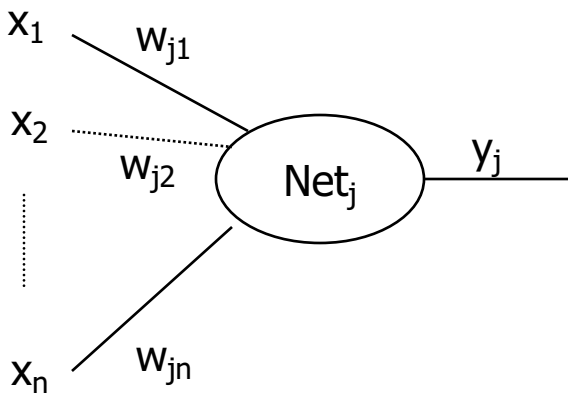


4. Clasificadores basados en redes neuronales

Redes neuronales

- Ventajas:
  - Habitualmente gran tasa de acierto en la predicción
  - Son más robustas que los árboles de decisión por los pesos
  - Robustez ante la presencia de errores (ruido, outliers,…)
  - Gran capacidad de salida: nominal, numérica, vectores,…
  - Eficiencia (rapidez) en la evaluación de nuevos casos
  - Mejoran su rendimiento mediante aprendizaje y éste puede continuar después de que se haya aplicado al conjunto de entrenamiento
- Desventajas:
  - Necesitan mucho tiempo para el entrenamiento
  - Entrenamiento: gran parte es ensayo y error
  - Poca (o ninguna) interpretabilidad del modelo (caja negra)
  - Difícil de incorporar conocimiento del dominio
  - Los atributos de entrada deben ser numéricos
  - Generar reglas a partir de redes neuronales no es inmediato
  - Pueden tener problemas de sobreaprendizaje
- ¿Cuándo utilizarlas?
  - Cuando la entrada tiene una dimensión alta
  - La salida es un valor discreto o real o un vector de valores
  - Posibilidad de datos con ruido
  - La forma de la función objetivo es desconocida
  - La interpretabilidad de los resultados no es importante

4. Clasificadores basados en redes neuronales



- Las señales que llegan a las dendritas se representan como $x_1, x_2, \ldots, x_n$
- Las conexiones sinápticas se representan por unos pesos w_{j1}, w_{j2}, w_{jn} que ponderan (multiplican) a las entradas. Si el peso entre las neuronas *j* e *i* es:
  a) positivo, representa una sinapsis excitadora
  b) negativo, representa una sinapsis inhibidora
  c) cero, no hay conexión

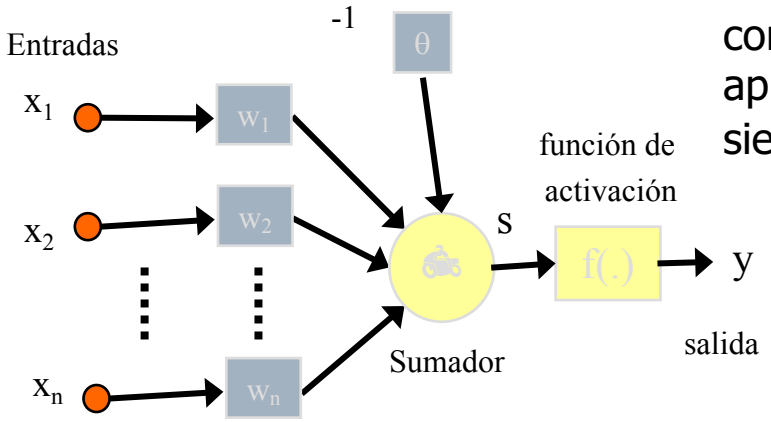
- La acción integradora del cuerpo celular (o actividad interna de cada célula) se presenta por

$$Net_j = w_{j1} \cdot x_1 + w_{j2} \cdot x_2 + \ldots + w_{jn} \cdot x_n = \sum_{i=1}^{n} w_{ji} \cdot x_i$$

