

2013 International Summer School on Trends in Computing

Tarragona, Spain, July 22-26, 2013

Organised by

Rovira i Virgili University



Imbalanced Classification: Common Approaches and Open Problems

Francisco Herrera

Research Group on Soft Computing and
Information Intelligent Systems (SCI²S)

<http://sci2s.ugr.es>

Dept. of Computer Science and A.I.
University of Granada, Spain

Email: herrera@decsai.ugr.es

<http://decsai.ugr.es/~herrera>



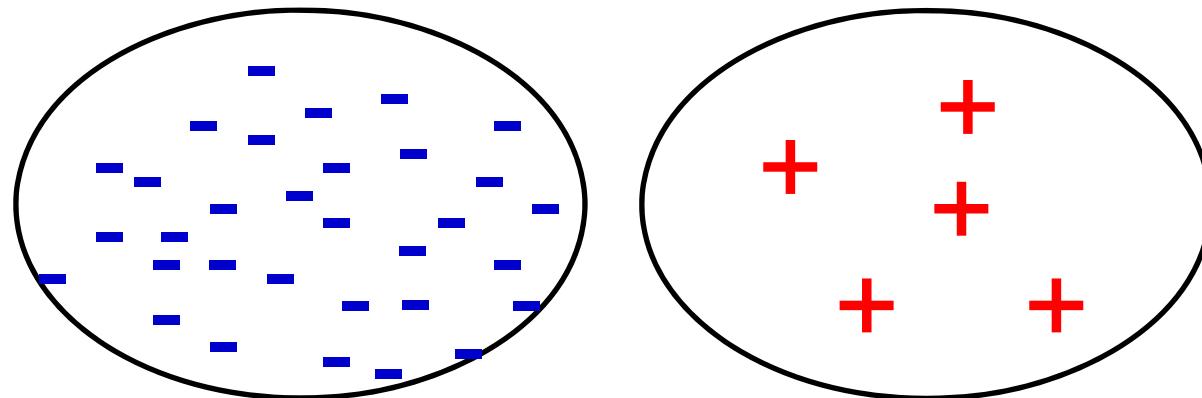
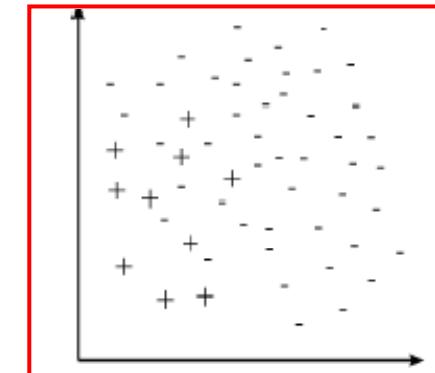
DECSAI
Universidad de Granada



Classification with Imbalanced Data Sets

Presentation

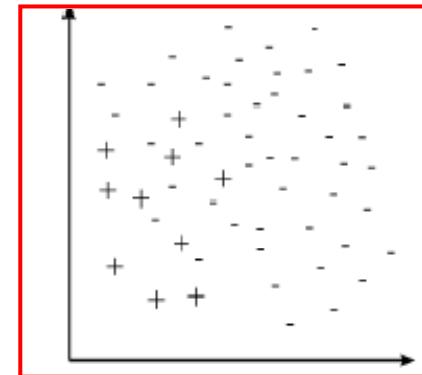
In a concept-learning problem, the data set is said to present a class imbalance if it contains many more examples of one class than the other.



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.



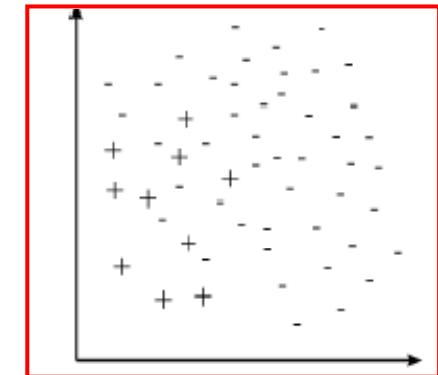
There exist many domains that do not have a balanced data set. There are a lot of problems where the most important knowledge usually resides in the minority class.

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 poses challenges for typical classifiers such as decision tree induction 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.

This course introduce the “classification with imbalanced data sets” analyzing in depth the problems and their solutions.

Contents

SESSION 1:

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning

Contents

SESSION 2:

V. 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

Contents

SESSION 3:

VI. Ensembles to address class imbalance

VII. Multiple class imbalanced data-sets: A pairwise learning approach

VIII. Some learning algorithms for imbalanced data sets

IX. Imbalanced Big Data

X. Class imbalance: Data sets, implementations, ...

XI. Class imbalance: Trends and final comments

Contents

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance
- VII. Multiple class imbalanced data-sets: A pairwise learning approach
- VIII. Imbalanced Big Data
- IX. Class imbalance: Data sets, implementations, ...
- X. Class imbalance: Trends and final comments

SESSION 1

SESSION 2

SESSION 3

Contents

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance
- VII. Multiple class imbalanced data-sets: A pairwise learning approach
- VIII. Imbalanced Big Data
- IX. Class imbalance: Data sets, implementations, ...
- X. Class imbalance: Trends and final comments

SESSION 1

SESSION 2

SESSION 3

Introduction to Imbalanced Data Sets

Imbalance Data Sets: Problem

Some recent applications

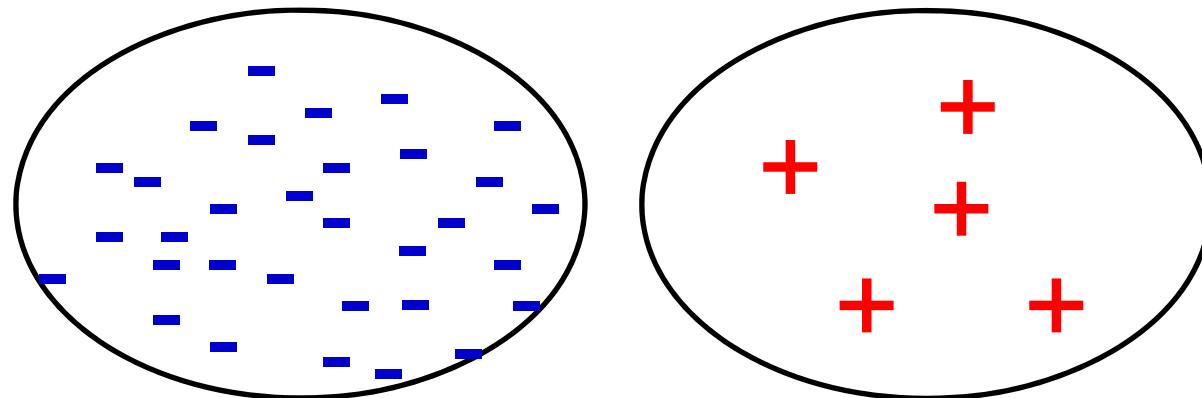
How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

Introduction to Imbalanced Data Sets

Data sets are said to be balanced if there are, approximately, as many positive examples of the concept as there are negative ones.

The positive examples are more interesting or their misclassification has a higher associate cost.



Introduction to Imbalanced Data Sets

Problem:

- The problem with class imbalances is that standard learners are often biased towards the majority class.
- That is because these classifiers attempt to reduce global quantities such as the error rate, not taking the data distribution into consideration.

Result:

- ✓ examples from the overwhelming class are well-classified
- ✓ whereas examples from the minority class tend to be misclassified.
- ✓ As a result, these classifiers tend to ignore small classes while concentrating on classifying the large ones accurately.

Introduction to Imbalanced Data Sets

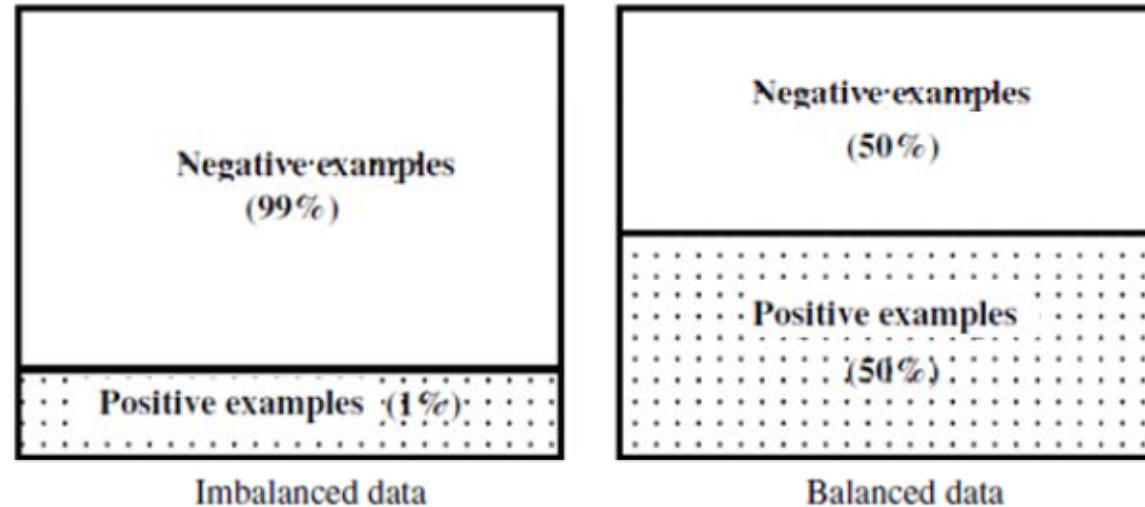


Fig. 1. Imbalanced and balanced data sets.

biased towards the majority class

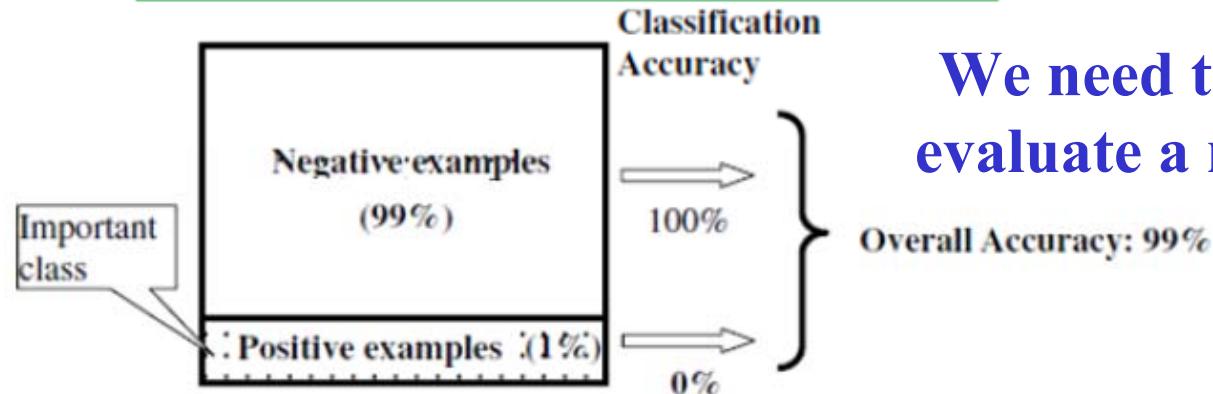


Fig. 2. The illustration of class imbalance problems.

Introduction to Imbalanced Data Sets

Imbalance Data Sets: Problem

Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

Introduction to Imbalanced Data Sets

Some recent applications

Sun, A., Lim, E.-P., Liu, Y. On strategies for imbalanced **text classification** using SVM: A comparative study (2009) *Decision Support Systems*, 48 (1), pp. 191-201.

Tsai, C.-h., Chang, L.-c., Chiang, H.-c. **Forecasting of ozone episode days** by cost-sensitive neural network methods (2009) *Science of the Total Environment*, 407 (6), pp. 2124-2135.

W.-Z. Lu, D. Wang, **Ground-level ozone prediction** by support vector machine approach with a cost-sensitive classification scheme, *Science of the Total Environment* 395 (2–3) (2008) 109–116

Y.-H. Liu, Y.-T. Chen, **Face recognition** using total margin-based adaptive fuzzy support vector machines, *IEEE Transactions on Neural Networks* 18 (1) (2007) 178–192.

L. Xu, M.Y. Chow, L.S. Taylor, **Power distribution fault cause identification** with imbalanced data using the data mining-based fuzzy classification E-Algorithm, *IEEE Transactions on Power Systems* 22 (1) (2007) 164–171.

Y. M. Huang, C. M. Hung, and H. C. Jiau, “Evaluation of neural networks and data mining methods on a **credit assessment task** for class imbalance problem,” *Nonlinear Anal. Real World Applicat.*, vol. 7, no. 4, pp. 720–747, 2006.

Introduction to Imbalanced Data Sets

Some recent applications

Vilariño, F., Spyridonos, P., Deiorio, F., Vitria, J., Azpiroz, F., Radeva, P.

Intestinal motility assessment with video capsule endoscopy: Automatic annotation of phasic intestinal contractions (2010) IEEE Transactions on Medical Imaging, 29 (2), art. no. 4909037, pp. 246-259.

Tek, F.B., Dempster, A.G., Kale, I. **Parasite detection and identification for automated thin blood film malaria diagnosis (2010) Computer Vision and Image Understanding, 114 (1), pp. 21-32.**

D. Williams, V. Myers, M. Silvious, **Mine classification with imbalanced data, IEEE Geoscience and Remote Sensing Letters 6 (3) (2009) 528–532.**

Yang, P., Xu, L., Zhou, B.B., Zhang, Z., Zomaya, A.Y. **A particle swarm based hybrid system for imbalanced medical data sampling (2009) BMC Genomics, 10 (SUPPL. 3), art. no. S34.**

Taft, L.M., Evans, R.S., Shyu, C.R., Egger, M.J., Chawla, N., Mitchell, J.A., Thornton, S.N., Bray, B., Varner, M. **Countering imbalanced datasets to improve adverse drug event predictive models in labor and delivery (2009) Journal of Biomedical Informatics, 42 (2), pp. 356-364.**

Introduction to Imbalanced Data Sets

Imbalance Data Sets: Problem

Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

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)

Classical evaluation:

$$\text{Error Rate: } (FP + FN)/N$$

$$\text{Accuracy Rate: } (TP + TN) / N$$

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

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**

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

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

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

Precision = $TP / (TP + FP)$

Recall = $TP / (TP + FN)$

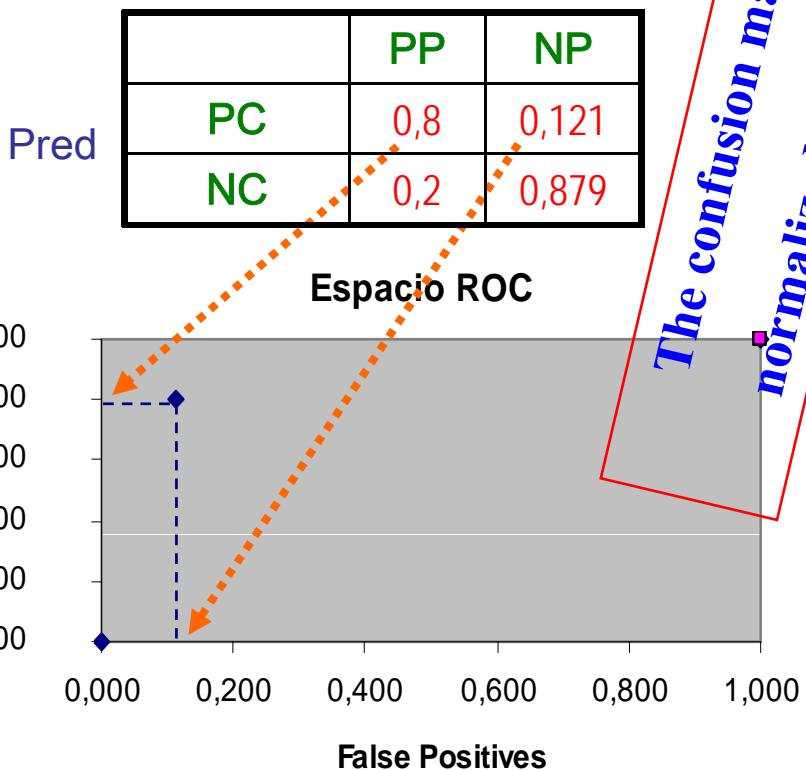
F-measure: $(2 \times \text{precision} \times \text{recall}) / (\text{recall} + \text{precision})$

Introduction to Imbalanced Data Sets

“*Receiver-Operator Characteristics*” – used by mathematicians to analyse radar data. Applied in signal detection to show tradeoff between **hit rate** and **false alarm rate** over noisy channel.

A ROC curve displays a relation between sensitivity and specificity for a given classifier (binary problems, parameterized classifier or a score classification)

Sensitivity



1 - Specificity

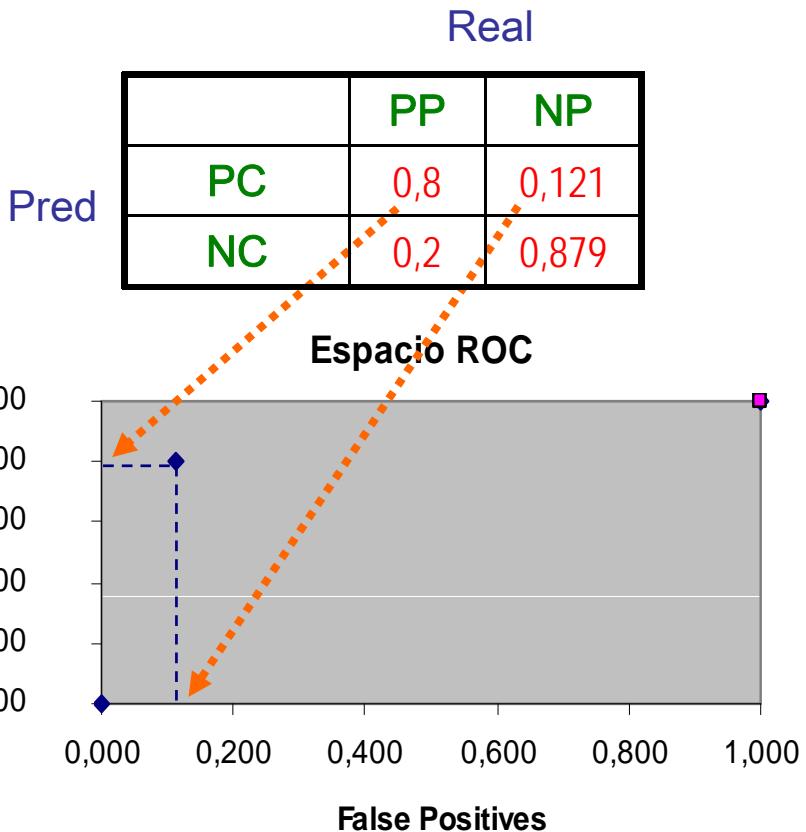
Introduction to Imbalanced Data Sets

ROC Curves (It is a two-dimensional graph to depicts trade-offs between benefits (true positives) and costs (false positives)).

The confusion matrix is normalized by columns

A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition 30(7) (1997) 1145-1159.

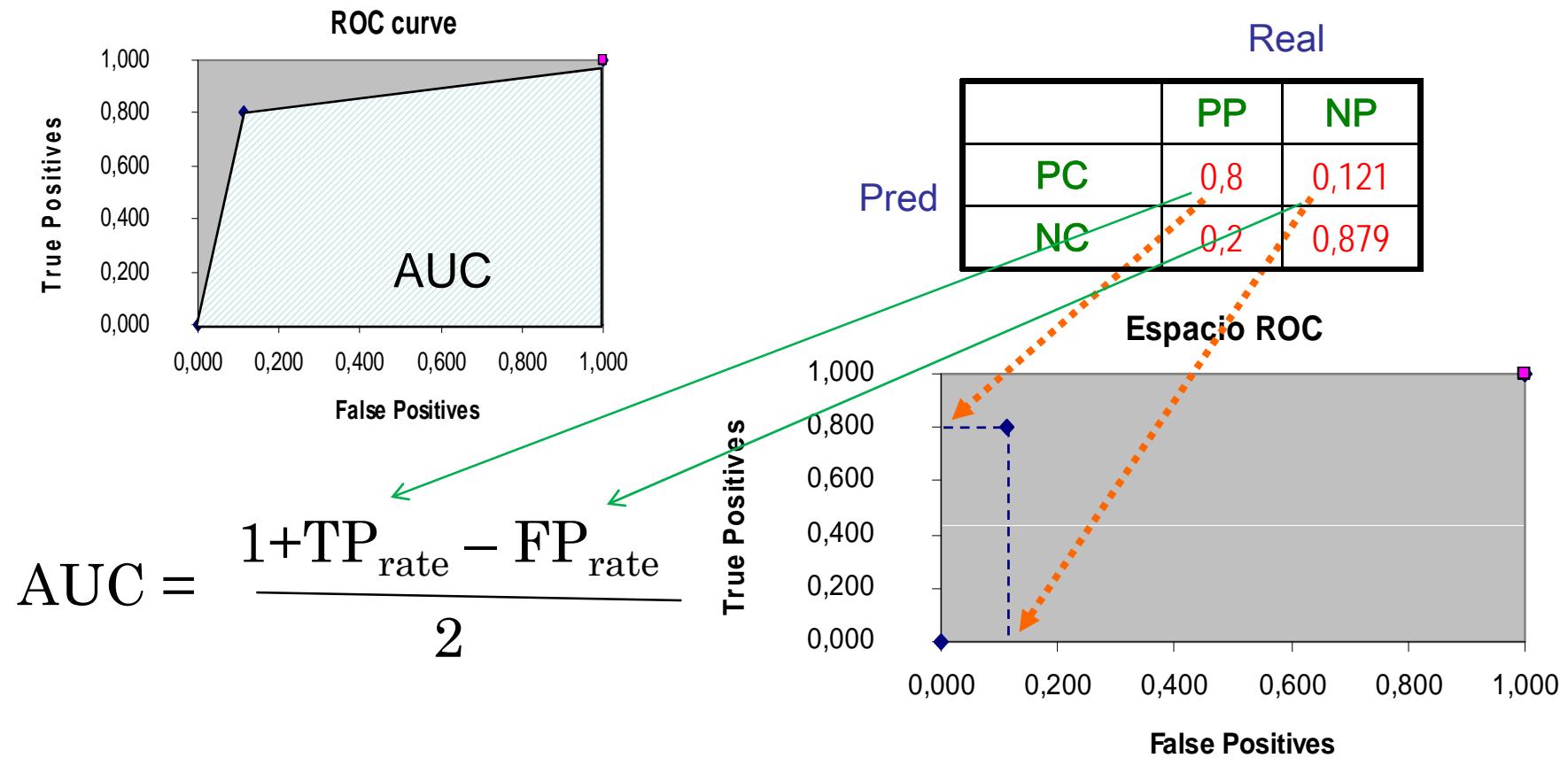
Sensitivity



1 - Specificity

Introduction to Imbalanced Data Sets

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



Introduction to Imbalanced Data Sets

Imbalance Data Sets: Problem

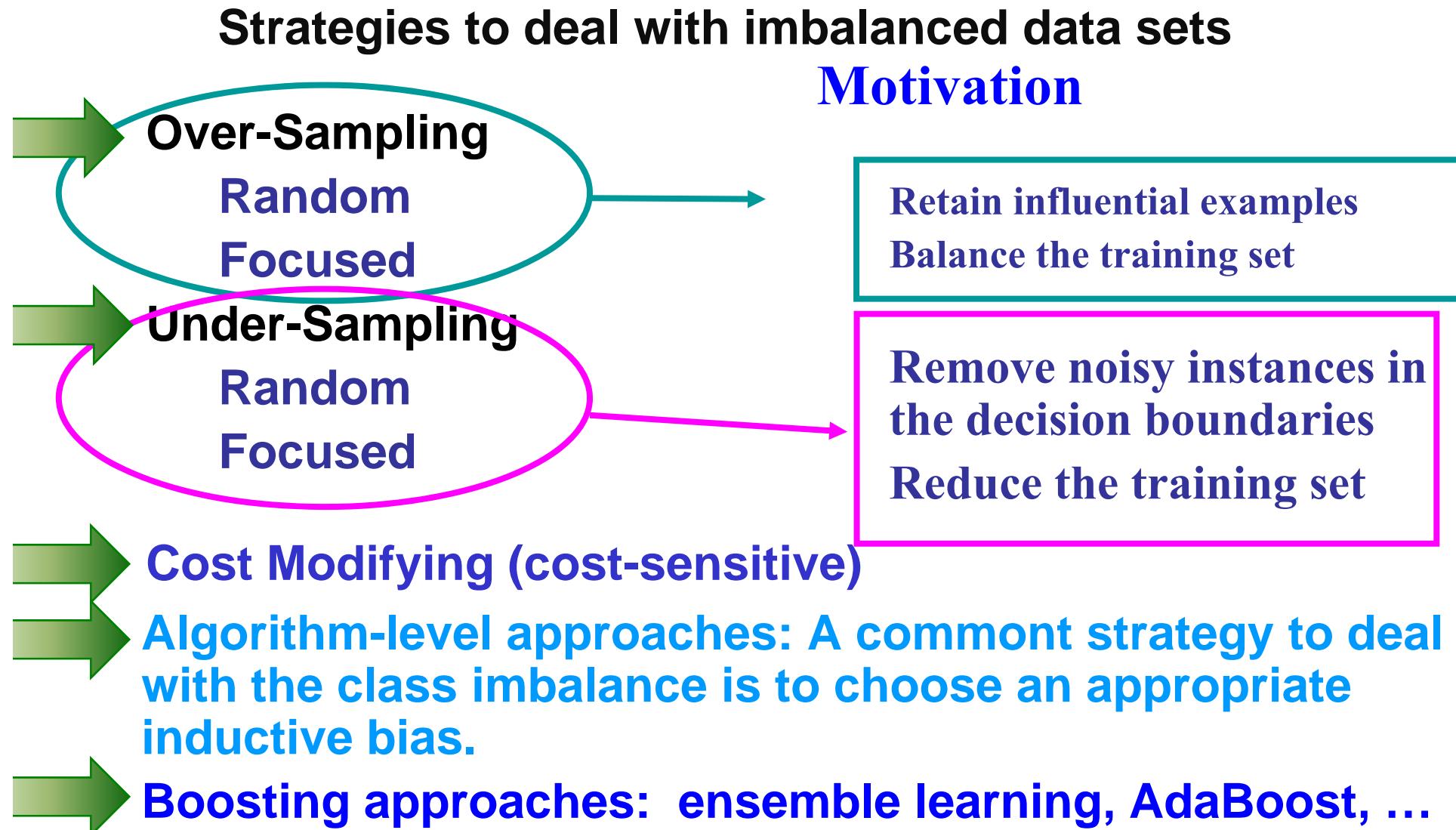
Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