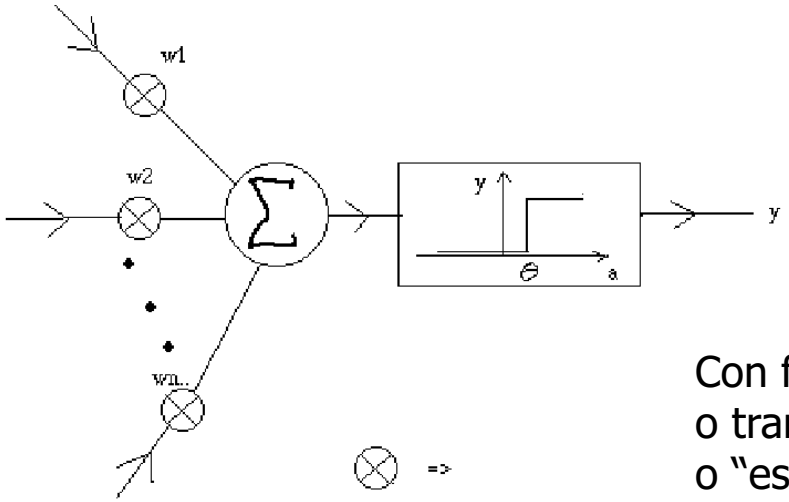
- La salida de la neurona se representa por y_j. Se obtiene mediante una función que, en general, se denomina función de salida, de transferencia o de activación. Esta función depende de Net_j y de un parámetro θ_j que representa el umbral de activación de la neurona

$$y_j = f(Net_j - \theta_j) = f\left(\sum w_{ji} \cdot x_i - \theta\right)$$

4. Clasificadores basados en redes neuronales



El umbral se puede interpretar como un peso sináptico que se aplica a una entrada que vale siempre -1



Con función de activación o transferencia tipo salto o "escalón"

4. Clasificadores basados en redes neuronales

- **Función de escalón**.
  Representa una neurona con sólo dos estados de activación: activada (1) y inhibida (0 ó -1)

$$y_j = H(Net_j - \theta_j) = \begin{cases} 1, & si\ Net_j >= \theta_j \\ -1, & si\ Net_j < \theta_j \end{cases}$$

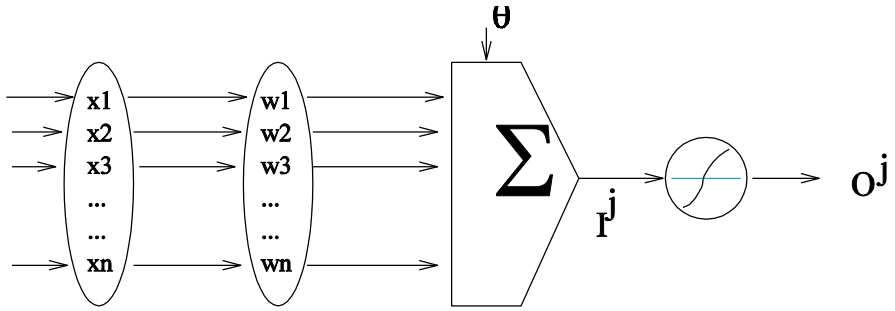
- **Función lineal**: $y_j = Net_j - \theta_j$

- **Función lineal a tramos**: $$y_j = \begin{cases} 1, & si\ Net_j >= \theta_j + a \\ Net_j - \theta_j, & si\ |Net_j - \theta_j| < a \\ -1, & si\ Net_j < \theta_j - a \end{cases}$$

- **Función sigmoidal**: $$y_j = \frac{1}{1+e^{-\lambda(Net_j-\theta_j)}} \qquad y_j = \frac{2}{1+e^{-\lambda(Net_j-\theta_j)}} - 1$$

- **Función base radial**: $$y_j = e^{-\left(\frac{Net_j-\theta_j}{\sigma}\right)^2}$$

4. Clasificadores basados en redes neuronales



- En definitiva, los componentes básicos de una red neuronal son:
  - Vector de pesos (w_{ij}), uno por cada entrada desde a la neurona *j*
  - Un sesgo (Θ_j) asociado a la neurona
  - Los datos de entrada $\{x_1,\ldots,x_n\}$ a una neurona N_j se hacen corresponder con un número real utilizando los componentes anteriores
  - Una función de activación f. Puede ser muy sencilla (función escalón) aunque normalmente es una función sigmoidal

4. Clasificadores basados en redes neuronales

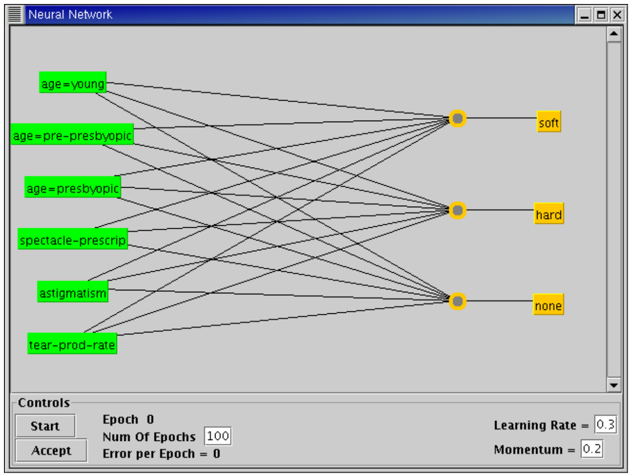
- Resolver un problema de clasificación con una red neuronal implica
  - Determinar (habitualmente con conocimiento experto)
    - el número de nodos de salida,
    - el número de nodos de entrada (y los atributos correspondientes,
    - el número de capas ocultas
  - Determinar pesos y funciones a utilizar
  - Para cada tupla en el conjunto de entrenamiento propagarlo en la red y evaluar la predicción de salida con el resultado actual.
    - Si la predicción es precisa, ajustar los pesos para asegurar que esa predicción tendrá un peso de salida más alto la siguiente vez
    - Si la predicción no es precisa, ajustar los pesos para que la siguiente ocasión se obtenga un valor menor para dicha clase
  - Para cada tupla en el conjunto de test propagarla por la red para realizar la clasificación

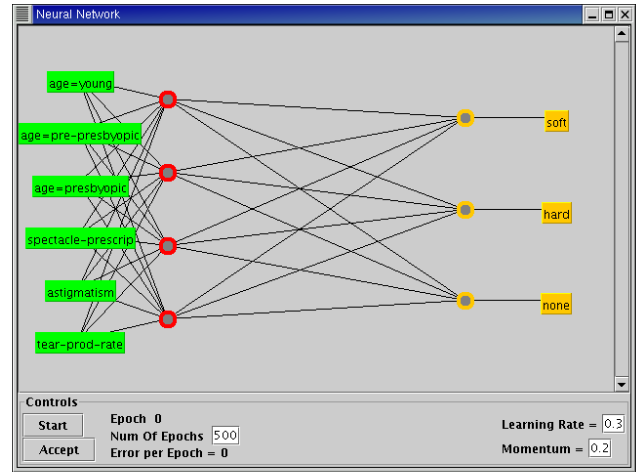
4. Clasificadores basados en redes neuronales