Introduction to Imbalanced Data Sets

Data level vs Algorithm Level



Contents

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance
- VII. Multiple class imbalanced data-sets: A pairwise learning approach
- VIII. Some learning algorithms for imbalanced data sets
- IX. Imbalanced Big Data
- X. Class imbalance: Data sets, implementations, ...
- XI. Class imbalance: Trends and final comments

SESSION 1

SESSION 2

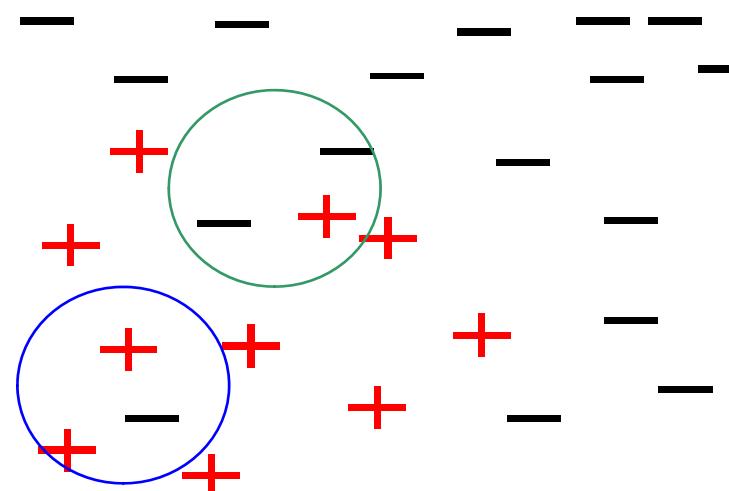
SESSION 3

Resampling the original data sets

Undersampling vs oversampling

Models for undersampling

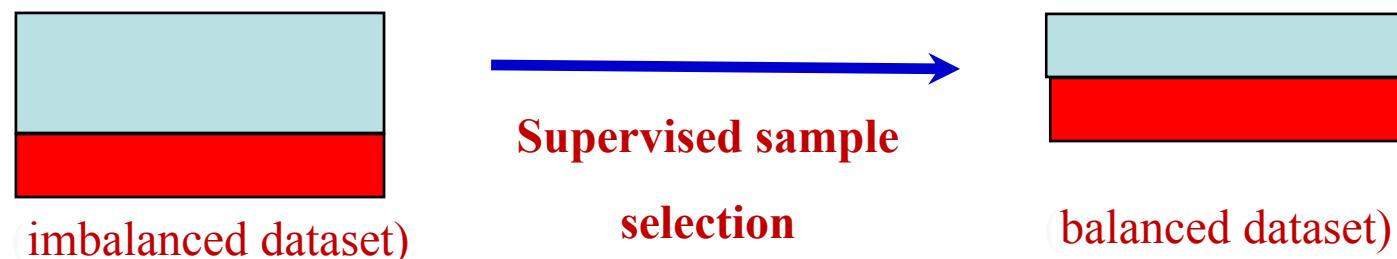
Oversampling. SMOTE (state of the art): Problems, hybridizations, analysis



Resampling the original data sets

Resampling is the process of manipulating the distribution of the training examples in an effort to improve the performance of classifiers.

There is no guarantee that the training examples occur in their optimal distribution in practical problems, and thus, the idea of resampling is “to add or remove examples with the hope of reaching the optimal distribution of the training examples” and thus, realizing the potential ability of classifiers.



Resampling the original data sets

Undersampling vs oversampling

examples -



examples +

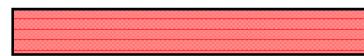


under-sampling

examples -



examples +



over-sampling

examples -



examples +



Resampling the original data sets

Undersampling vs oversampling

Over Sampling

Random

Focused

Under Sampling

Random

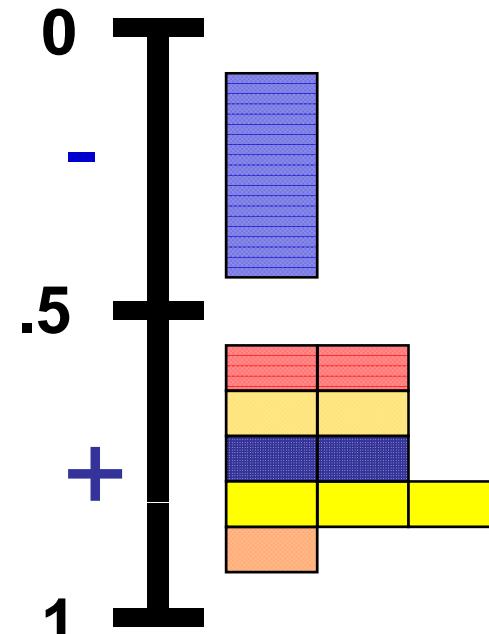
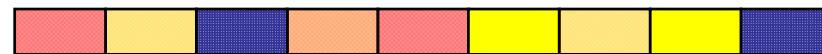
Focused

Cost Modifying

examples of -



examples of +



Resampling the original data sets

Undersampling vs oversampling

Over Sampling

Random

Focused

Under Sampling

Random

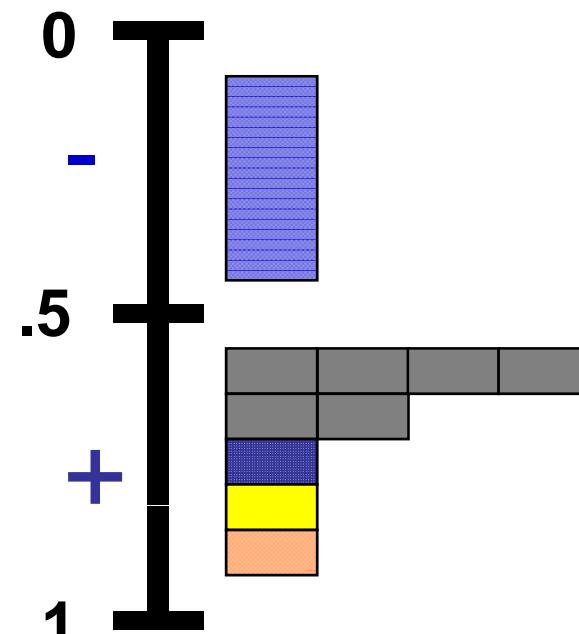
Focused

Cost Modifying

examples of -



examples of +



Resampling the original data sets

Undersampling vs oversampling

Over Sampling

Random

Focused

Under Sampling

Random

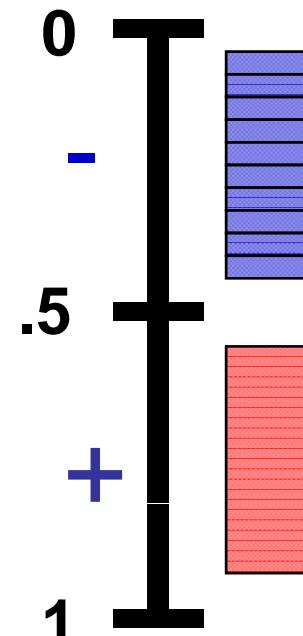
Focused

Cost Modifying

examples of -



examples of +



Resampling the original data sets

Undersampling vs oversampling

Over Sampling

Random

Focused

Under Sampling

Random

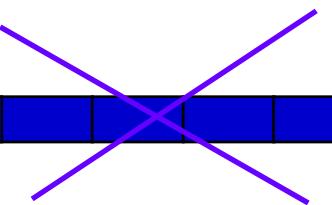
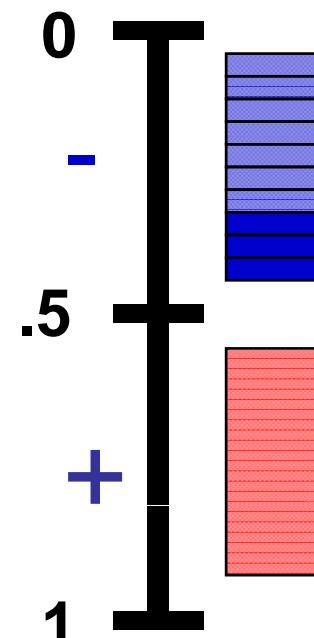
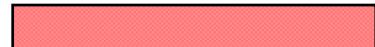
Focused

Cost Modifying

examples of -



examples of +

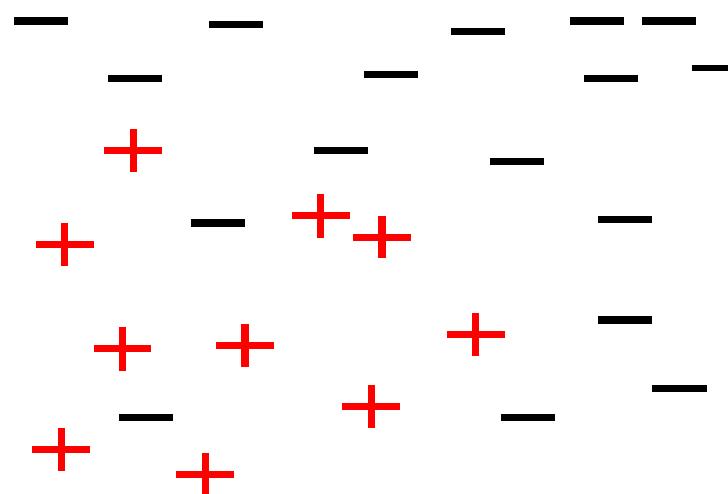


Resampling the original data sets

Undersampling vs oversampling

Models for undersampling

Oversampling. SMOTE (state of the art): Problems, hybridizations, analysis

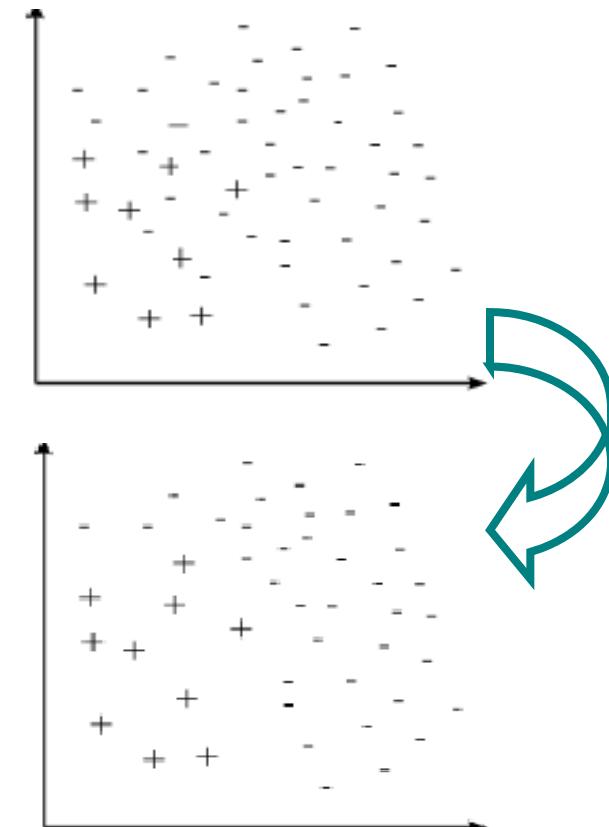


Resampling the original data sets

Under-sampling

Tomek Links

- To remove both noise and borderline examples of the majority class
- Tomek link
 - E_i, E_j belong to different classes,
 $d(E_i, E_j)$ is the distance between them.
 - A (E_i, E_j) pair is called a Tomek link if there is no example E_l , such that $d(E_i, E_l) < d(E_i, E_j)$ or $d(E_j, E_l) < d(E_i, E_j)$.

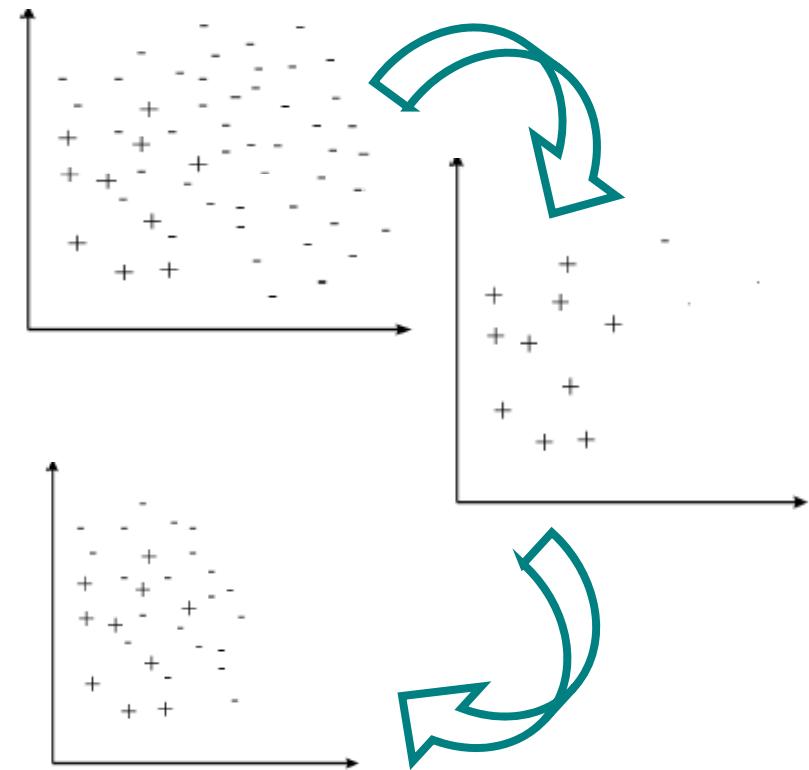


Resampling the original data sets

Under-sampling

CNN

- To remove both noise and borderline examples
- Algorithm:
 - Let E be the original training set
 - Let E' contains all positive examples from S and one randomly selected negative example
 - Classify E with the 1-NN rule using the examples in E'
 - Move all misclassified example from E to E'



Resampling the original data sets

Under-sampling

OSS, CNN+TL, NCL

- One-sided selection
 - Tomek links + CNN
- CNN + Tomek links
 - Proposed by the author
 - Finding Tomek links is computationally demanding, it would be computationally cheaper if it was performed on a reduced data set.

• NCL

To remove majority class examples
Different from OSS, emphasize more data cleaning than data reduction

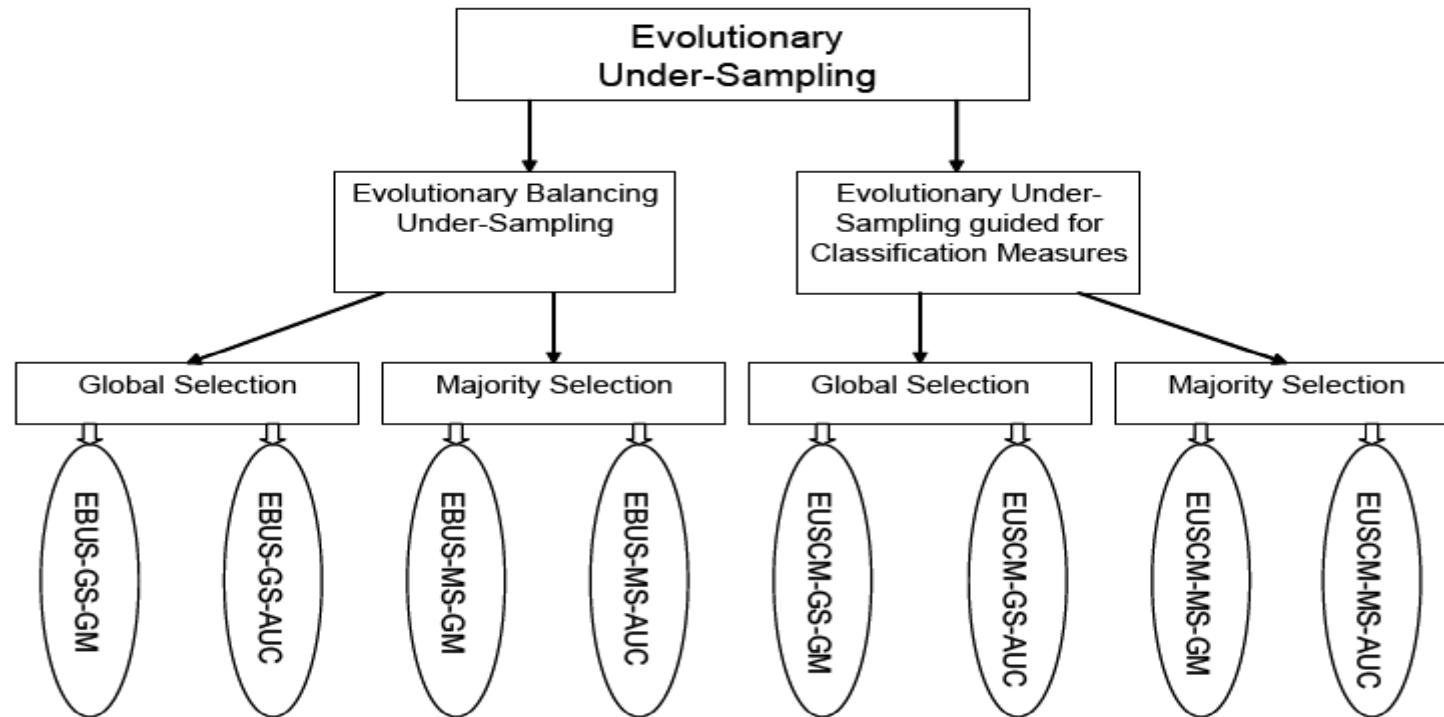
Algorithm:

- Find three nearest neighbors for each example E_i in the training set
- If E_i belongs to majority class, & the three nearest neighbors classify it to be minority class, then remove E_i
- If E_i belongs to minority class, and the three nearest neighbors classify it to be majority class, then remove the three nearest neighbors

Resampling the original data sets

Under-sampling

Evolutionary Algorithms

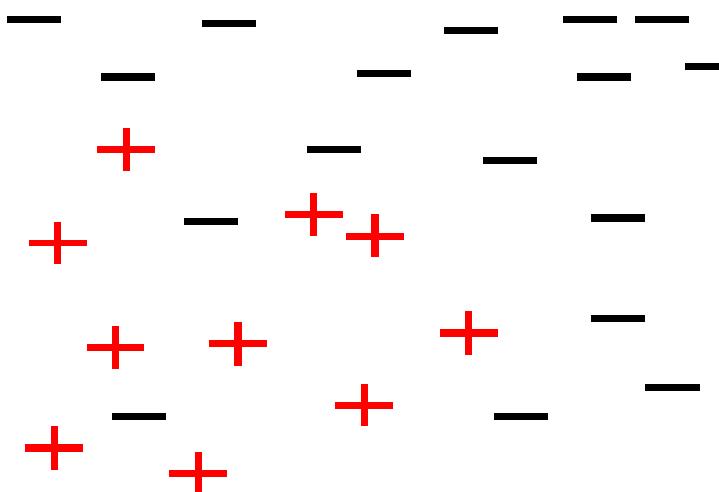


Resampling the original data sets

Undersampling vs oversampling

Models for undersampling

Oversampling. SMOTE (state of the art): Problems, hybridizations, analysis



Resampling the original data sets

Oversampling: Replicating examples

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

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

Resampling the original data sets

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

To form new minority class examples by interpolating between several minority class examples that lie together.

- in ``feature space" rather than ``data space"
- **Algorithm:** For each minority class example, introduce synthetic examples along the line segments joining any/all of the k minority class nearest neighbors.
- **For each minority Sample**
 - Find its k-nearest minority neighbours
 - Randomly select j of these neighbours
 - Randomly generate synthetic samples along the lines joining the minority sample and its j selected neighbours
(j depends on the amount of oversampling desired)

Resampling the original data sets

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

Note: Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen.

For example: if we are using 5 nearest neighbors, if the amount of over-sampling needed is 200%, only two neighbors from the five nearest neighbors are chosen and one sample is generated in the direction of each.

Resampling the original data sets

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

- Synthetic samples are generated in the following way:
 - Take the difference between the feature vector (sample) under consideration and its nearest neighbor.
 - Multiply this difference by a random number between 0 and 1
 - Add it to the feature vector under consideration.

Consider a sample (6,4) and let (4,3) be its nearest neighbor.

(6,4) is the sample for which k-nearest neighbors are being identified

(4,3) is one of its k-nearest neighbors.