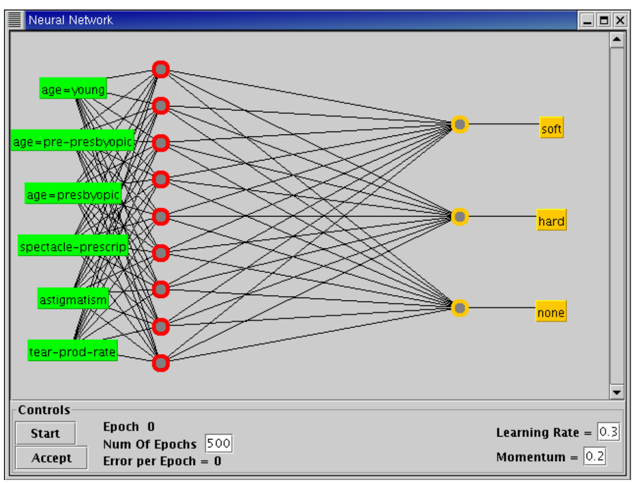
- El aprendizaje de la estructura de una red neuronal requiere experiencia, aunque existen algunas guías
- Entradas: Por cada variable numérica o binaria/booleana se pone una neurona de entrada. Por cada variable nominal (con más de dos estados) se pone una neurona de entrada por cada estado posible.
- Salidas: Si es para predicción se pone una única neurona de salida por cada valor de la variable clase
- Capas ocultas. Hay que indicar cuántas capas y cuantas neuronas hay que poner en cada capa. En Weka,
  - 0: No se pone capa oculta (sólo particiones linealesi)
  - Números enteros separados por comas: cada número representa la cantidad de neuronas de esa capa. P.e. 4,3,5: representa una red con tres capas ocultas de 4, 3 y 5 neuronas respectivamente
  - Algunos comodines:
    - i/o: neuronas en la entrada/salida
    - t: i+o
    - a: t/2 (es el valor por defecto)
- Cuando la red neuronal está entrenada, para clasificar una instancia, se introducen los valores de la misma que corresponden a las variables de los nodos de entrada.
  - La salida de cada nodo de salida indica la probabilidad de que la instancia pertenezca a esa clase
  - La instancia se asigna a la clase con mayor probabilidad

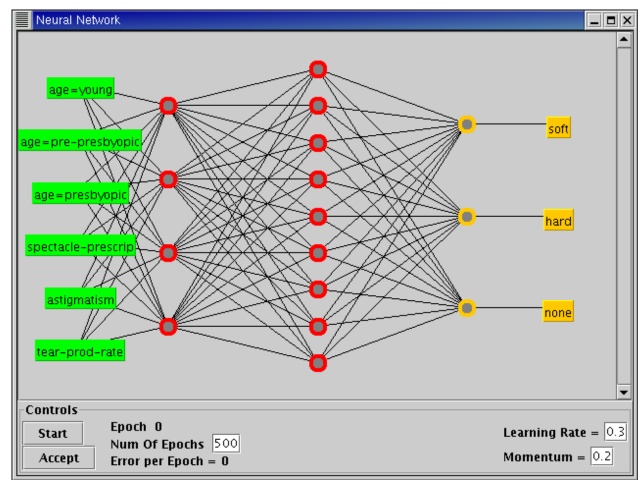
4. Clasificadores basados en redes neuronales

Clasificadores basados en Redes neruronales









4. Clasificadores basados en redes neuronales

- Todas las variables numéricas se normalizan [-1,1]

- Algoritmo de *Backprogation* o retropropagación:
  - Basado en la técnica del gradiente descendiente
  - No permite conexiones hacia atrás (retroalimentación) en la red

- Esquema del algoritmo de retropropagación
  1. Inicializar los pesos y sesgos aleatoriamente
  2. Para r=1 hasta número de *epoch* hacer
     a) Para cada ejemplo *e* de la BD hacer
     b) Lanzar un proceso *forward* de propagación en la red neuronal para obtener la salida asociada a *e* usando las expresiones vistas anteriormente
     c) Almacenar el valor O_j producido en cada neurona N_j
     d) Lanzar un proceso *backward* para recalcular los pesos y los sesgos asociados a cada neurona