Let:

$$f1_1 = 6 \quad f2_1 = 4 \quad f2_1 - f1_1 = -2$$

$$f1_2 = 4 \quad f2_2 = 3 \quad f2_2 - f1_2 = -1$$

The new samples will be generated as

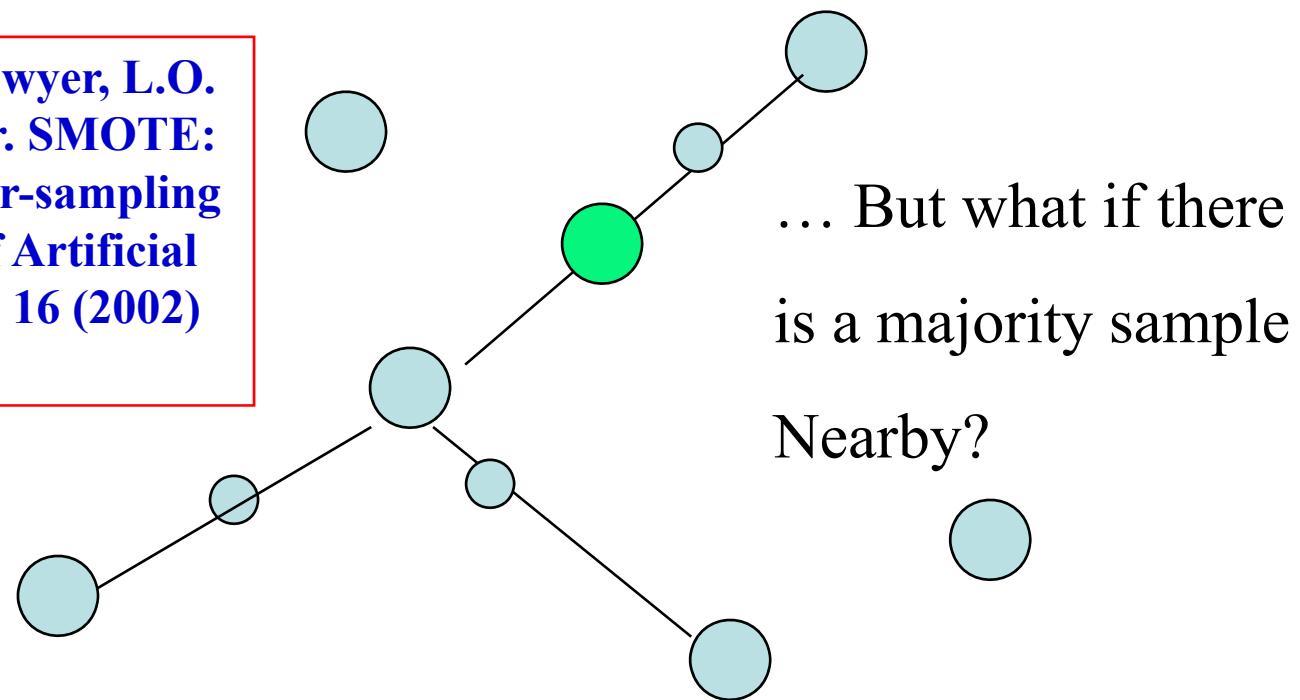
$$(f1', f2') = (6,4) + \text{rand}(0-1) * (-2,-1)$$

`rand(0-1)` generates a random number between 0 and 1.

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*



: Minority sample



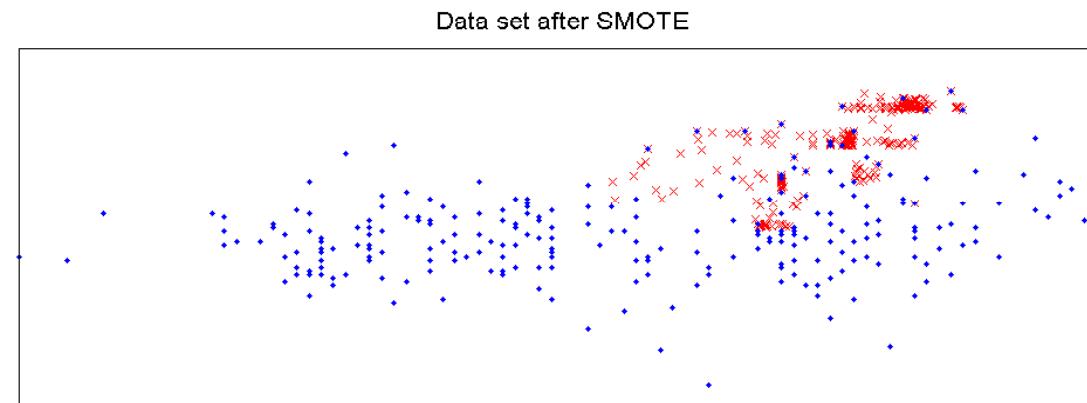
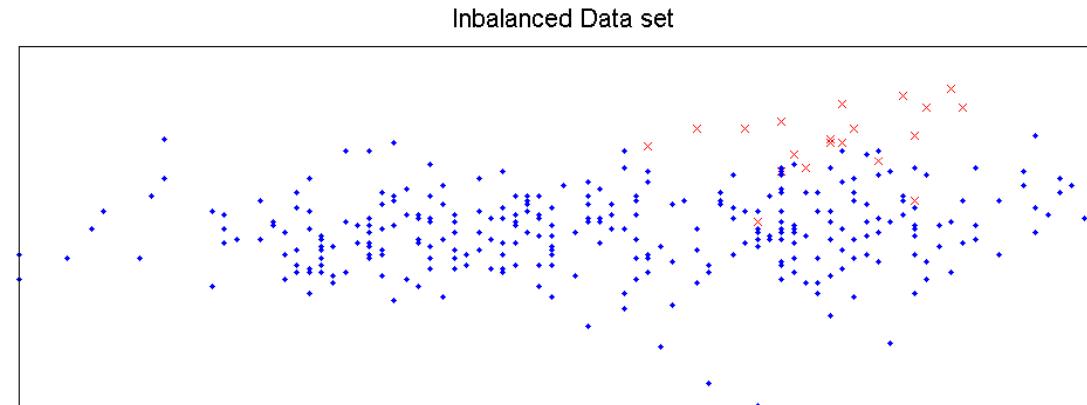
: Synthetic sample



: Majority sample

Resampling the original data sets

SMOTE : Example of a run



✖ Minority class + Majority class

Resampling the original data sets

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

SMOTE's Informed vs. Random Oversampling

- Random Oversampling (with replacement) of the minority class has the effect of making the decision region for the minority class very specific.
- In a decision tree, it would cause a new split and often lead to overfitting.
- SMOTE's informed oversampling generalizes the decision region for the minority class.
- As a result, larger and less specific regions are learned, thus, paying attention to minority class samples without causing overfitting.

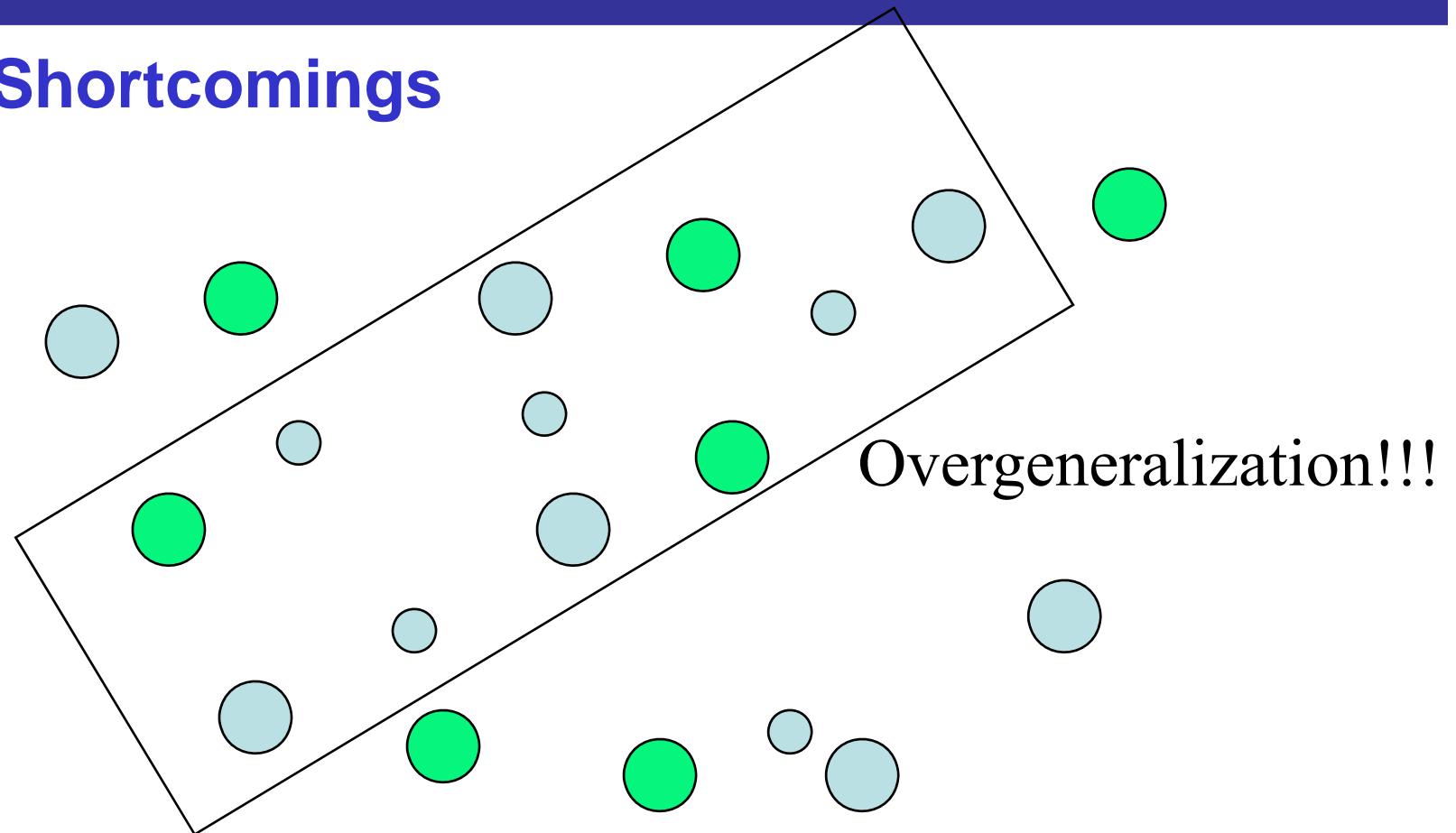
Resampling the original data sets

SMOTE Shortcomings

- **Overgeneralization**
 - SMOTE's procedure is inherently dangerous since it blindly generalizes the minority area without regard to the majority class.
 - This strategy is particularly problematic in the case of highly skewed class distributions since, in such cases, the minority class is very sparse with respect to the majority class, thus resulting in a greater chance of class mixture.
- **Lack of Flexibility**
 - The number of synthetic samples generated by SMOTE is fixed in advance, thus not allowing for any flexibility in the re-balancing rate.

Resampling the original data sets

SMOTE Shortcomings



: Minority sample



: Majority sample



: Synthetic sample

Resampling the original data sets

SMOTE: Hybridization

- Problem with Smote: might introduce the artificial minority class examples too deeply in the majority class space.
- Tomek links: data cleaning
- **Smote + Tomek links:** Instead of removing only the majority class examples that form Tomek links, examples from both classes are removed

Resampling the original data sets

SMOTE hybridization: SMOTE + Tomek links

Figure: SMOTE+TomekLink

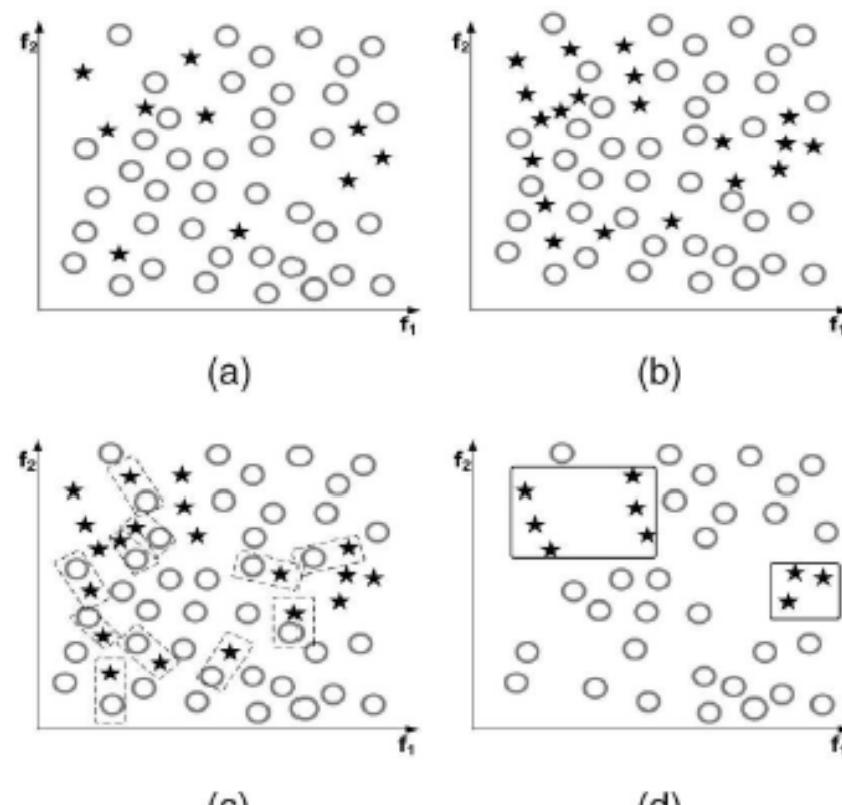
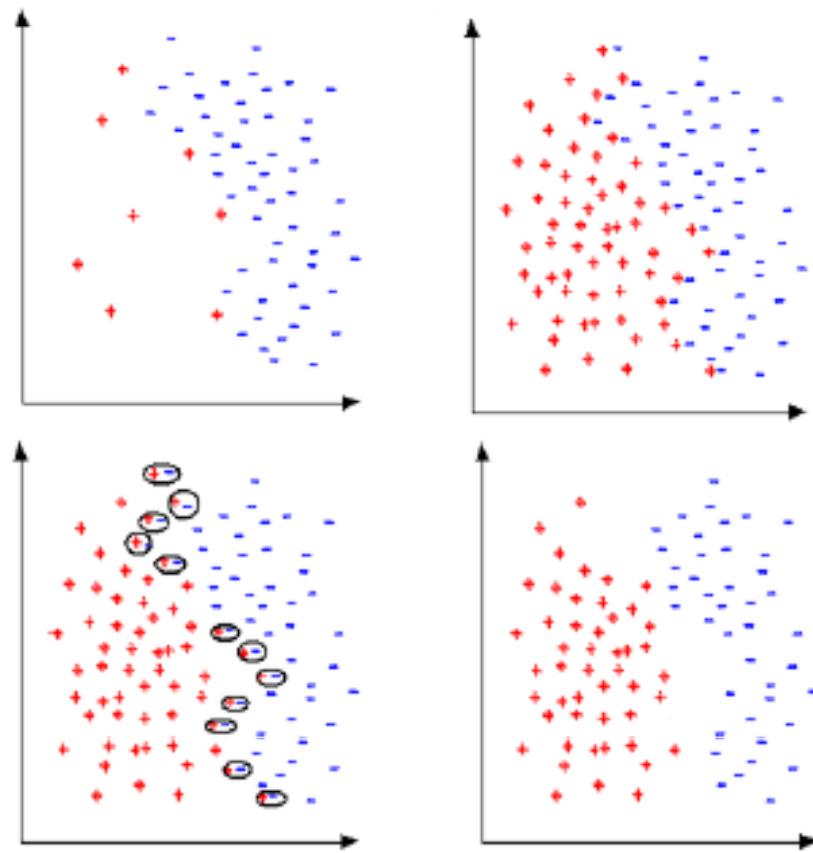


Figure 17: (a) Original data-set distribution. (b) Post-SMOTE data-set. (c) The identified Tomek Links. (d) The data-set after removing Tomek links

Resampling the original data sets

SMOTE hybridization: SMOTE + ENN

- ENN removes any example whose class label differs from the class of at least two of their neighbors
- ENN remove more examples than the Tomek links does
- ENN remove examples from both classes

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*
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*
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+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

SMOTE hybridization: SMOTE-Bordeline

Example: Borderline-SMOTE

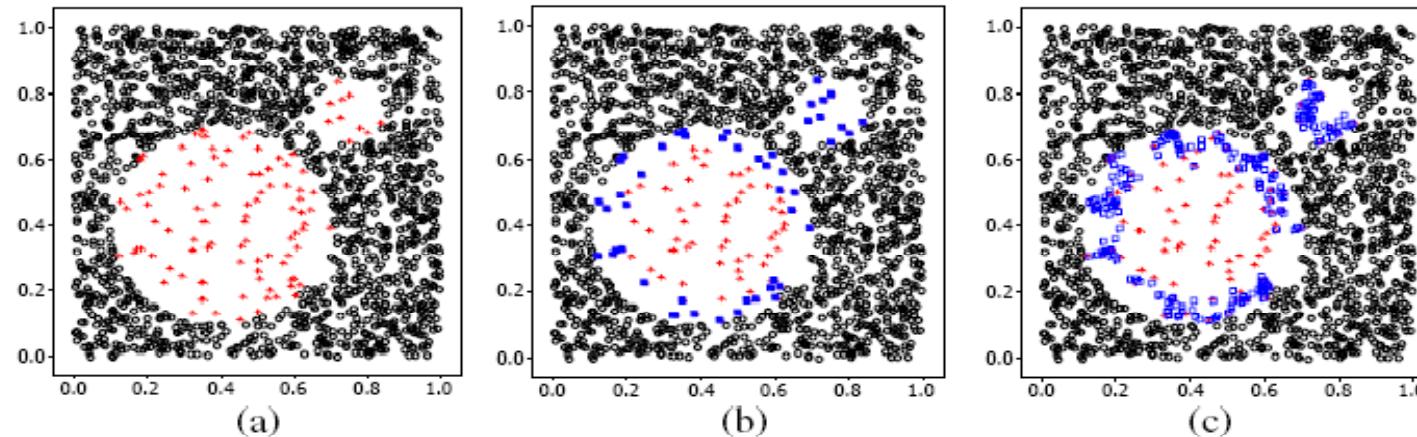


Fig. 1. (a) The original distribution of Circle data set. (b) The borderline minority examples (*solid squares*). (c) The borderline synthetic minority examples (*hollow squares*).

H. Han, W. Wang, B. Mao. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: ICIC 2005. LNCS 3644 (2005) 878-887.

Resampling the original data sets

Other Oversampling algorithms:

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

Algorithm steps

1. **Step1:** Using SMOTE we create a set synthetic data (*syntheticInstance[]*) for the minority class until the training set is balanced.
2. **Step2:** Create *resultSet*: including the original instances.
3. **Step3:** Construct the *similarityMatrix* for synthetic instances organized in rows and negative instances (majority class) in columns, using expression 7 and considering all the features in the equivalence relation. In the *similarityMatrix(i, j)* we will find the similarity degree between instances *i* and *j*.
4. **Step4:** For every synthetic example count the number of similar examples in the negative class.
npos - syn: number of synthetic instances
similarityValue := 0.4
While (*resultSet* is empty) & (*similarityValue* ≤ 0.9) **do**
 for *i* → 1 **to** *npos - syn*
 for *j* → 1 **to** *nneg*
 if (*similarityMatrix(i, j)* > *similarityValue*)
 then *cont[i]* ++
 endfor
 if *cont[i]* = 0 **then** // the instances are in the lower approximation,
 insert *syntheticInstance[i]* in *resultset* // (final training set)
 endfor
 similarityValue := *similarityValue* + 0.05
endwhile
5. **Step5:** If there are no instances in the lower approximation, i.e all synthetic instances are similar to other positive ones, the solution is given as the set balanced with SMOTE, all synthetic instances are included in "resultSet".

Fig. 5 Algorithm SMOTE-RSB*

Resampling the original data sets

Other Oversampling algorithms:

SMOTE-RSB*: Experimental Study: 44 high imbalanced data-sets, C4.5 learning algorithm

Table 3 Winner algorithm

	Original	Smote	S-TL	S-ENN	Border1	Border2	Safelevel	S-RSB _*	Total
Test	1/3	3/1	4/1	4/2	0/4	5/2	7/1	15/3	39/5 ties

Absolute winner/ties

Table 5 Rankings obtained through Friedman's test

Algorithm	Ranking
S-RSB _*	2.61364
S-ENN	3.92045
S-TL	3.96591
Smote	4.23864
Safelevel	4.34091
Borderline-SMOTE2	5.18182
Borderline-SMOTE1	5.36364

Table 6 Holm's table for $\alpha = 0.05$, S-RSB_{*} is the control method

i	Algorithm	$z = (R_0 - R_i)/SE$	p	Holm/Hochberg/Hommel	Hypothesis
6	Borderline-SMOTE1	5.2658490926	1.395E-7	0.008333	Reject
5	Borderline-SMOTE2	4.9176937807	8.756E-7	0.01	Reject
4	Safelevel	3.3074754631	9.414E-4	0.0125	Reject
3	Smote	3.1116381002	0.001860	0.016667	Reject
2	S-TL	2.5894051323	0.009614	0.025	Reject
1	S-ENN	2.5023663043	0.012336	0.05	Reject

Resampling the original data sets

Other Oversampling algorithms:

ADASYN: H. He, Y. Bai, E. A. Garcia, S. Li. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. International Joint Conference on Neural Networks (IJCNN'08) art. no. 4633969, pp. 1322-1328

ADOMS: S. Tang, S. Chen. The Generation Mechanism of Synthetic Minority Class Examples. 5th Int. Conference on Information Technology and Applications in Biomedicine, ITAB 2008 in conjunction with 2nd Int. Symposium and Summer School on Biomedical and Health Engineering, IS3BHE 2008, art. no. 4570642, pp. 444-447

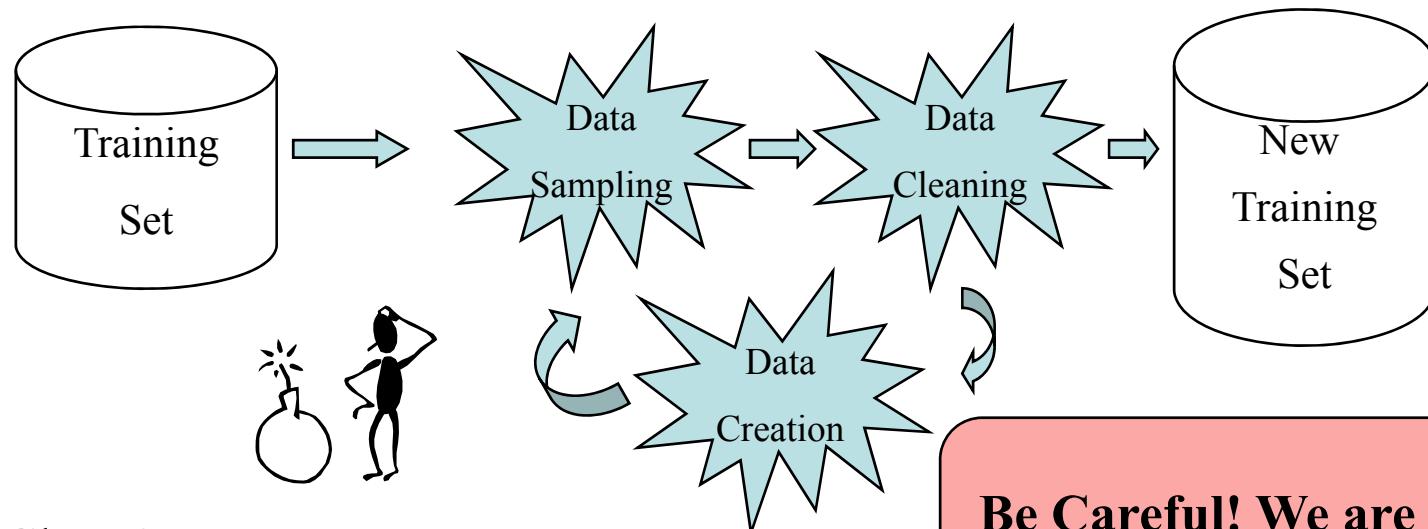
ASMO: Wang, B.X. and Japkowicz, N., "Imbalanced Data Set Learning with Synthetic Examples", presented at the IRIS Machine Learning Workshop, Ottawa, June 9, 2004.

Polynomial Fitting: S. Gazzah, N.E.B. Amara. New oversampling approaches based on polynomial fitting for imbalanced data sets. The Eighth IAPR International Workshop on Document Analysis Systems (DAS08). (2008) 677-684

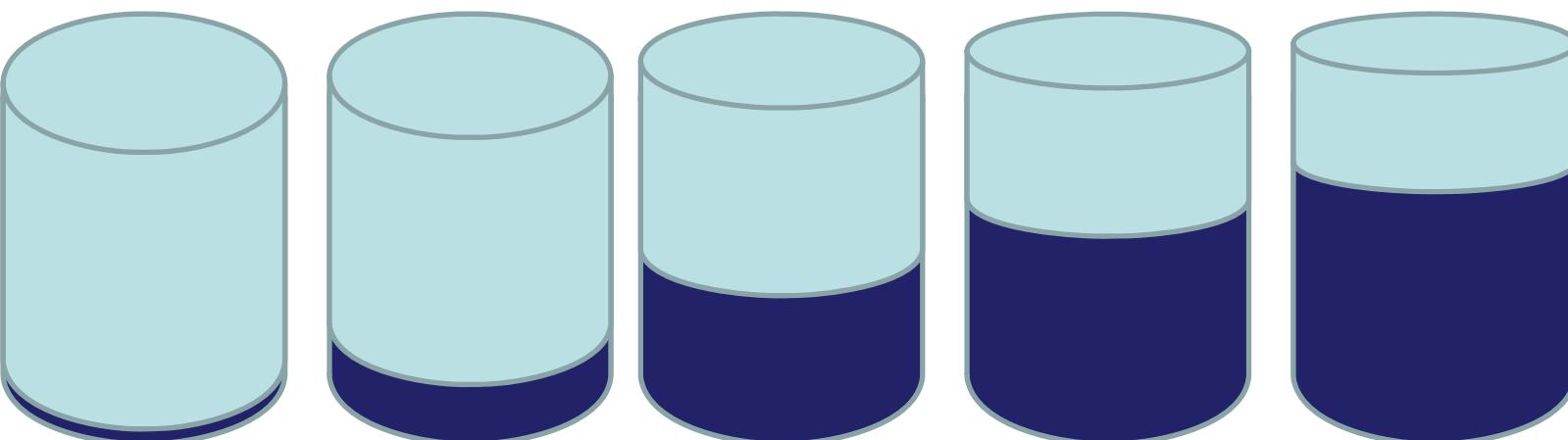
SPIDER 2: Krystyna Napierala, Jerzy Stefanowski, and Szymon Wilk. Learning from Imbalanced Data in Presence of Noisy and Borderline Examples. M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 158–167, 2010.

Resampling the original data sets

Final comments



**Be Careful! We are changing
what we were supposed to learn!**



Resampling the original data sets

Final comments

❖ Analysing the balance for resampling (see the study)

Chawla NV, Cieslak DA, Hall LO, Joshi A. Automatically counteracting imbalance and its empirical relationship to cost.

Data Mining and Knowledge Discovery 17:2 (2008) 225-252.

- ❖ It is not possible to know, apriori, whether a given domain favours oversampling or undersampling and what resampling rate is best.
- ❖ It would have interest to create combination schemes that considers both strategies at various rates.
- ❖ Improvements on resampling – specialized resampling
 - ❖ New approaches for creating artificial instances
 - ❖ How to choose the amount to sample?
 - ❖ New hybrid approaches oversampling vs undersampling



Contents

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance
- VII. Multiple class imbalanced data-sets: A pairwise learning approach
- VIII. Imbalanced Big Data
- IX. Class imbalance: Data sets, implementations, ...
- X. Class imbalance: Trends and final comments

SESSION 1

SESSION 2

SESSION 3

Some results on the use of evolutionary prototype selection for imbalanced data sets

Why evolutionary prototype selection?

- Randomly we can remove majority class examples.
- Risk of losing potentially important majority class examples that help establish the discriminating power.
- Evolutionary algorithms can be guided by different measures avoiding the lost of potential important examples.

Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Under-Sampling

Experimental Framework and Results

Evolutionary Training Set Selection to Optimize C4.5 in Imbalanced Problems

Concluding Remarks Future work

Source:

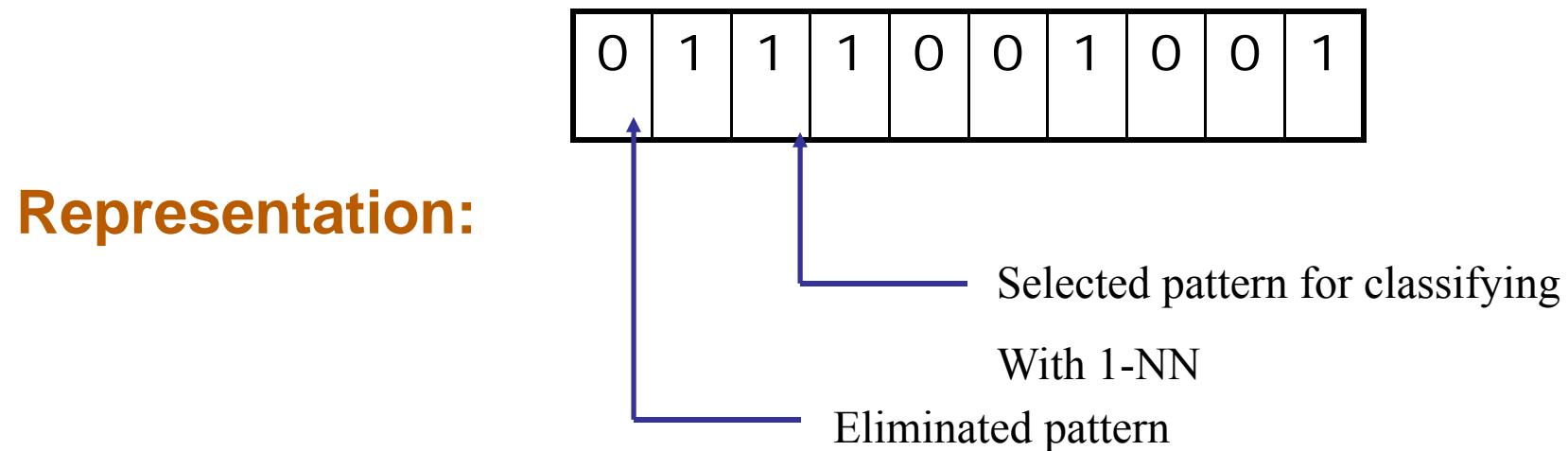
S. García, [F. Herrera](#), Evolutionary Under-Sampling for Classification with Imbalanced Data Sets: Proposals and Taxonomy. *Evolutionary Computation* 17:3 (2009) 275-306.

S. García, A. Fernandez, [F. Herrera](#), Enhancing the Effectiveness and Interpretability of Decision Tree and Rule Induction Classifiers with Evolutionary Training Set Selection over Imbalanced Problems. *Applied Soft Computing* 9 (2009) 1304-1314

Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Under-sampling

Motivation: Evolutionary algorithms/genetic algorithms for instance selection (prototype selection and training sets selection)



Evolutionary algorithms are good global search methods

Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Under-sampling

Previous results on Evolutionary Instance Selection:

J.R. Cano, F. Herrera, M. Lozano, Using Evolutionary Algorithms as Instance Selection for Data Reduction in KDD: An Experimental Study. *IEEE Trans. on Evolutionary Computation* 7:6 (2003) 561-575, [doi: 10.1109/TEVC.2003.819265](https://doi.org/10.1109/TEVC.2003.819265)

J.R. Cano, F. Herrera, M. Lozano, Stratification for Scaling Up Evolutionary Prototype Selection. *Pattern Recognition Letters*, 26, (2005), 953-963, [doi: 10.1016/j.patrec.2004.09.043](https://doi.org/10.1016/j.patrec.2004.09.043)

J.R. Cano, F. Herrera, M. Lozano, On the Combination of Evolutionary Algorithms and Stratified Strategies for Training Set Selection in Data Mining. *Applied Soft Computing* 6 (2006) 323-332, [doi: 10.1016/j.asoc.2005.02.006](https://doi.org/10.1016/j.asoc.2005.02.006)

J.R. Cano, F. Herrera, M. Lozano, Evolutionary Stratified Training Set Selection for Extracting Classification Rules with Trade-off Precision-Interpretability. *Data and Knowledge Engineering* 60 (2007) 90-108, [doi:10.1016/j.datwk.2006.01.008](https://doi.org/10.1016/j.datwk.2006.01.008)

S. García, J.R. Cano, F. Herrera, A Memetic Algorithm for Evolutionary Prototype Selection: A Scaling Up Approach. *Pattern Recognition* 41:8 (2008) 2693-2709, [doi:10.1016/j.patcog.2008.02.006](https://doi.org/10.1016/j.patcog.2008.02.006)

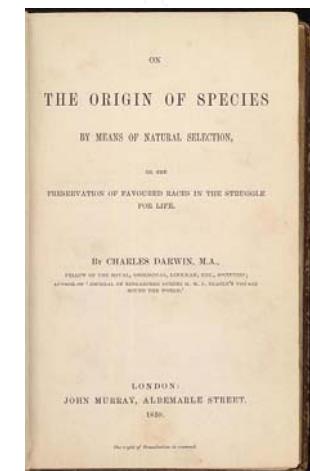
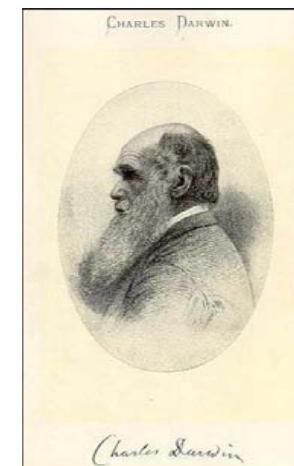
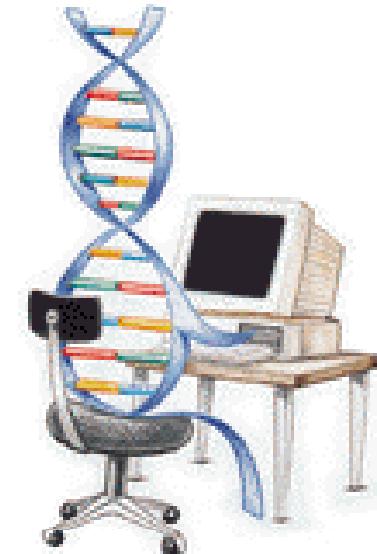
Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Under-sampling

Genetic algorithms

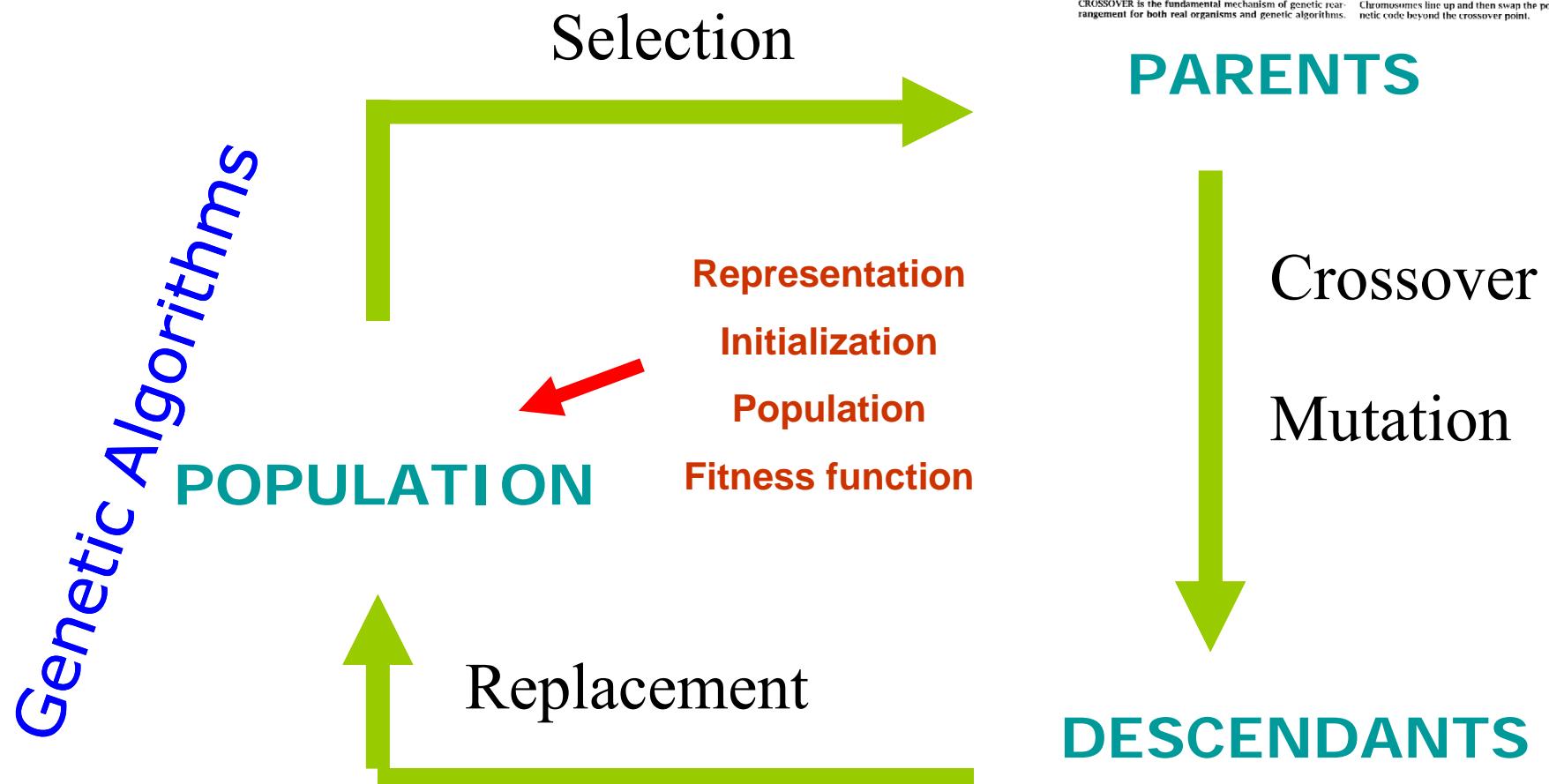
They are optimization algorithms,
search
and learning
inspired in the process of

Natural and Genetic Evolution



Some results on the use of evolutionary prototype selection for imbalanced data sets

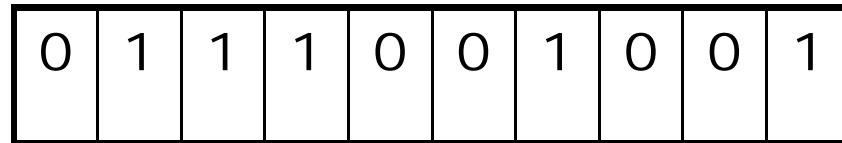
Evolutionary Under-sampling



Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Under-sampling

Representation:



Base Method: CHC

Models: EBUS and EUSCM

-EBUS: Aim for an optimal balancing of data without loss of effectiveness in classification accuracy

-EUSCM: Aim for an optimal power of classification without taking into account the balancing of data, considering the latter as a subobjective that may be an implicit process.

It introduces different features to obtain a trade-off between exploration and exploitation; such as incest prevention, reinitialization of the search process when it becomes blocked and the competition among parents and offspring into the replacement process

Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Under-sampling

Type of Selection:

- **GS: Global Selection**, the selection scheme proceeds over any kind of instance.
- **MS: Majority Selection**, the selection scheme only proceeds over majority class instances.

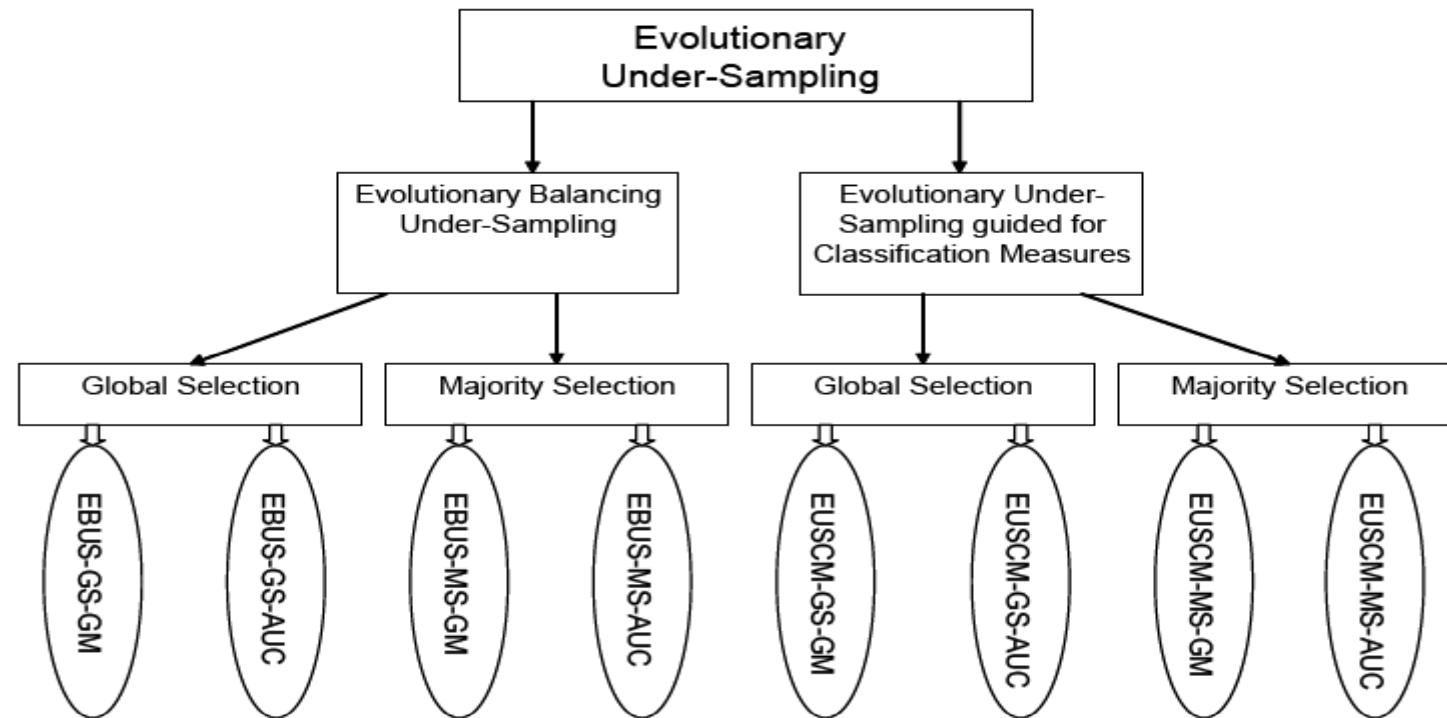
Evaluation Measures:

- **GM: Geometric Mean**
- **AUC: Area under ROC Curve**

Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Under-sampling

Taxonomy:



Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Under-sampling

Fitness function in EBUS model:

$$Fitness_{Bal}(S) = \begin{cases} g - |1 - \frac{n^+}{n^-}| \cdot P & \text{if } n^- > 0 \\ g - P & \text{if } n^- = 0 \end{cases} \quad Fitness_{Bal}(S) = \begin{cases} AUC - |1 - \frac{n^+}{n^-}| \cdot P & \text{if } n^- > 0 \\ AUC - P & \text{if } n^- = 0 \end{cases}$$

P: is a penalization factor that controls the intensity and importance of the balance during the evolutionary search.

P = 0.2 works appropriately.

Fitness function in EUSCM model:

$$Fitness(S) = g, \quad Fitness(S) = AUC,$$

Some results on the use of evolutionary prototype selection for imbalanced data sets

Experimental framework and results

Algorithms used in the comparison:

Prototype Selection:

IB3

DROP3

EPS-CHC

EPS-IGA

Under-Sampling
based on
clustering

Undersampling:

Random Under-Sampling

TomekLinks (TL)

CNN

OSS

CNN+TL

NCL

CPM

SBC



Some results on the use of evolutionary prototype selection for imbalanced data sets

Experimental framework and results

Data sets:

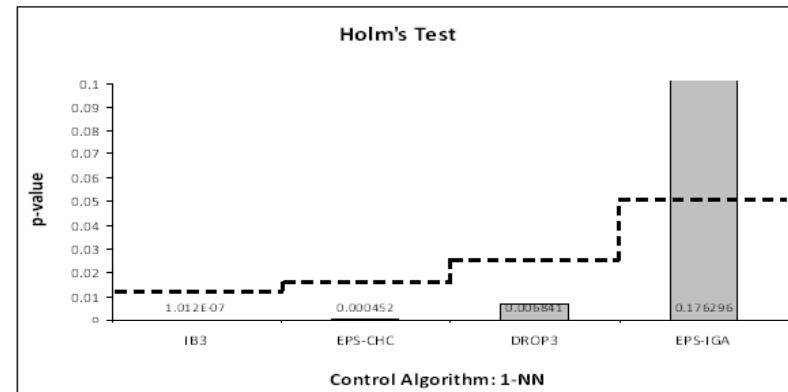
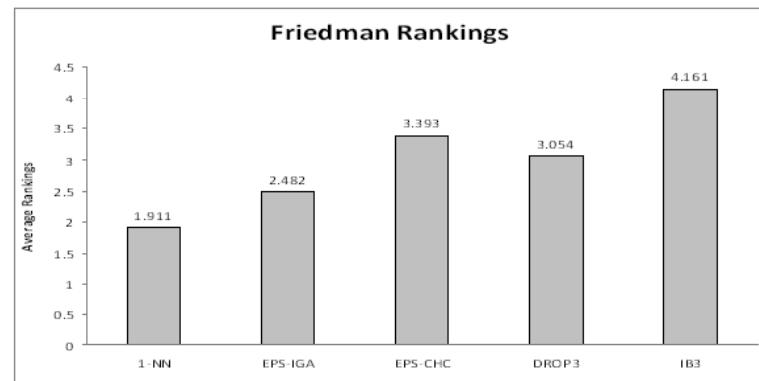
IR:
Imbalance ratio:
Number negative examples / Number positive examples

Data set	#Examples	#Attributes	Class (min., maj.)	%Class(min.,maj.)	IR
GlassBWNFP	214	9	(build-window-non_float-proc, remainder)	(35.51, 64.49)	1.82
EcoliCP-IM	220	7	(im, cp)	(35.00, 65.00)	1.86
Pima	768	8	(1,0)	(34.77, 66.23)	1.9
GlassBWFP	214	9	(build-window-float-proc, remainder)	(32.71, 67.29)	2.06
German	1000	20	(1, 0)	(30.00, 70.00)	2.33
Haberman	306	3	(Die, Survive)	(26.47, 73.53)	2.68
Splice-ie	3176	60	(ie, remainder)	(24.09, 75.91)	3.15
Splice-ei	3176	60	(ei, remainder)	(23.99, 76.01)	3.17
GlassNW	214	9	(non-windows glass, remainder)	(23.93, 76.17)	3.19
VehicleVAN	846	18	(van, remainder)	(23.52, 76.48)	3.25
EcoliIM	336	7	(im, remainder)	(22.92, 77.08)	3.36
New-thyroid	215	5	(hypo, remainder)	(16.28, 83.72)	4.92
Segment1	2310	19	(1, remainder)	(14.29, 85.71)	6.00
EcoliIMU	336	7	(iMU, remainder)	(10.42, 89.58)	8.19
Optdigits0	5564	64	(0, remainder)	(9.90, 90.10)	9.10
Satimage4	6435	36	(4, remainder)	(9.73, 90.27)	9.28
Vowel0	990	13	(0, remainder)	(9.01, 90.99)	10.1
GlassVWFP	214	9	(Ve-win-float-proc, remainder)	(7.94, 92.06)	10.39
EcoliOM	336	7	(om, remainder)	(6.74, 93.26)	13.84
GlassContainers	214	9	(containers, remainder)	(6.07, 93.93)	15.47
Abalone9-18	731	9	(18, 9)	(5.75, 94.25)	16.68
GlassTableware	214	9	(tableware, remainder)	(4.2, 95.8)	22.81
YeastCYT-POX	483	8	(POX, CYT)	(4.14, 95.86)	23.15
YeastME2	1484	8	(ME2, remainder)	(3.43, 96.57)	28.41
YeastME1	1484	8	(ME1, remainder)	(2.96, 97.04)	32.78
YeastEXC	1484	8	(EXC, remainder)	(2.49, 97.51)	39.16
Car	1728	6	(good, remainder)	(3.99, 96.01)	71.94
Abalone19	4177	9	(19, remainder)	(0.77, 99.23)	128.87

Some results on the use of evolutionary prototype selection for imbalanced data sets

Experimental framework and results

Part I: Classical prototype selection as imbalanced undersampling

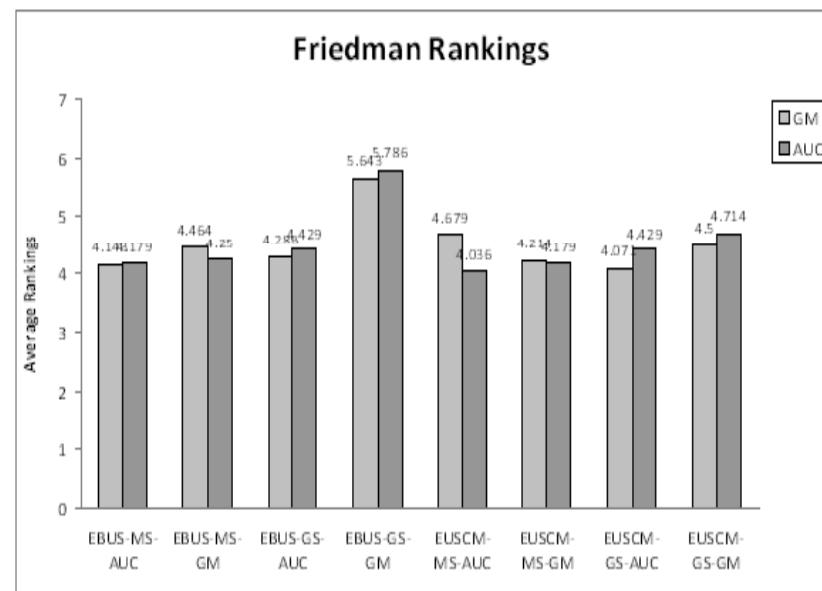


Classical prototype selection is not recommendable for tackling imbalanced data sets. 1-NN without preprocessing behaves the best.