  1 *epoch* = procesamiento de todos los ejemplos de la BD

4. Clasificadores basados en redes neuronales

- Fase de retropagación:
  - Para cada neurona de salida N_s hacer BP(N_s)
  - BP(N_j):
    1. Si N_j es una neurona de entrada, finalizar
    2. Si N_j es una neurona de salida
       entonces $Err_j = O_j \cdot (1-O_j) \cdot (T_j - O_j)$
       si no $Err_j = O_j(1-Oj) \Sigma_{k \text{ de salida}} Err_k \cdot w_{jk}$
       con T_j el valor predicho en la neurona N_j
    3. Actualizar los pesos $w_{ij} = w_{ij} + \alpha \cdot Err_j \cdot O_i$
    4. Actualizar los sesgos: $\Theta_j = \Theta_j + a \cdot Err_j$
    5. Para cada neurona N_i tal que $N_i \rightarrow N_j$ hacer BP(N_i)

4. Clasificadores basados en redes neuronales

En definitiva,

- Inicializar todos los pesos a números aleatorios pequeños
- Hasta que se verifique la condición de parada hacer
  - Para cada ejemplo de entrenamiento hacer
    - Introducir el ejemplo en la red y propagarla para obtener la salida (O_k)
    - Para cada unidad de salida k
      - $Error_k \leftarrow O_k(1-O_k)(t_k-O_k)$
    - Para cada unidad oculta
      - $Error_k \leftarrow O_k(1-O_k)\sum_j salida\ w_{k,j}\ error_j$
    - Modificar cada peso de la red
      - $w_{i,j} \leftarrow w_{i,j} + \Delta w_{ij}$
      - $\Delta w_{i,j} = \eta\ Error_j\ x_{i,j}$

4. Clasificadores basados en redes neuronales

- Existen muchos otros modelos de redes neuronales (recurrentes, memorias asociativas, redes neuronales de base radial, redes ART, …)

- El término de Deep Learning (procesamiento masivo de datos que utiliza redes neuronales) ha centrado la atención en modelos escalables de redes neuronales

- Google cuenta, desde hace tres años con el departamento conocido como ' **Google Brain'** (dirigido por Andrew Ng, Univ. Stanford, responsable de un curso de coursera de machine learning) dedicado a esta técnica entre otras. En 2013, Google adquirió la compañía DNNresearch Inc de uno de los padres del Deep Learning (Geoffrey Hinton). En enero de 2014 se hizo con el control de la 'startup' Deepmind Technologies una pequeña empresa londinense en la trabajaban que algunos de los mayores expertos en 'deep learning'.

4. Clasificadores basados en máquinas de soporte vectorial (SVM)

"Una **SVM(support vector machine)** es un modelo de aprendizaje que se fundamenta en la *Teoría de Aprendizaje Estadístico*. La idea básica es encontrar un hiperplano canónico que maximice el margen del conjunto de datos de entrenamiento, esto nos garantiza una buena capacidad de generalización."