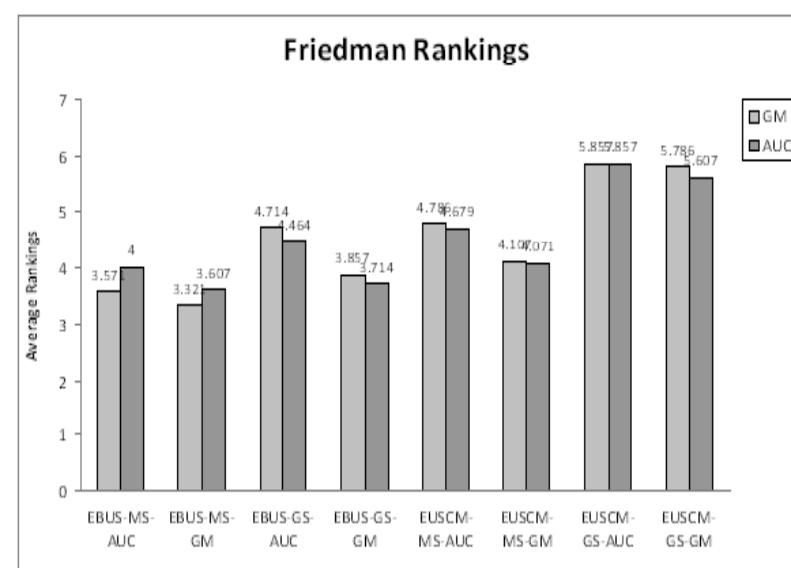
Some results on the use of evolutionary prototype selection for imbalanced data sets

Experimental framework and results

Part II: Comparison among the eight proposals of Evolutionary Under-Sampling



IR < 9



IR > 9

Some results on the use of evolutionary prototype selection for imbalanced data sets

Experimental framework and results

Part II: Comparison among the eight proposals of Evolutionary Under-Sampling

IR < 9:

- EUSCM behaves better than EBUS (P factor has little interest)
- Little differences between GM and AUC.

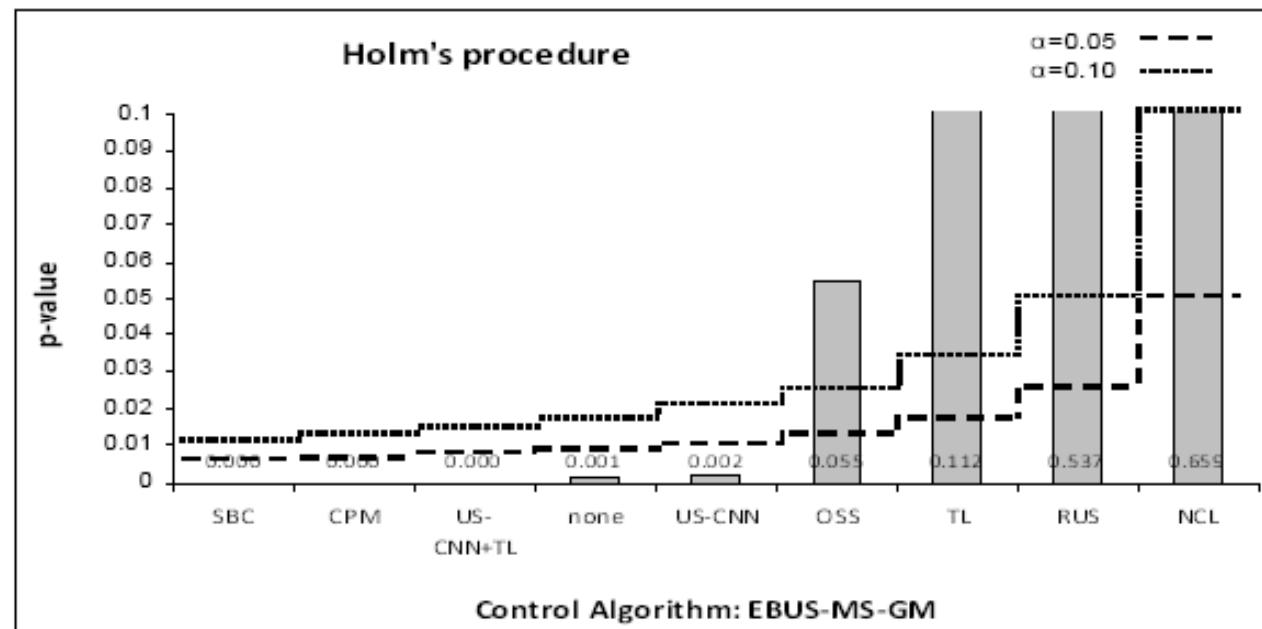
IR > 9:

- GS mechanism has no sense due to the high imbalance ratio. MS is preferable.
- P factor is very useful in this case. EBUS outperforms EUSCM

Some results on the use of evolutionary prototype selection for imbalanced data sets

Experimental framework and results

Part III: Comparison with other under-sampling approaches

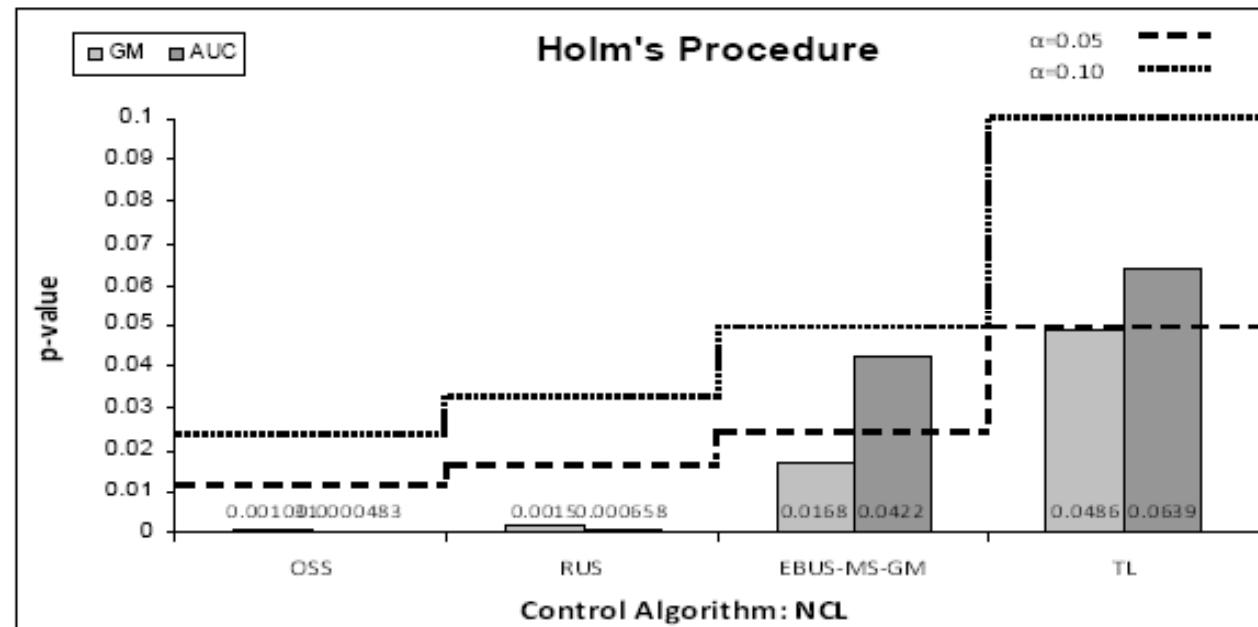


Considering all data sets

Some results on the use of evolutionary prototype selection for imbalanced data sets

Experimental framework and results

Part III: Comparison with other under-sampling approaches

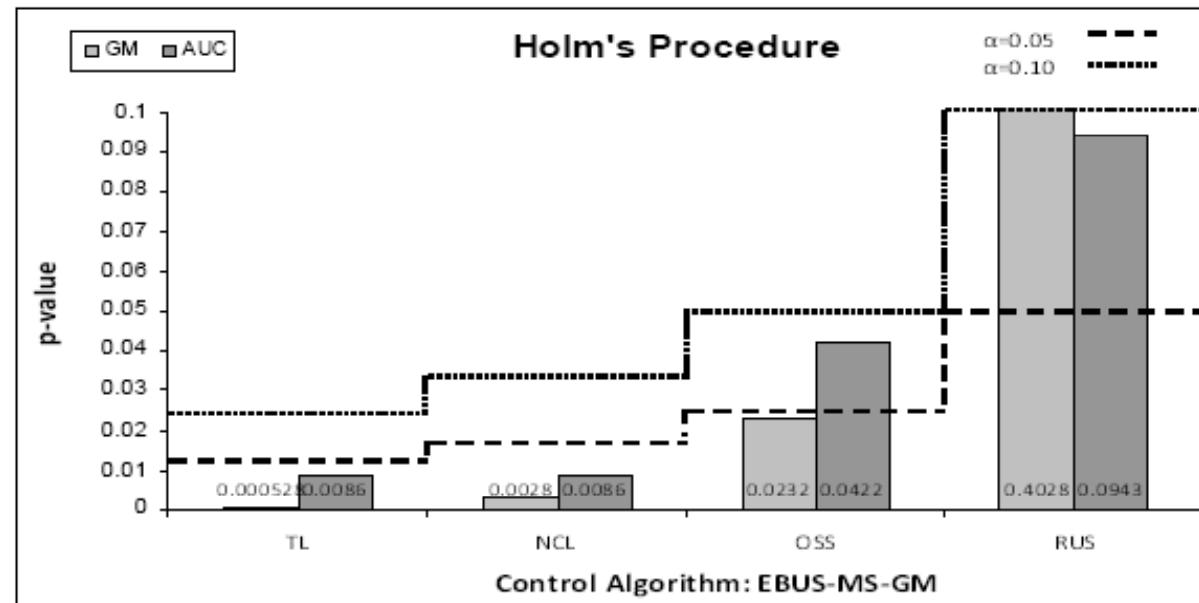


Considering data sets with IR < 9

Some results on the use of evolutionary prototype selection for imbalanced data sets

Experimental framework and results

Part III: Comparison with other under-sampling approaches



Considering data sets with $IR > 9$

Some results on the use of evolutionary prototype selection for imbalanced data sets

Experimental framework and results

Part III: Comparison with other under-sampling approaches

- EUS models usually present an equal or better performance than the remaining methods, independently of the degree of imbalance of data.
- The best performing under-sampling model over imbalance data sets is EBUS-MSGM
- The tendency of the EUS models follows an improving of the behaviour in classification when the data turns to a high degree of imbalance.

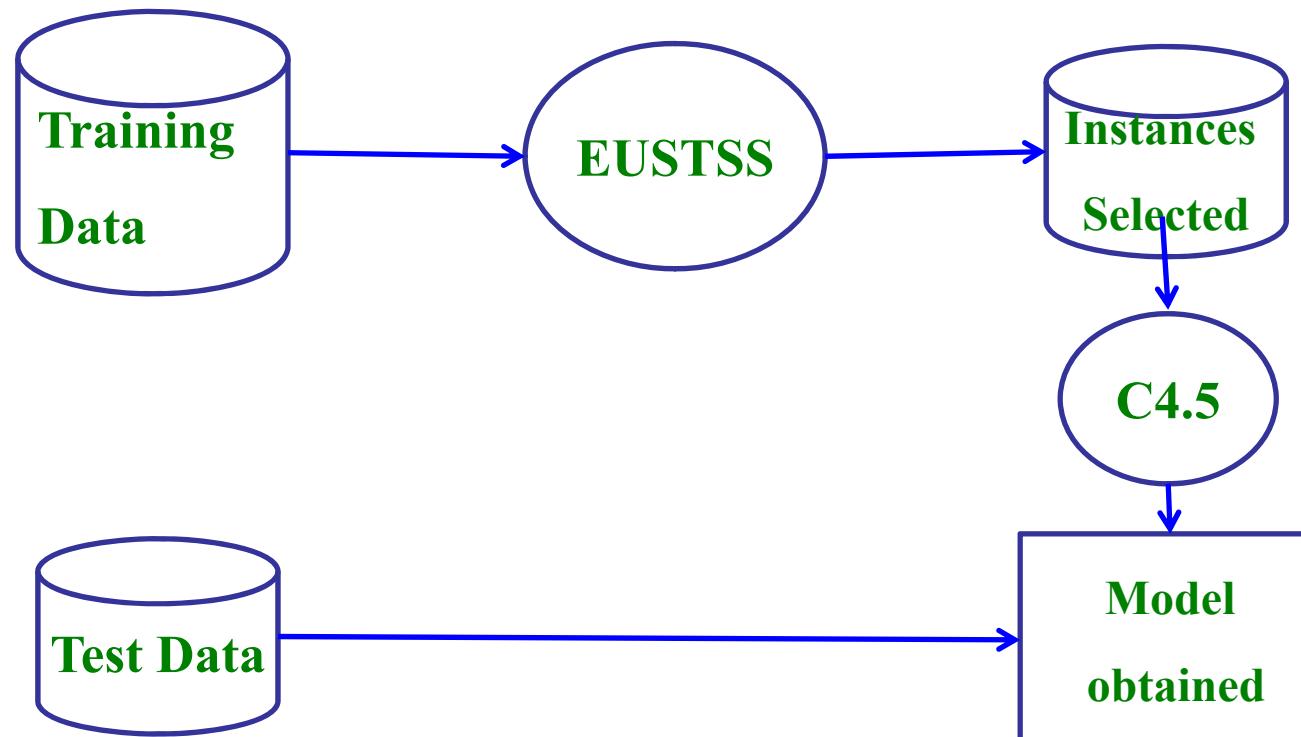
Some results on the use of evolutionary prototype selection for imbalanced data sets

Experimental framework and results

- Prototype Selection methods are not useful when handling imbalanced problems.
- Evolutionary under-sampling is an effective model in instance-based learning.
- Majority selection mechanism obtains more accurate subsets of instances, but presents a lower reduction rate.
- No difference between GM and AUC (different evaluation measures) is observed.
- For dealing with low imbalance rates, EUSCM model is the best choice
- For dealing with high imbalance rates, EBUS model is the best.

Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Training Set Selection to Optimize C4.5 in Imbalanced Problems



S. García, A. Fernández, F. Herrera, **Enhancing the Effectiveness and Interpretability of Decision Tree and Rule Induction Classifiers with Evolutionary Training Set Selection over Imbalanced Problems**. *Applied Soft Computing* 9 (2009) 1304-1314

Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Training Set Selection to Optimize C4.5 in Imbalanced Problems

Avoiding overfitting in TSS:

- Although C4.5 incorporates a pruning mechanism to avoid overfitting, the use of an induction tree process within an evolutionary cycle may yield optimal models for training data, losing generalization capabilities.
- To avoid it, we include classification costs in the fitness function. A well classified instance scores a value of W if it is not selected in the chromosome and it scores 1 if it is selected in the chromosome. The penalization yielded in case of misclassification is the same.
- Our empirical studies determine that $W=3$ works well.

Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Training Set Selection to Optimize C4.5 in Imbalanced Problems

Data sets:

Data set	#Examples	#Attributes	Class (min., maj.)	% Class(min.,maj.)
Abalone9-18	731	9	(18, 9)	(5.75, 94.25)
EcoliCP-IM	220	7	(im,cp)	(35.00, 65.00)
EcoliIM	336	7	(im,remainder)	(22.92, 77.08)
EcoliIMU	336	7	(iMU, remainder)	(10.42, 89.58)
EcoliOM	336	7	(om, remainder)	(6.74, 93.26)
German	1000	20	(1, 0)	(30.00, 70.00)
GlassBWFP	214	9	(build-window-float-proc, remainder)	(32.71, 67.29)
GlassBWNFP	214	9	(build-window-non_float-proc, remainder)	(35.51, 64.49)
GlassNW	214	9	(non-windows glass, remainder)	(23.93, 76.17)
GlassVWFP	214	9	(Ve-win-float-proc, remainder)	(7.94, 92.06)
Haberman	306	3	(Die, Survive)	(26.47, 73.53)
New-thyroid	215	5	(hypo,remainder)	(16.28, 83.72)
Pima	768	8	(1,0)	(34.77, 66.23)
VehicleVAN	846	18	(van,remainder)	(23.52, 76.48)
Vowel0	990	13	(0, remainder)	(9.01, 90.99)
YeastCYT-POX	483	8	(POX, CYT)	(4.14, 95.86)

We have used 10-fcv. Algorithms in the comparison: OSS, NCL, SMOTE, SMOTE + TL, SMOTE + ENN

Parameters of EUSTSS: Pop = 50, Eval = 10000.

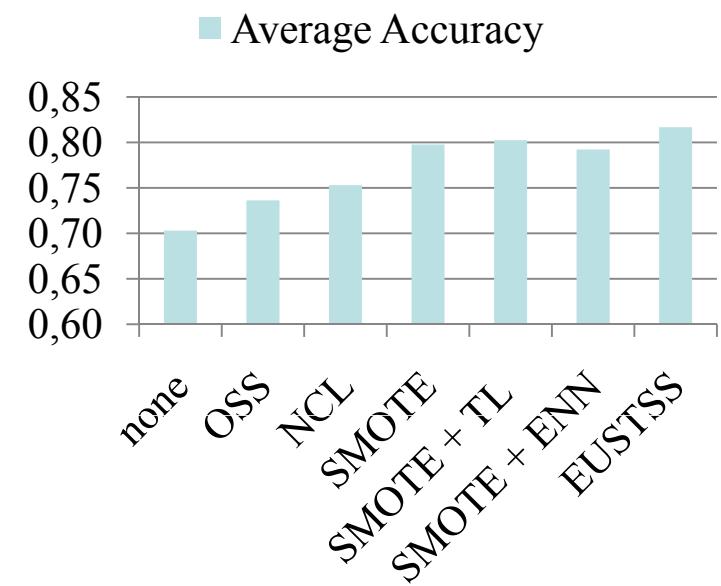
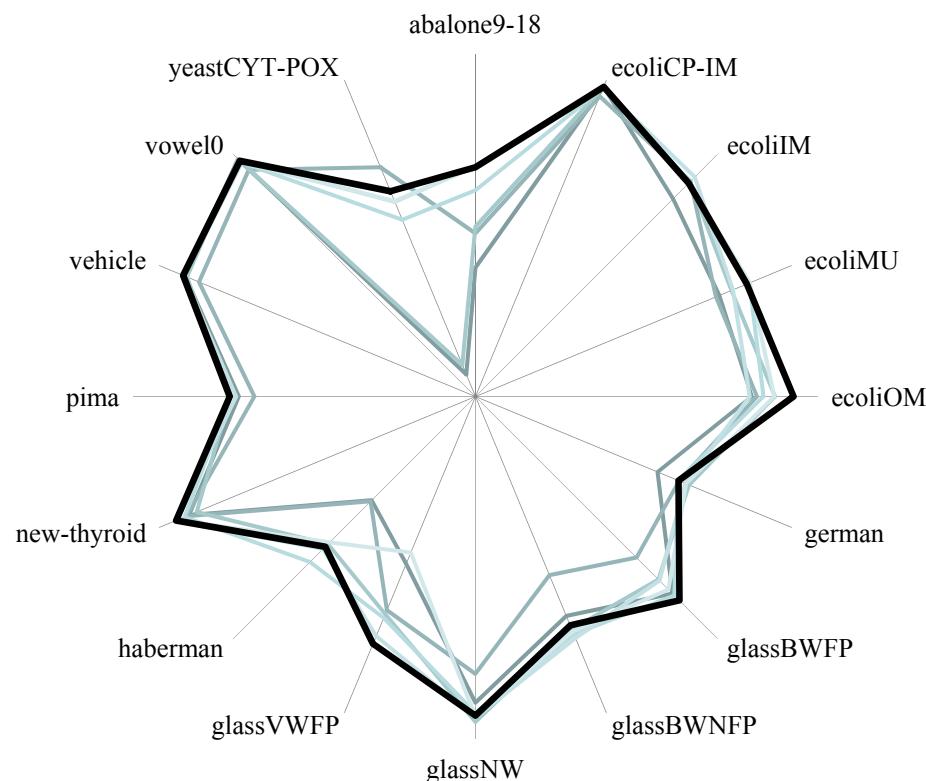
Parameters of SMOTE: k = 5, Balancing Ratio 1:1.

Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Training Set Selection to Optimize C4.5 in Imbalanced Problems

Results obtained by C4.5 using GM evaluation measure over test data

— none — OSS — NCL — SMOTE
— SMOTE + TL — SMOTE + ENN — EUSTSS

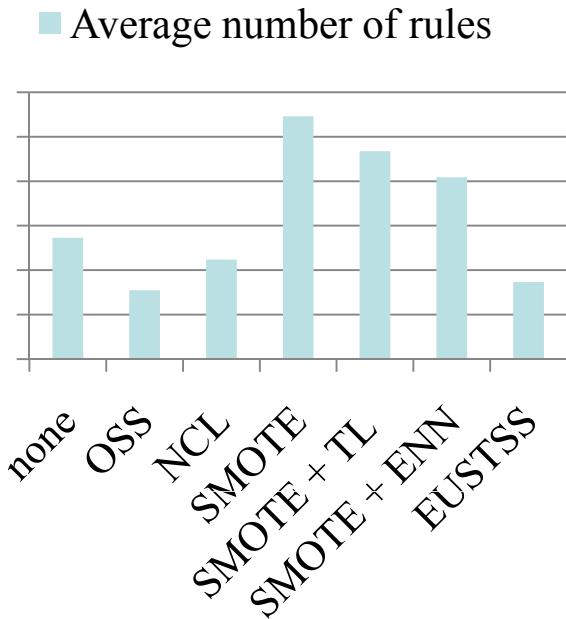
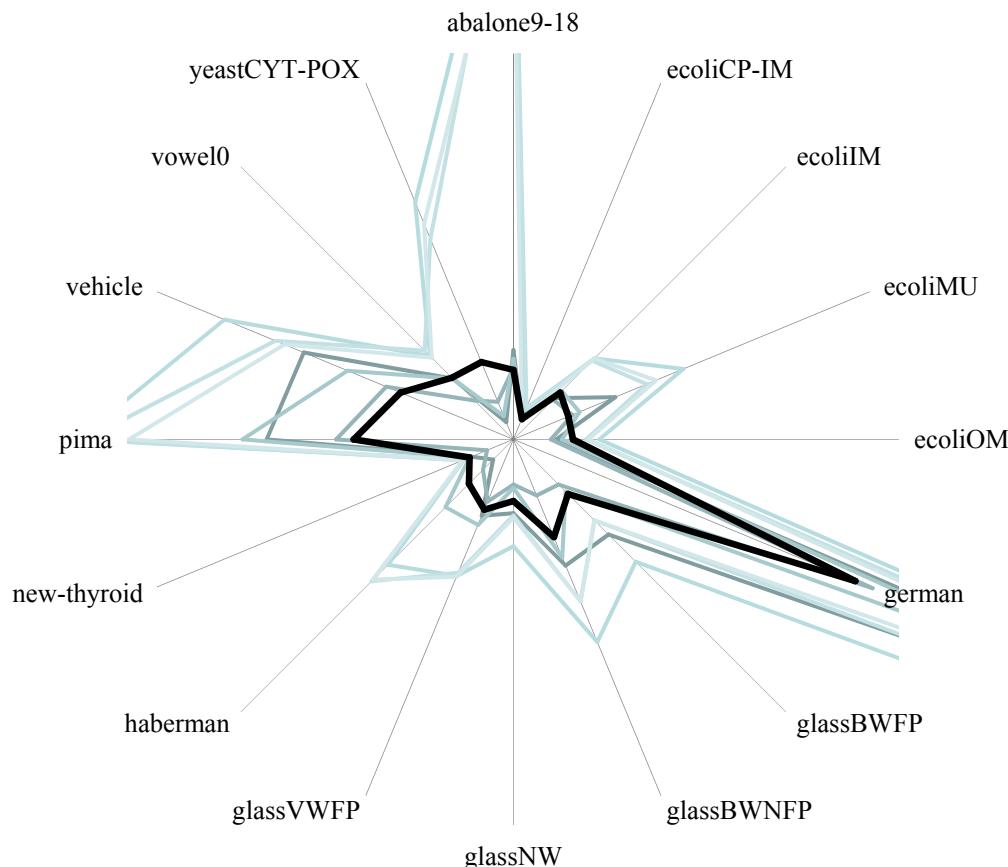


Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Training Set Selection to Optimize C4.5 in Imbalanced Problems

Average number of rules obtained by C4.5 decision tree

— none — OSS — NCL — SMOTE — SMOTE + TL — SMOTE + ENN — EUSTSS



Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Training Set Selection to Optimize C4.5 in Imbalanced Problems

- EUSTSS obtains the best average result in GM. It clearly outperforms the other undersampling methods (OSS and NCL) and it improves the accuracy even when comparing with over-sampling approaches.
- Over-sampling is clearly superior to under-sampling, except for the EUSTSS technique.
- Except for OSS, EUSTSS produces decision trees with lower number of rules than the remaining methods. Although the combination OSS + C4.5 yields less rules, its accuracy in GM is the worst of all the resampling methods.
- Over-sampling force to C4.5 to produce many rules. This fact is not desirable when our interest lies in obtaining interpretable models.

Some results on the use of evolutionary prototype selection for imbalanced data sets

Evolutionary Training Set Selection to Optimize C4.5 in Imbalanced Problems

Wilcoxon's test results over GM and number of

algorithm	EUSTSS GM	EUSTSS num. rules
none	+ (.001)	+ (.088)
OSS	+ (.003)	- (.052)
NCL	+ (.047)	+ (.074)
SMOTE	+ (.052)	+ (.000)
SMOTE + TL	= (.501)	+ (.000)
SMOTE + ENN	= (.363)	+ (.000)

- Wilcoxon's test confirms the improvement offered by EUSTSS in accuracy.
- It again confirms that EUSTSS produces smaller decision trees than all the remaining method, except OSS.

Some results on the use of evolutionary prototype selection for imbalanced data sets

EGIS-CHC: Evolutionary Based selection of Generalized Instances

This paper proposes a method belonging to the family of the nested generalized exemplar that accomplishes learning by storing objects in Euclidean n-space.

Classification of new data is performed by computing their distance to the nearest generalized exemplar. The method is optimized by the selection of the most suitable generalized exemplars based on evolutionary algorithms.

S. García, [J. Derrac](#), [I. Triguero](#), C.J. Carmona, [F. Herrera](#), **Evolutionary-Based Selection of Generalized Instances for Imbalanced Classification.**
Knowledge Based Systems 25:1 (2012) 3-12.

Some results on the use of evolutionary prototype selection for imbalanced data sets

IPADE-ID: Evolutionary instance generation

Iterative Instance Adjustment for Imbalanced Domains (IPADE-ID) algorithm.

- It is an evolutionary framework, which uses an instance generation technique, designed to face the existing imbalance modifying the original training set.
- The method, iteratively learns the appropriate number of examples that represent the classes and their particular positioning.
- The learning process contains three key operations in its design: a customized initialization procedure, an evolutionary optimization of the positioning of the examples and a selection of the most representative examples for each class.

V. López, I. Triguero, C.J. Carmona, S. García, F. Herrera, **Addressing Imbalanced Classification with Instance Generation Techniques: IPADE-ID.** *Neurocomputing, in press (2013).*

Some results on the use of evolutionary prototype selection for imbalanced data sets

Concluding remarks and future work

- Evolutionary under-sampling is an effective model in instance-based learning.
- For dealing with low imbalance rates, EUSCM model is the best choice
- For dealing with high imbalance rates, EBUS model is the best.
- EUSTSS allows to C4.5 to obtain very accurate trees in imbalanced classification.
- The models are very competitive with respect to advanced hybrids of over-sampling and under-sampling.
- The number of leafs is decreased so the trees obtained are more interpretable.

Some results on the use of evolutionary prototype selection for imbalanced data sets

Concluding remarks and future work

- Study the scalability of these models in very large data sets.
- Hybridize evolutionary under-sampling with SMOTE or other over-sampling approaches.
- Data complexity analysis of the the EUS and EUSTSS behaviour

[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) (2011) 1909-1936

Contents

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance
- VII. Multiple class imbalanced data-sets: A pairwise learning approach
- VIII. Imbalanced Big Data
- IX. Class imbalance: Data sets, implementations, ...
- X. Class imbalance: Trends and final comments

SESSION 1

SESSION 2

SESSION 3

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.

This, in effect, rectifies the bias given to the majority class by standard classifiers when the training error corresponds to the simple (non-weighted) accuracy.

B. Zadrozny, J. Langford, N. Abe, Cost-sensitive learning by cost—proportionate example weighting, in: Proceedings of the 2003 IEEE International Conference on Data Mining (ICDM'03), 2003.

C. Elkan, The foundations of cost-sensitive learning, in: Proceedings of the 17th International Joint Conference on Artificial Intelligence, 2001, pp. 973–978.

Cost-sensitive learning

Over Sampling

Random

Focused

Under Sampling

Random

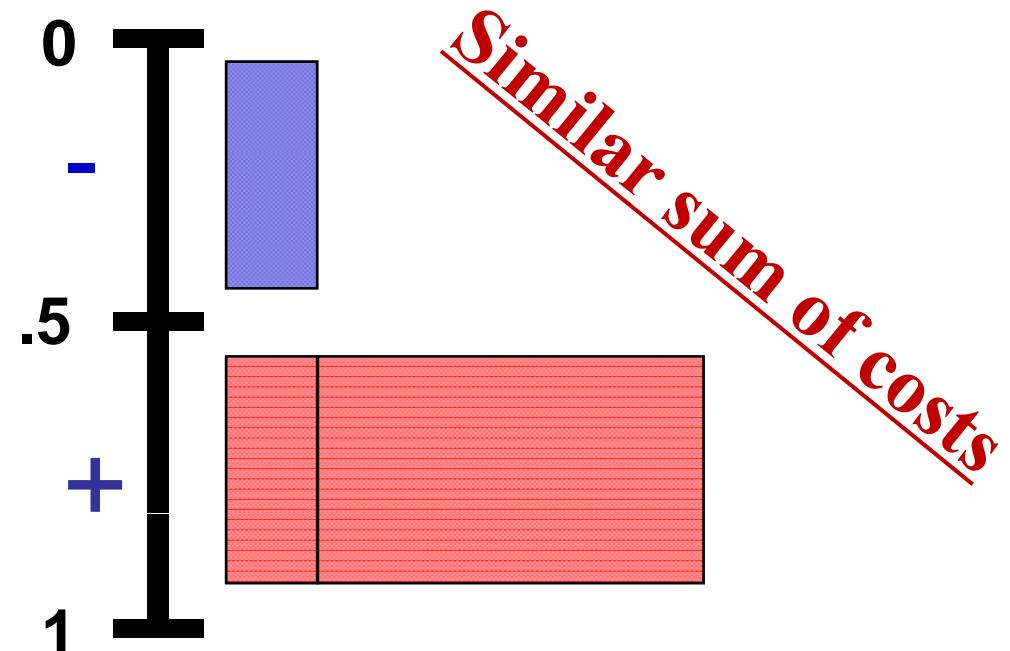
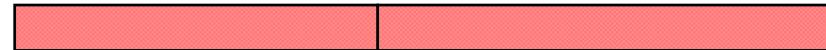
Focused

Cost Modifying

examples of -



examples of +



Cost-sensitive learning

- Needs a cost matrix, which encodes the penalty of misclassifying samples.
- In the scenario of imbalanced data-sets, the significance of the recognition of positive instances might be higher:
 - $C(+, -) > C(-, +)$
 - $C(+, +) = C(-, -) = 0$
- Consider the cost-matrix during the model building for achieving the lowest cost.
- However, the cost matrix is often unavailable

	actual negative	actual positive
predict negative	$C(0, 0) = c_{00}$	$C(0, 1) = c_{01}$
predict positive	$C(1, 0) = c_{10}$	$C(1, 1) = c_{11}$

	fraudulent	legitimate
refuse	\$20	-\$20
approve	$-x$	$0.02x$

C. Elkan, The foundations of cost-sensitive learning, in: Proceedings of the 17th International Joint Conference on Artificial Intelligence, 2001, pp. 973–978.

Cost-sensitive learning

- How can it be applied?
 - Weighting the data space (data level):
 - Change the distribution of the training sets (translation theorem) or
 - Modifying final decision thresholds
 - Making a specific classifier learning algorithm cost-sensitive (algorithm level)
 - Change the inner way the classifier works
 - Use a boosting approach
 - Using Bayes risk theory to assign each sample to its lowest risk class (algorithm level)

$$L(x, i) = \sum_j P(j|x)C(i, j)$$

P. Domingos, Metacost: a general method for making classifiers cost sensitive, in: 5th Int'l Conf. on Knowledge Discovery and Data Mining (KDD'99), San Diego, CA, 1999, pp. 155–164.

Ting, K.M. An instance-weighting method to induce cost-sensitive trees (2002) *IEEE Transactions on Knowledge and Data Engineering*, 14 (3), pp. 659-665.

Y. Sun, M. S. Kamel, A. K. C. Wong and Y. Wang, Cost-sensitive boosting for classification of imbalanced data, *Pattern Recognition* 40(12) (2007) 3358–3378

Cost-sensitive learning

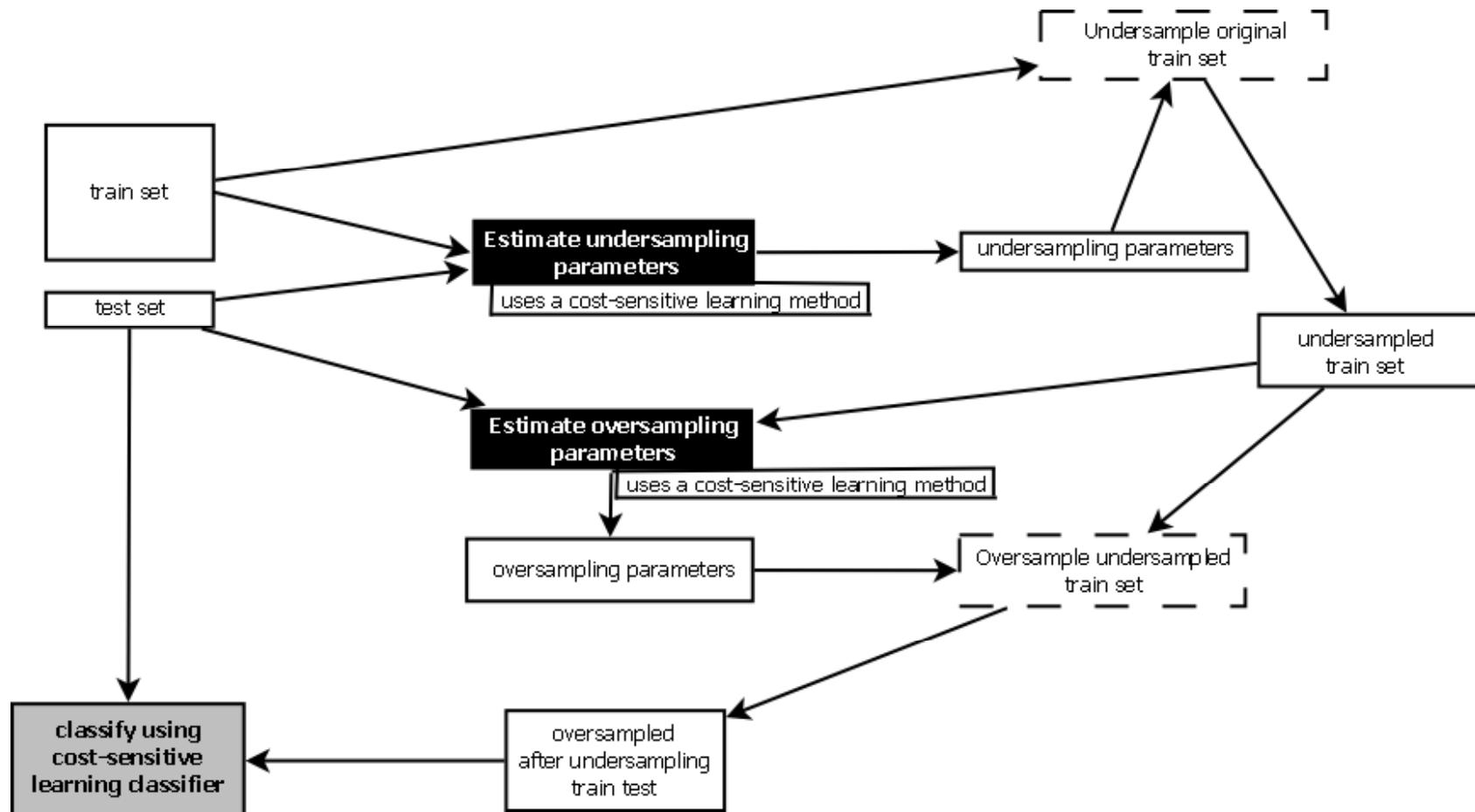
Two weighting approaches

- Up-weighting, analogous to over-sampling, increases the weight of one of the classes keeping the weight of the other class at one

- Down-weighting, analogous to under-sampling, decreases the weight of one of the classes keeping the weight of the other class at one

Cost-sensitive learning

Hybridization. Automatically countering imbalance



Chawla, N. V., Cieslak, D. A., Hall, L. O., Joshi, A., 2008. Automatically countering imbalance and its empirical relationship to cost. Data Mining and Knowledge Discovery 17 (2), 225–252

Cost-sensitive learning

Algorithms selected for the study

Acronym	Description
None	The original classifier that names the algorithm family
SMOTE	The original classifier that names the algorithm family applied to a data-set preprocessed with the SMOTE algorithm
SENN	The original classifier that names the algorithm family applied to a data-set preprocessed with the SMOTE+ENN algorithm
CS	The cost-sensitive version of the original classifier from the corresponding algorithm family.
Wr_SMOTE	Version of the Wrapper routine that uses as main algorithm the cost-sensitive version of the algorithm family and only performs the oversampling step with SMOTE
Wr_US	Version of the Wrapper routine that uses as main algorithm the cost-sensitive version of the algorithm family, performs the undersampling step with a random undersampling algorithm and the oversampling step with the SMOTE algorithm
Wr_SENN	Version of the Wrapper routine that uses as main algorithm the cost-sensitive version of the algorithm family and only performs the oversampling step with the SMOTE+ENN algorithm

Cost-sensitive learning

Algorithms selected for the study (2)

- **Decision Trees: C4.5**
 - Original: Quinlan, J. R., 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo—California.
 - Cost-Sensitive: Ting, K.M. An instance-weighting method to induce cost-sensitive trees (2002) *IEEE Transactions on Knowledge and Data Engineering*, 14 (3), pp. 659-665.
- **Support Vector Machines**
 - Original: Vapnik, V., 1998. *Statistical Learning Theory*. Wiley, New York, U.S.A.
 - Cost-Sensitive: R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using second order information for training SVM. *Journal of Machine Learning Research* 6, 1889-1918, 2005.
- **Fuzzy Rule Based Classification Systems: FH-GBML**
 - Original: Ishibuchi, H., Yamamoto, T., Nakashima, T., 2005. Hybridization of fuzzy GBML approaches for pattern classification problems. *IEEE Transactions on System, Man and Cybernetics B* 35 (2), 359–365.
 - Cost-Sensitive: V. López, A. Fernandez, F. Herrera, A First Approach for Cost-Sensitive Classification with Linguistic Genetic Fuzzy Systems in Imbalanced Data-sets. *10th International Conference on Intelligent Systems Design and Applications (ISDA2010)*, pp 676-681.
- **Lazy Learning: k-NN**
 - Original: Cover, T. M., and P. E. Hart, (1967). Nearest Neighbor Pattern Classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, Vol. 13, No. 1, pp. 21-27.
 - Cost-Sensitive: D.J. Hand and V. Vinciotti. Choosing k for two-class nearest neighbour classifiers with unbalanced classes. *Pattern Recognition Letters*, 24:1555–1562, 2003.

Cost-sensitive learning

Data-sets and Parameters

- **66 real-world data-sets**
- **5 fold-cross validation**
- **Available at KEEL data-set repository:**
<http://www.keel.es/dataset.php>

Algorithm Family	Parameters
C4.5	pruned = True confidence = 0.25 minimum number of item-sets per leaf = 2
SVM	kernel type = polynomial C = 100.0 tolerance of termination criterion = 0.001 degree (for kernel function) = 1 gamma (for kernel function) = 0.01 coef0 (for kernel function) = 0.0 use the shrinking heuristics = true
FH-GBML	conjunction operator = product t-norm rule weight = PCF (FH-GBML and FH-GBML+preprocessing) and PCF-SC (FH-GBML-CS) fuzzy reasoning method = winning rule number of fuzzy rules = $5 \cdot d$ (max. 50 rules) number of rule sets = 200 crossover probability = 0.9 mutation probability = $1/d$ number of replaced rules = all rules except the best-one (Pittsburgh-part, elitist approach) number of rules/5 (GCCL-part) total number of generations = 1.000 don't care probability = 0.5 probability of the application of the GCCL iteration = 0.5
k-NN	k = 3 distance = Heterogeneous Value Difference Metric (HVDM)

Cost-sensitive learning

Results and Statistical Analysis

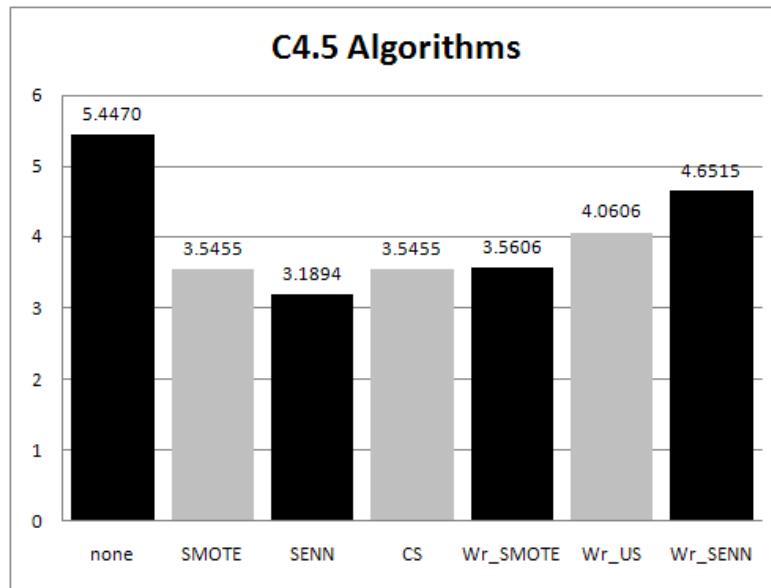
- Case of Study: C4.5
- Similar results and conclusions for the remaining classification paradigms

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	0.8390 ± 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. Fernández, 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	(6.404E-6)	(4.850E-6)	(6.404E-6)	(7.904E-6)	(.00341)	=(.37846)
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.

Contents

SESSION 2:

V. 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

Contents

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance
- VII. Multiple class imbalanced data-sets: A pairwise learning approach
- VIII. Imbalanced Big Data
- IX. Class imbalance: Data sets, implementations, ...
- X. Class imbalance: Trends and final comments

SESSION 1

SESSION 2

SESSION 3

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

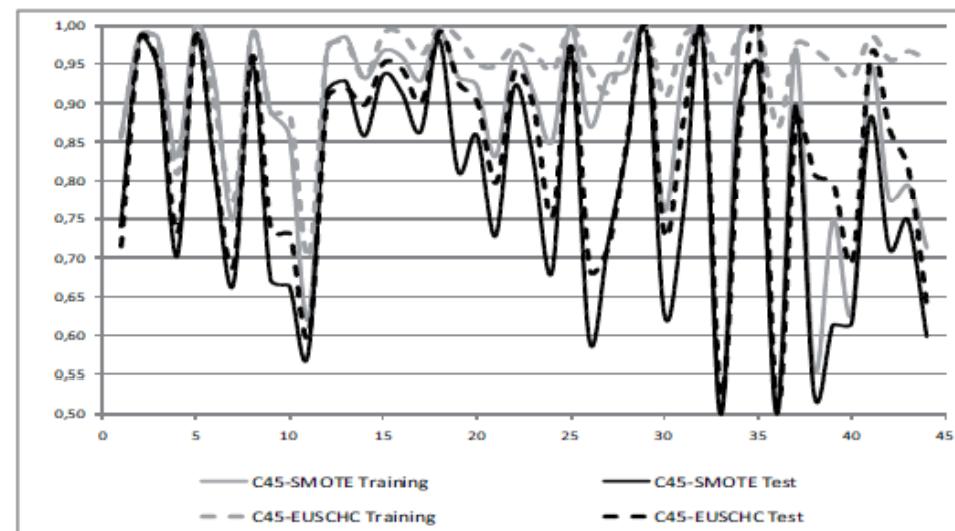
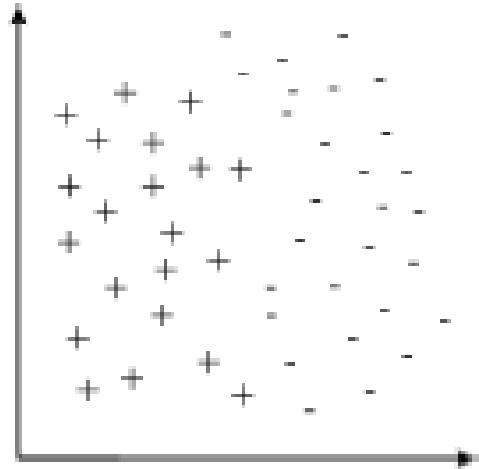


Fig. 4 C4.5 AUC in Training/Test sorted by IR

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.