Representación dual de un problema

+

Funciones Kernel

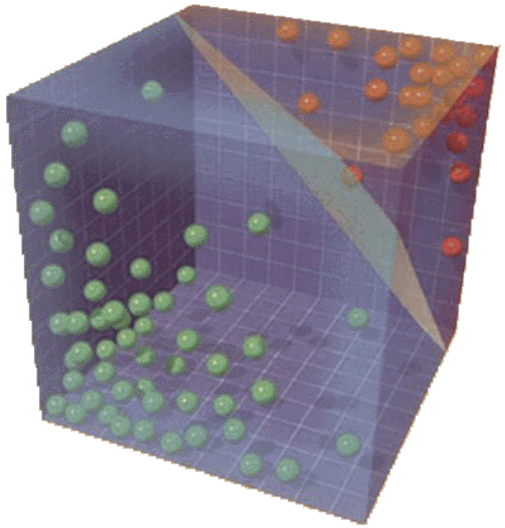
+

Teoría de Aprendizaje Estadística

+

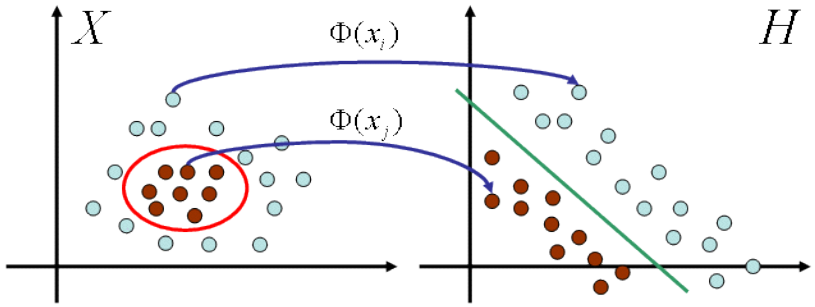
Teoría Optimización Lagrange

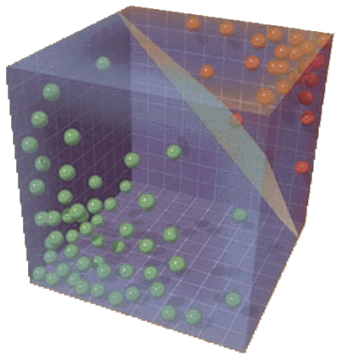
=



4. Clasificadores basados en máquinas de soporte vectorial (SVM)

- Una **SVM** es un **máquina de aprendizaje lineal** (requiere que los datos sean linealmente separables).
- Estos métodos explotan la información que proporciona el **producto interno (escalar)** entre los datos disponibles.
- La idea básica es:





4. Clasificadores basados en máquinas de soporte vectorial (SVM)

- **Problemas de esta aproximación:**
  - ¿Cómo encontrar la función ϕ?
  - El espacio de características inducido **H** es de **alta dimensión.**
    - **Problemas de cómputo y memoria.**

4. Clasificadores basados en máquinas de soporte vectorial (SVM)

- **Solución:**
  - Uso de funciones kernel.
  - **Función kernel =** producto interno de dos elementos en algún espacio de características inducido (**potencialmente de gran dimensionalidad**).
  - Si usamos una función kernel no hay necesidad de especificar la función ϕ**.**

4. Clasificadores basados en máquinas de soporte vectorial (SVM)

- Polinomial: $K(x,y) = \langle x, y \rangle^d$
- Gausiano: $K(x,y) = e^{-\|x-y\|^2/2\sigma}$
- Sigmoide: $K(x,y) = \tanh(\alpha\langle x, y \rangle + \beta)$

4. Clasificadores basados en máquinas de soporte vectorial (SVM)

- Para resolver el problema de optimización planteado se usa la **teoría de Lagrange**.

Minimizar

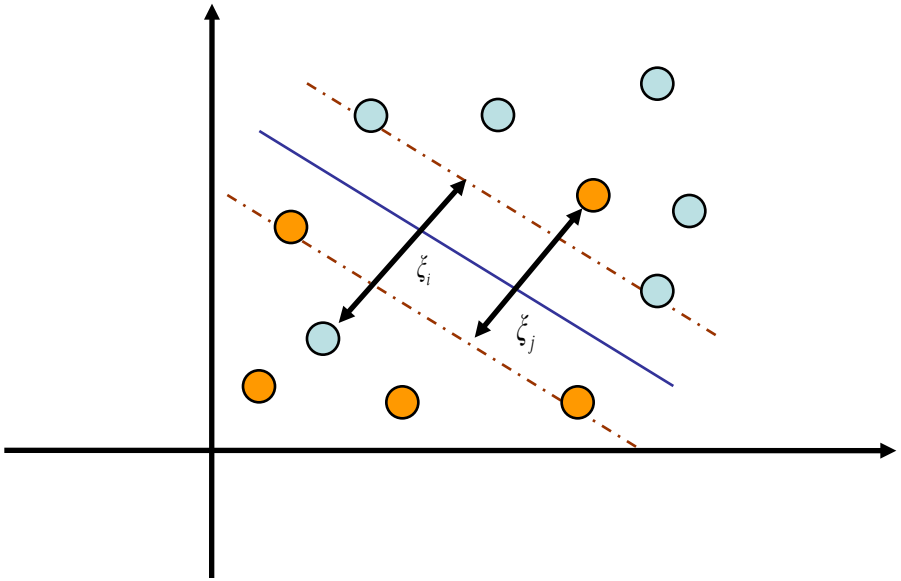
$$L(w,b) = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\langle w, x_i \rangle + b \right) - 1 \right]$$