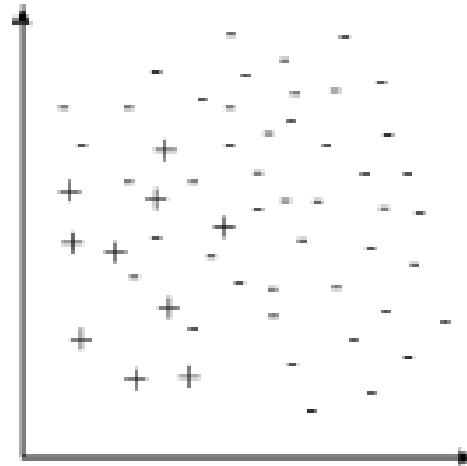
Introduction to Imbalanced Data Sets

Why is difficult to learn in imbalanced domains?



An easier problem

Imbalance – why is it difficult?



More difficult one

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

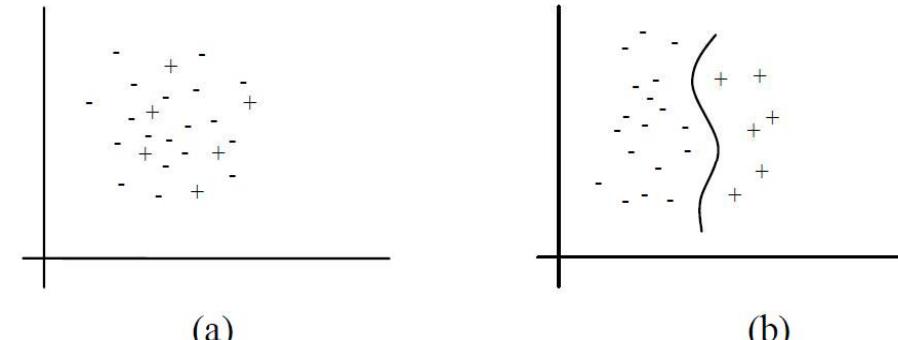
Density: Lack of data

Bordeline and Noise data

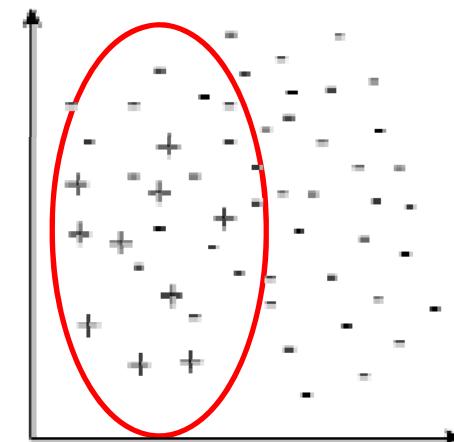
Dataset shift

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.**

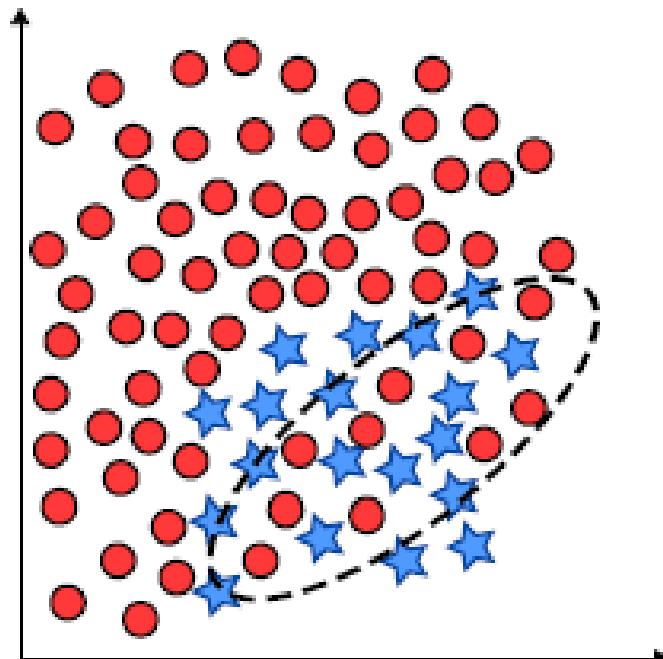


V. García, R.A. Mollineda, J.S. Sánchez. On the k-NN performance in a challenging scenario of imbalance and overlapping. Pattern Anal Applic (2008) 11: 269-280

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

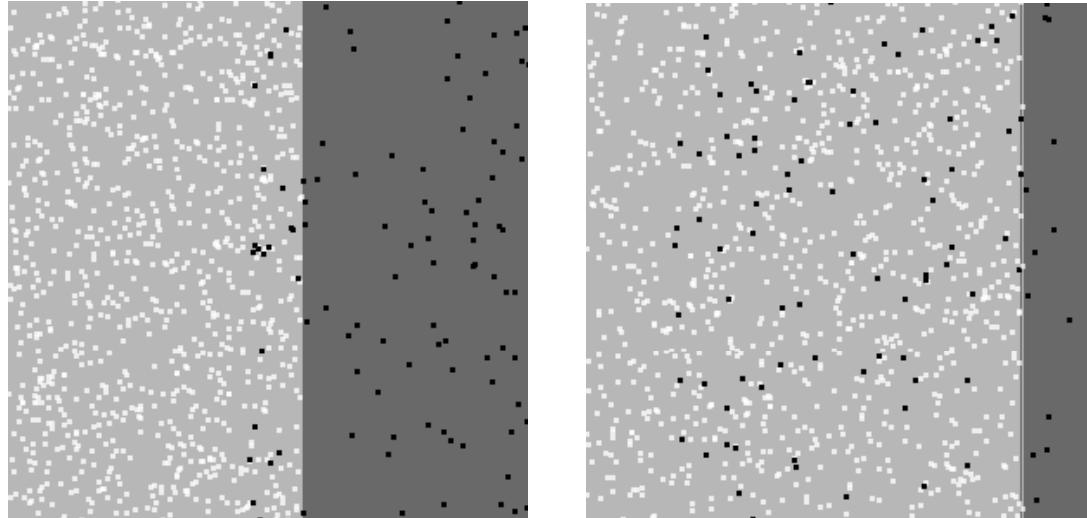
- There is an interesting relationship between imbalance and **class overlapping**:



(a) Class overlapping

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) 20% and (b) 80%

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

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

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

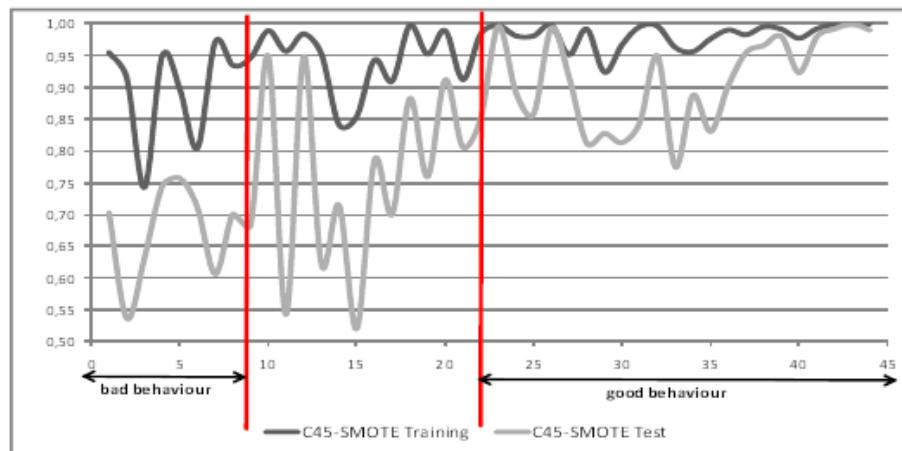


Fig. 6 C4.5 AUC with SMOTE in Training/Test sorted by F1

$F1 \leq 0.366$

$F1 \geq 1.469$

$F1$: maximum Fisher's discriminant ratio.

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

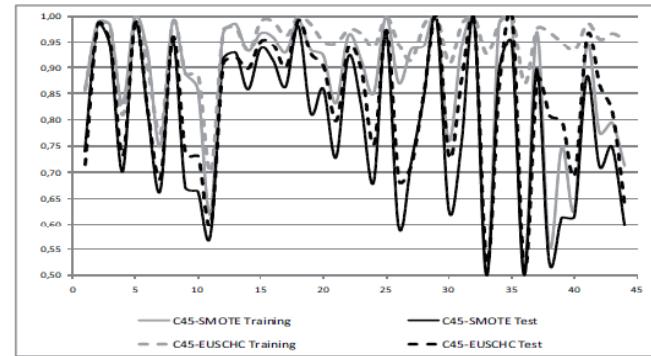


Fig. 4 C4.5 AUC in Training/Test sorted by IR

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

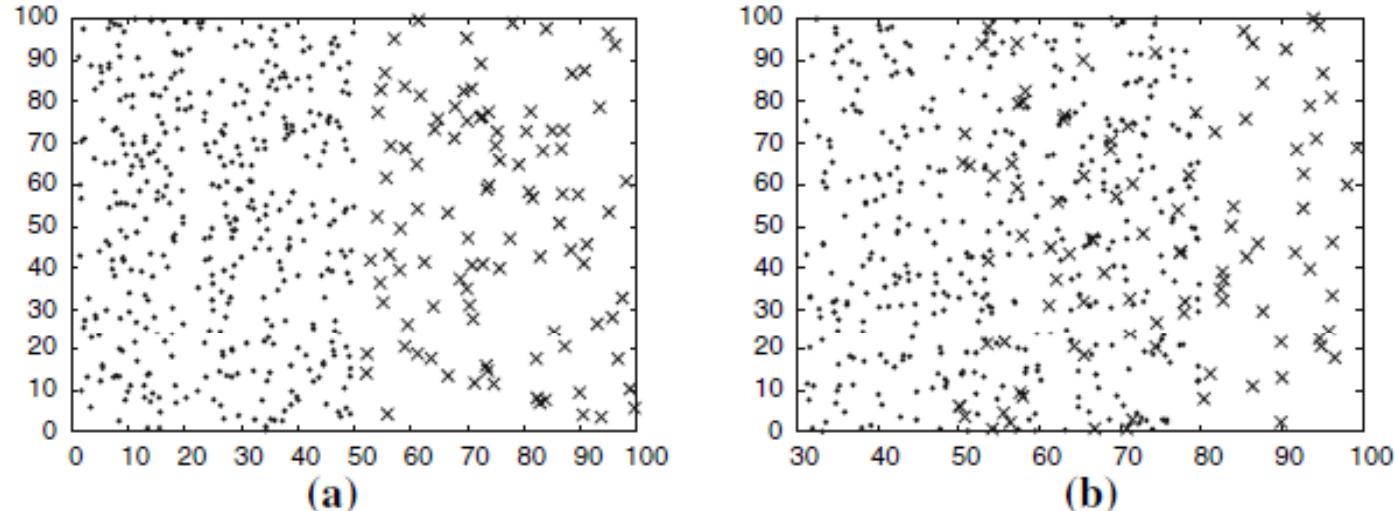


Fig. Two different levels of class overlapping: a 0% and b 60%

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

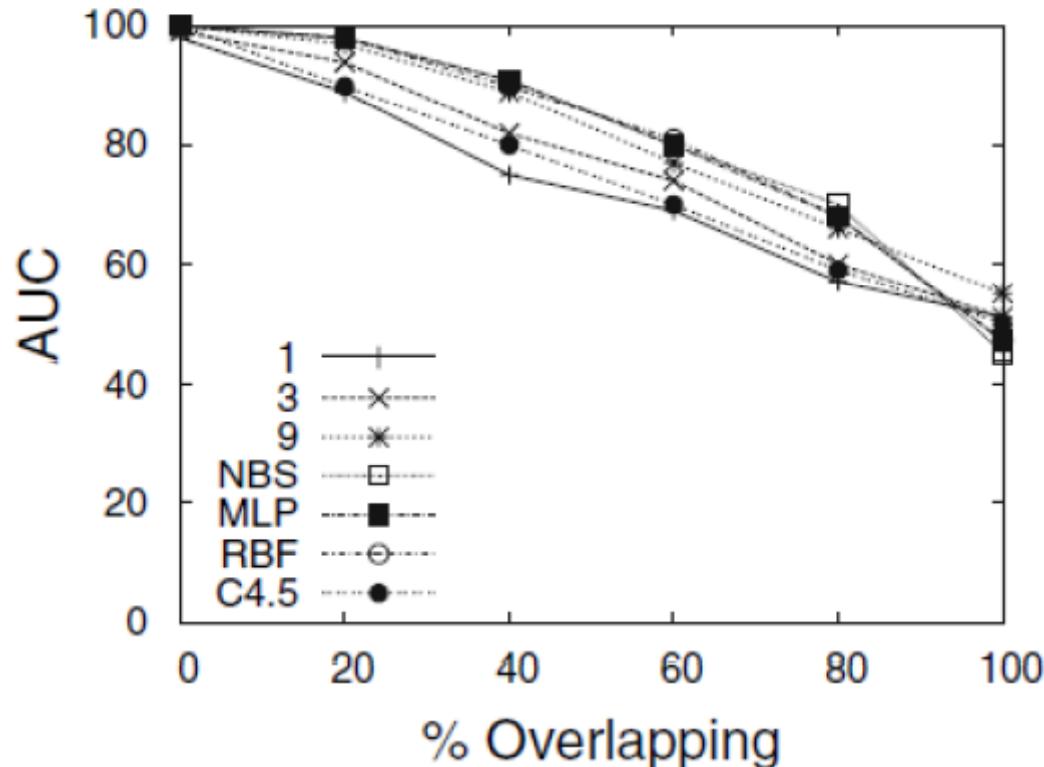
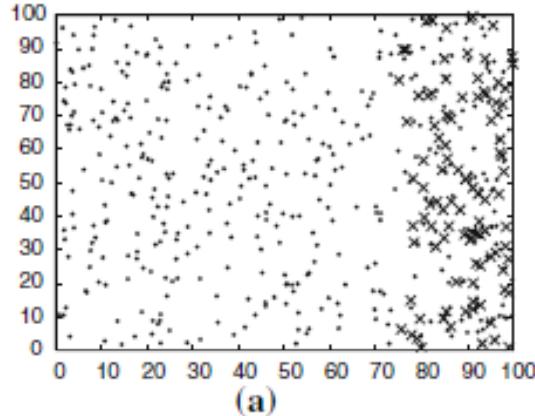


Fig. Performance metrics in k-NN rule and other learning algorithms for experiment I

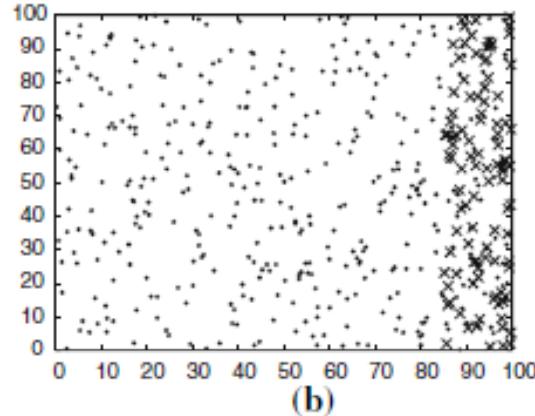
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping



(a)



(b)

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

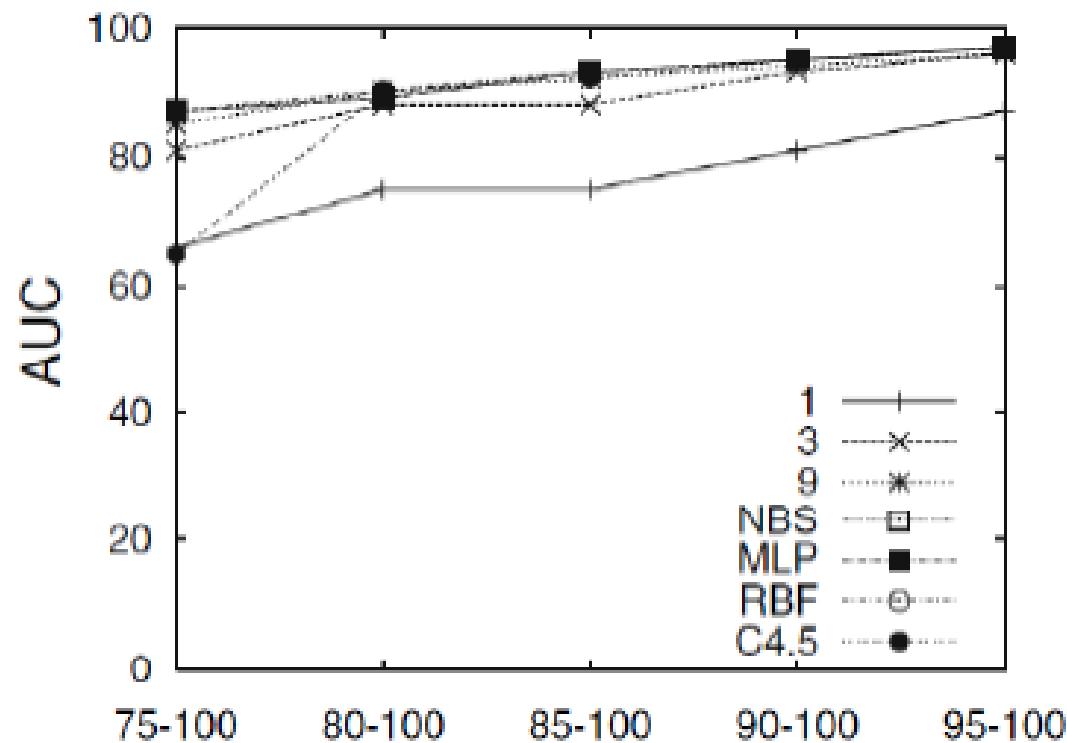


Fig. Performance metrics in k-NN rule and other learning algorithms for experiment II

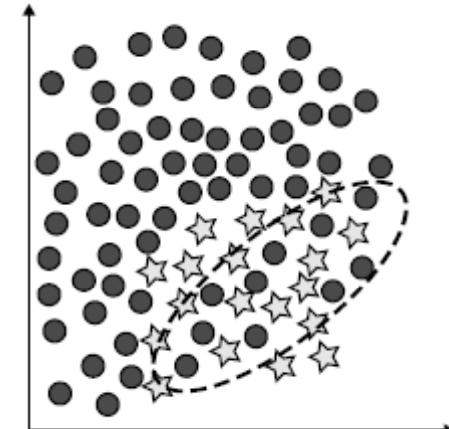
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

Conclusions: Results (in this paper) show that the class more represented in overlap 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.



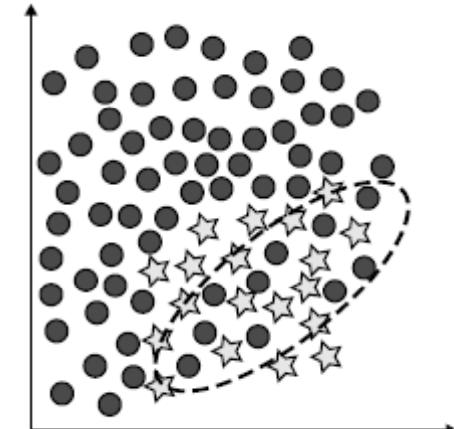
(a) Class overlapping

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

Open problem: To design new approaches (resampling or learning methods) to deal with the overlapping.



(a) Class overlapping

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

Small disjuncts/rare data sets

Density: Lack of data

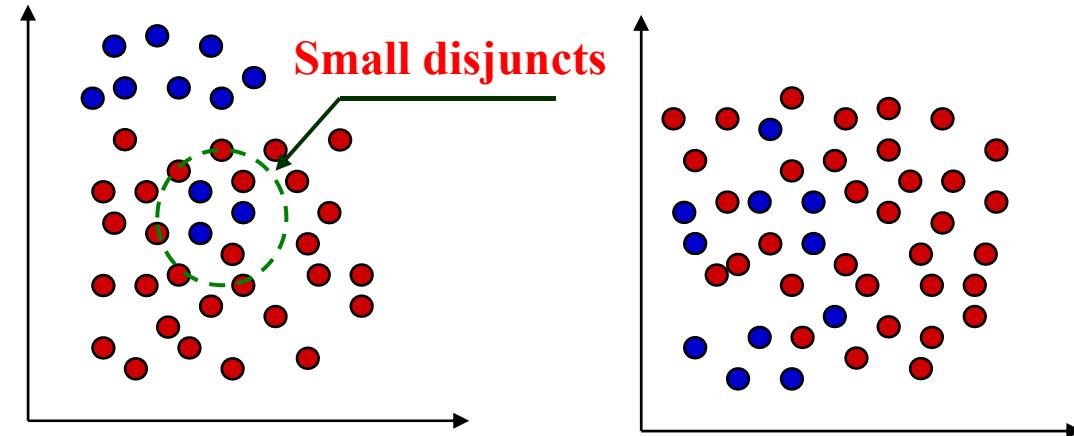
Bordeline and Noise data

Dataset shift

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.

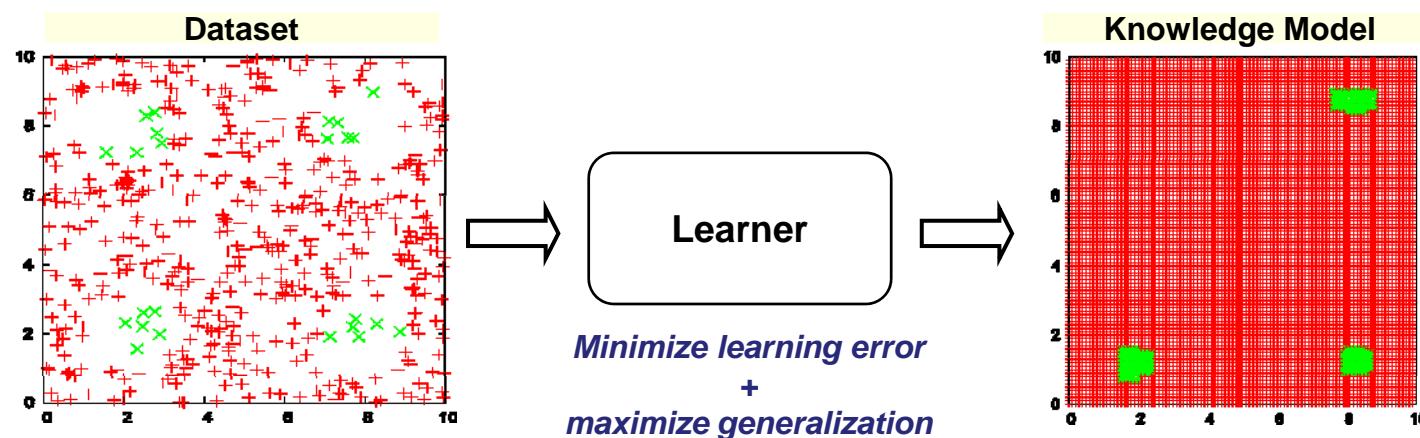
Rare cases or Small disjuncts are those disjuncts in the learned classifier that cover few training examples.

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.