Condicionado a

$$\alpha_i \geq 0$$

- Cada una de las variables α_i es un multiplicador de Lagrange y existe una variable por cada uno de los datos de entrada.

4. Clasificadores basados en máquinas de soporte vectorial (SVM)



$$y_i\left[\langle w, x_i\rangle + b\right] \geq 1 - \xi_i$$

4. Clasificadores basados en máquinas de soporte vectorial (SVM)

- Originariamente el **modelo de aprendizaje basado en SVMs** fue diseñado para **problemas de clasificación binaria**.

- La extensión a multi-clases se realiza mediante combinación de clasificadores binarios.

Sistemas Inteligentes para la Gestión de la Empresa

TEMA 3. Análisis Predictivo para la Empresa

(Modelos predictivos avanzados de clasificación)

1. Clasificación no balanceada
2. Multiclasificadores: Bagging y Boosting
3. Multiples clases: Descomposición binaria
4. Redes Neuronales y Maquinas de soporte Vectorial

Conclusiones

Bibliografía

V. Cherkassky, F.M. Mulier
Learning from Data: Concepts, Theory, and Methods (Sections 8 and 9)
2nd Edition, Wiley-IEE Prees, 2007

O. Shmueli, N.R. Patel, P.C. Bruce
Data mining for business intelligence (Part IV)
Wiley 2010 (2nd. edition)

Data Mining and Analysis: Fundamental Concepts and Algorithms (Part 4)
M. Zaki and W. Meira Jr.
Cambridge University Press, 2014.
http://www.dataminingbook.info/DokuWiki/doku.php

Conclusiones

Clasificación es el tipo de problema más estudiado en el ámbito de Minería de Datos.

Existen múltiples aproximaciones algorítmicas que presentan buenos resultados y que deben se consideradas para su uso en función de lo que queremos obtener en cuanto a modelos:

- ✓ sistemas basados en reglas (interpretables),
- ✓ prototipos de clasificación (algoritmos basados en instancias),
- ✓ algoritmos no interpretables cuya finalidad es la aproximación (redes neuronales, SVM ...)
- ✓ la combinación de clasificadores

Clasificación: Conclusiones

Existen muchos tipos concretos de problemas de clasificación que están siendo objeto de estudio en la actualidad, que dependen de la tipología de los datos.

- Multi-label Classification (MLC)
- Multi-Instance Learning (MIL)
- Semi-supervised Learning (SSL)
- Monotonic Classification
- Label Ranking

Adicionalmente existen muchos otros estudios asociados a la calidad de los datos (data complexity), escalabilidad,…

Sistemas Inteligentes para la Gestión de la Empresa

2016 - 2017

- Tema 1. Introducción a la Ciencia de Datos
- Tema 2. Depuración y Calidad de Datos. Preprocesamiento de datos
- Tema 3. Análisis Predictivo para la Empresa
- Tema 5. Análisis de Transacciones y Mercados
- Tema 4. Modelos avanzados de Analítica de Empresa
- Tema 6. Big Data
- Tema 7. Aplicaciones de la Ciencia de Datos en la Empresa