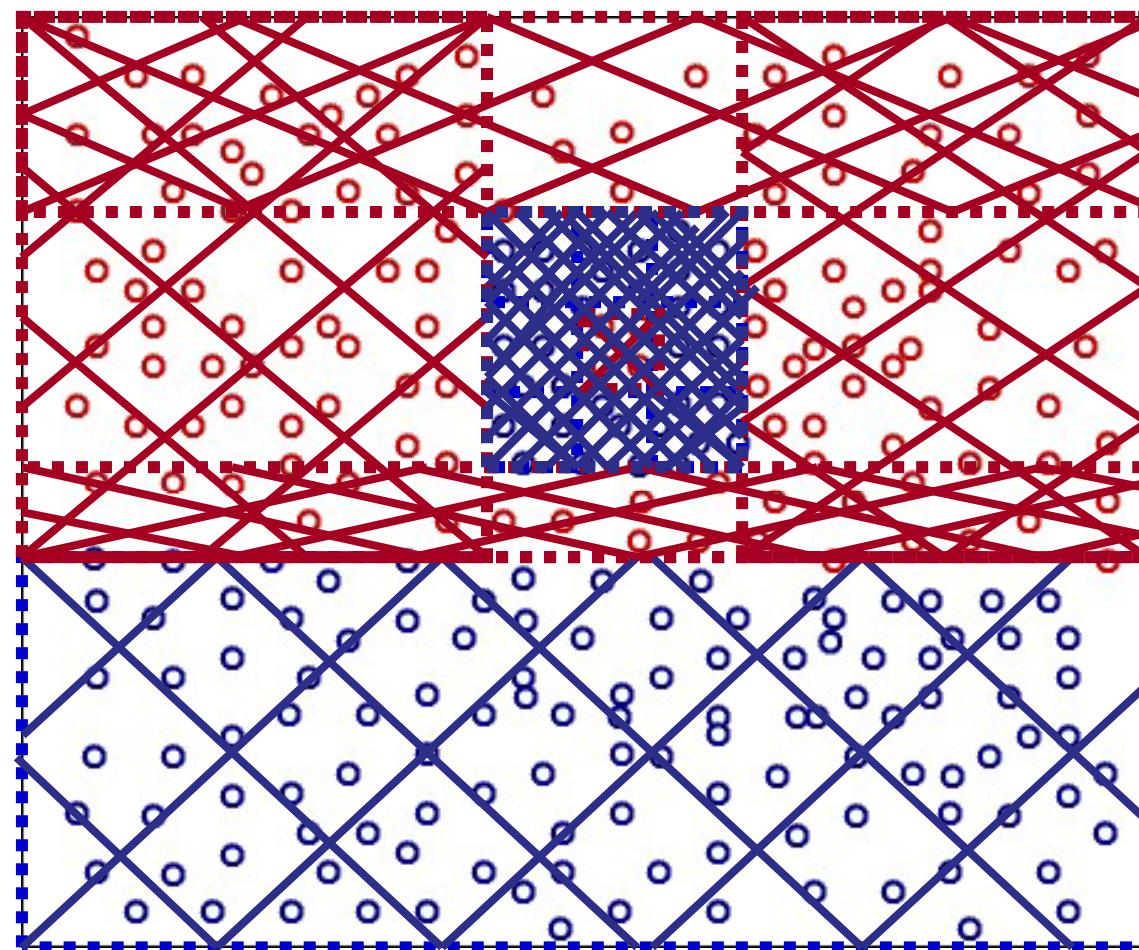


Why is difficult to learn in imbalanced domains?

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

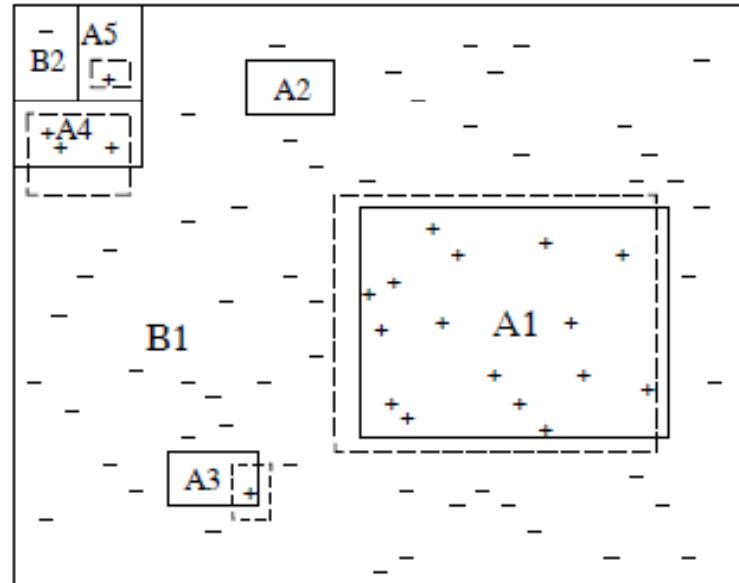


Figure 1: Graphical representation of a rare class and rare case

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.

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: CBO: Cluster-based oversampling: A method for inflating small disjuncts.

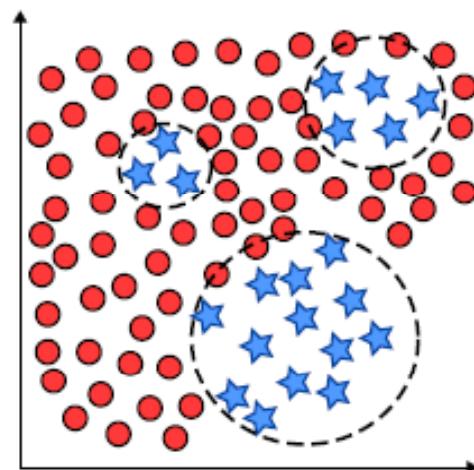
CBO method: Cluster-based resampling identifies rare cases and re-samples them individually, so as to avoid the creation of small disjuncts in the learned hypothesis.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Small disjuncts/Rare or excepcional cases

CBO method:

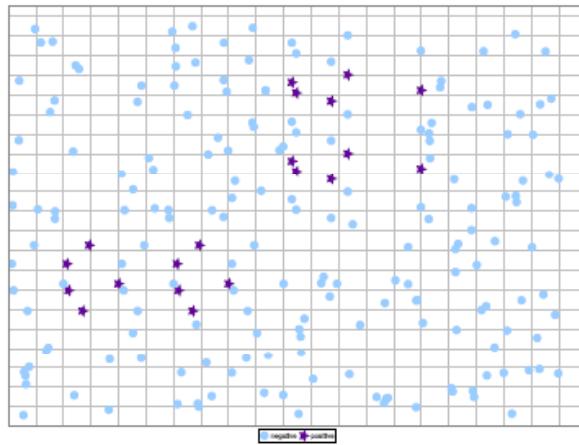


(b) Small disjuncts

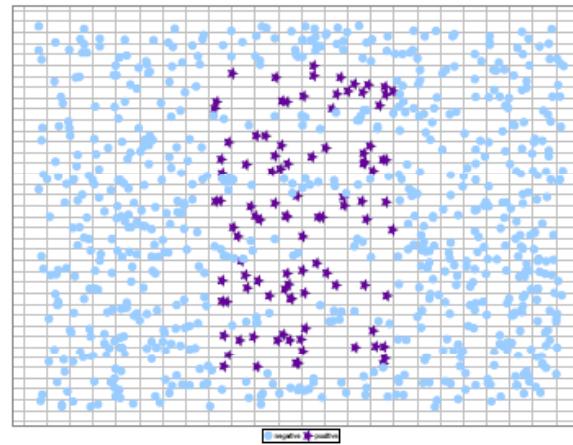
Once the training examples of each class have been clustered, oversampling starts. In the majority class, all the clusters, except for the largest one, are randomly oversampled so as to get the same number of training examples as the largest cluster. Let maxclasssize be the overall size of the large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

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

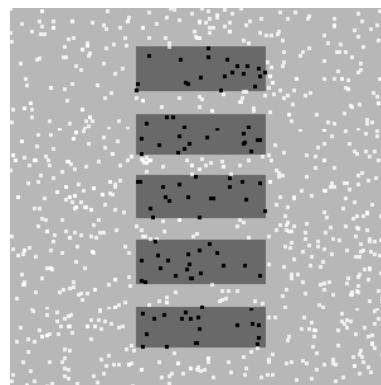
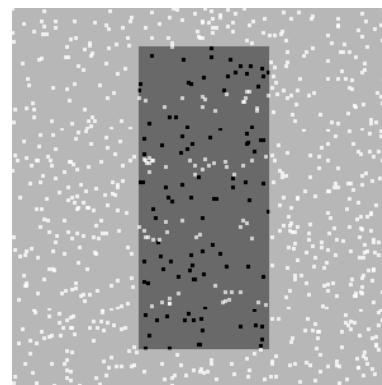
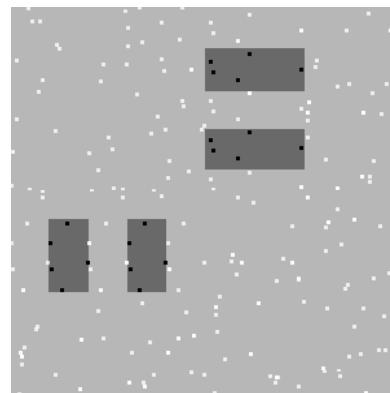
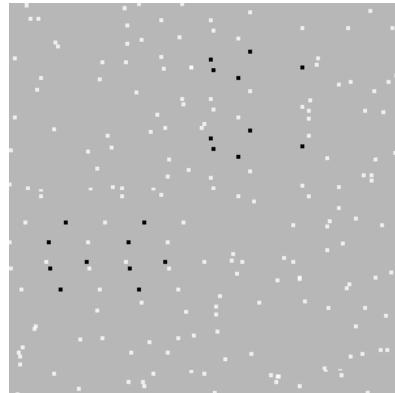
Table 12 Performance obtained by C4.5 in datasets suffering from small disjuncts

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

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Small disjuncts/Rare or excepcional cases



(a) Artificial dataset with the original data: 20 positive and 182 negative instances

(b) Artificial dataset with CBO: 228 positive and 228 negative instances

(c) Subclus dataset with the original data: 100 positive and 700 negative instances

(d) Subclus dataset with CBO: 780 positive and 780 negative instances

Figure: Boundaries obtained by C4.5 with the original and preprocessed data using CBO for addressing the problem of small disjuncts. The new instances for (b) and (d) are just replicates of the initial examples.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Small disjuncts/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.

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

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Density: Lack of data

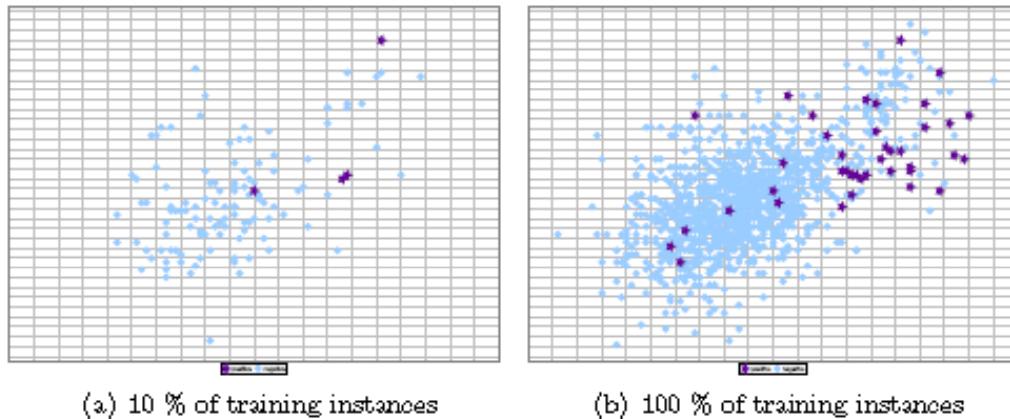


Figure 11: Lack of density or small sample size on the yeast4 dataset

The lack of density in the training data may also cause the introduction of small disjuncts.

It becomes very hard for the learning algorithm to obtain a model that is able to perform a good generalization when there is not enough data that represents the boundaries of the problem and, what it is also most significant, when the concentration of minority examples is so low that they can be simply treated as noise.

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

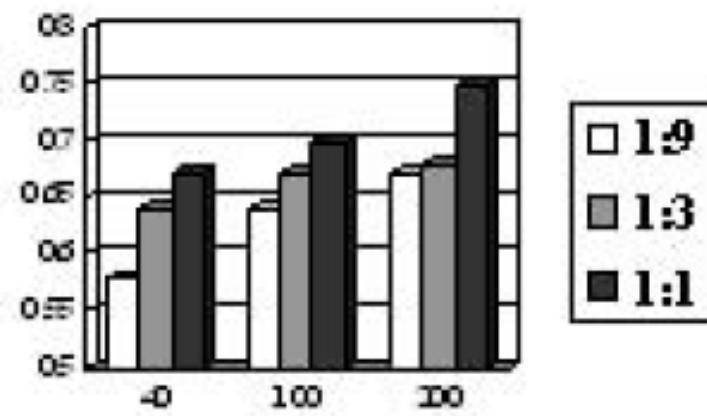
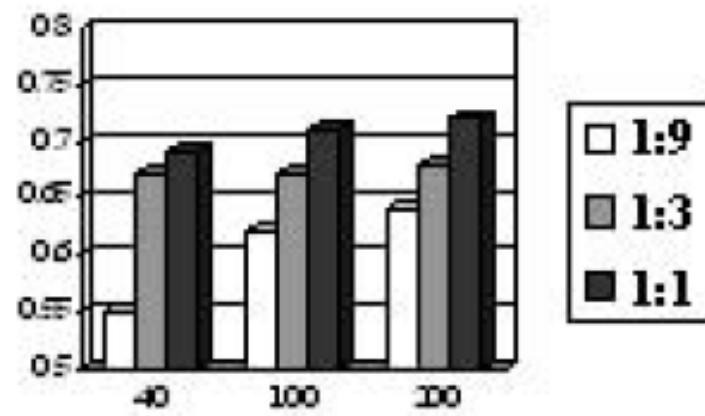
Experimental study with different levels of imbalance and density

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Density: Lack of data

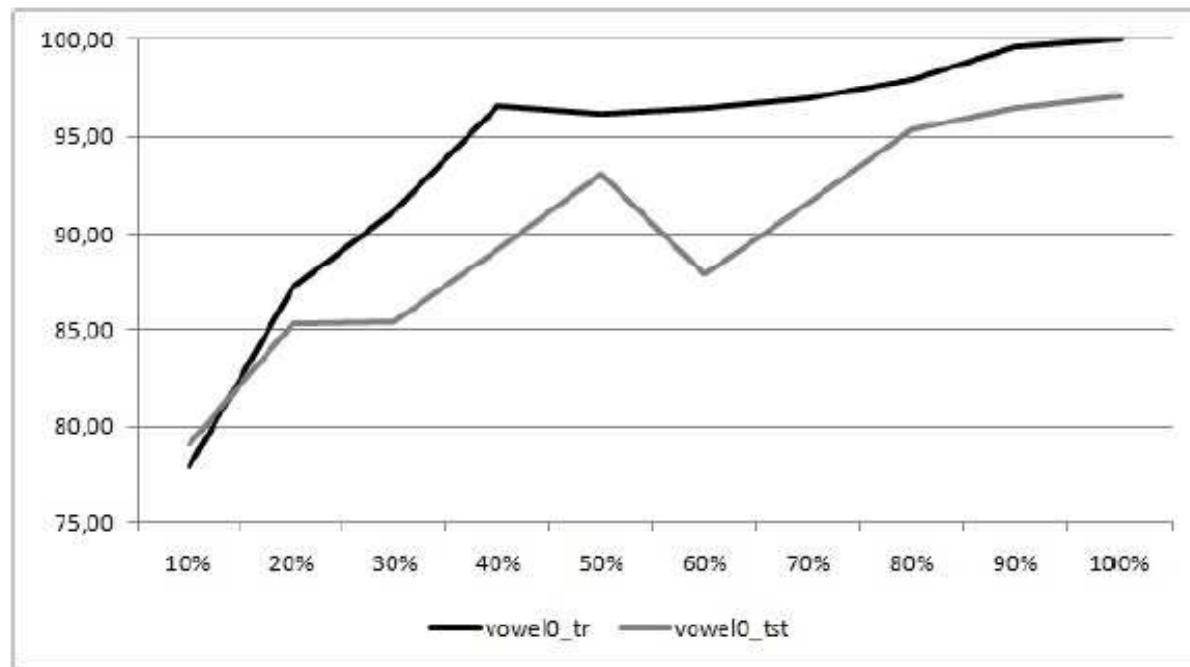
Left-C4.5, right-Backpropagation: These results show that the performance of classifiers, though hindered by class imbalances, is repaired as the training set size increases. This suggests that small disjuncts play a role in the performance loss of class imbalanced domains.



Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Density: Lack of data



From this graph, we may distinguish a growth rate directly proportional to the number of training instances that are being used. This behavior reflects the findings enumerated previously.

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

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

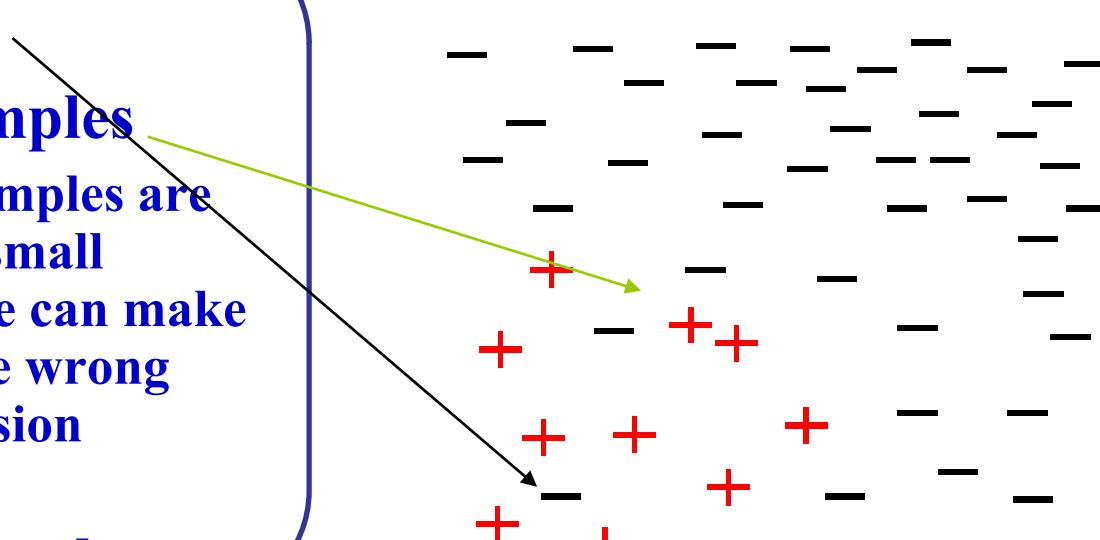
Borderline examples

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

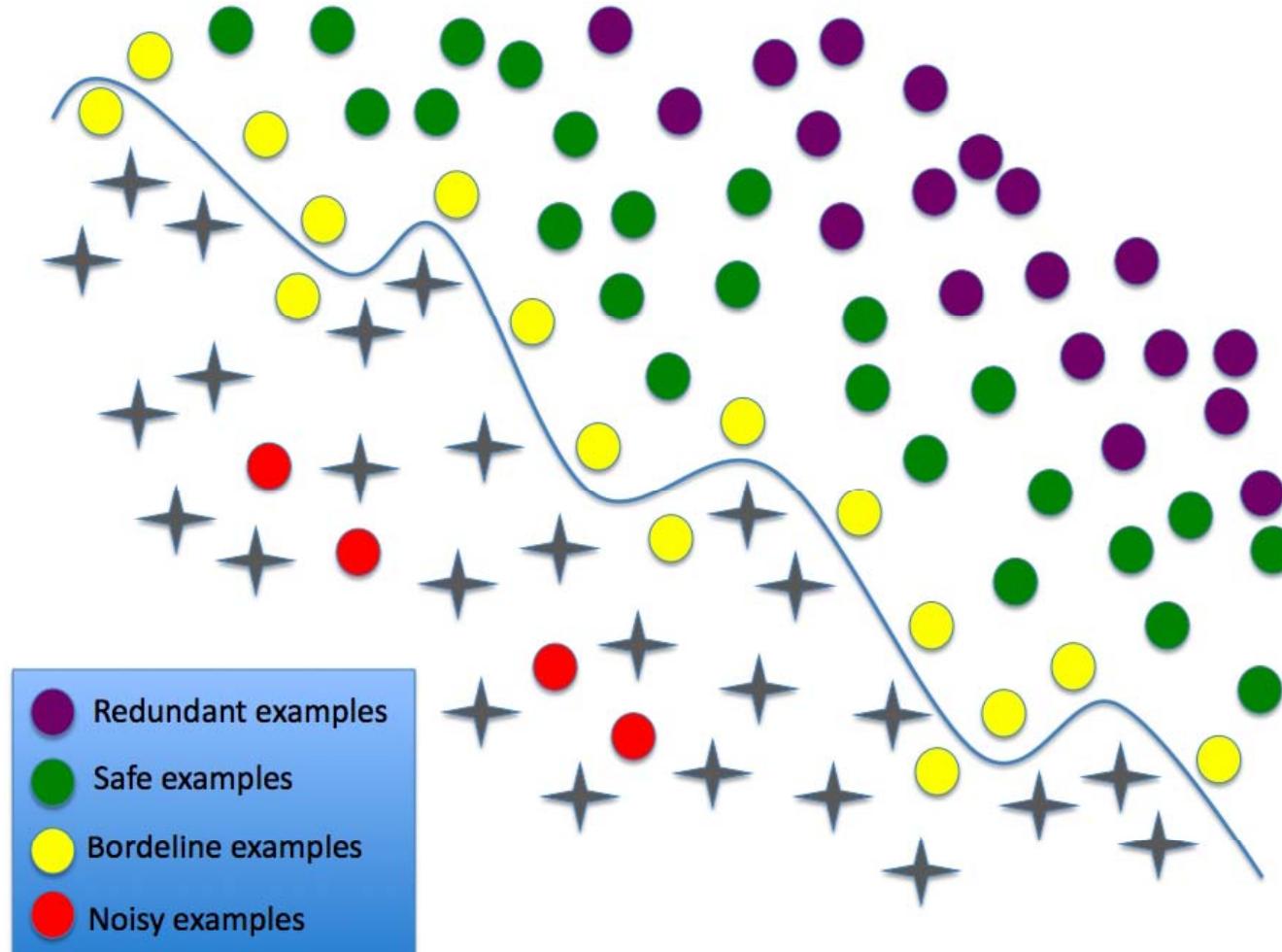
An approach: Detect and remove such majority noisy and borderline examples in filtering before inducing the classifier.



Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Borderline and Noise data



Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

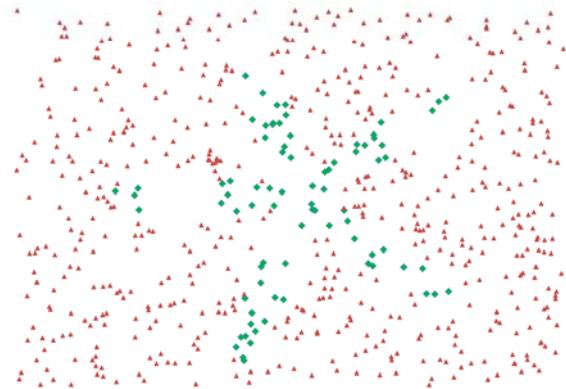
Borderline and Noise data

3 kind of artificial problems:

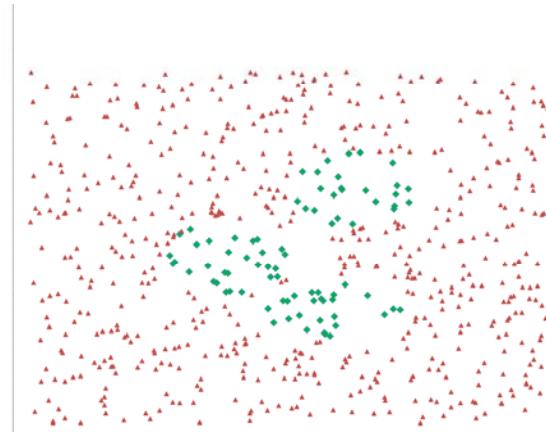
Subclus: examples from the minority class are located inside rectangles following related works on small disjuncts.

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

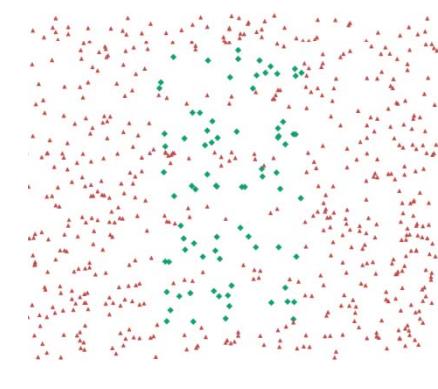
Paw: The minority class is decomposed into 3 elliptic sub-regions of varying cardinalities, where two subregions are located close to each other, and the remaining smaller sub-region is separated.



Clover data



Paw data

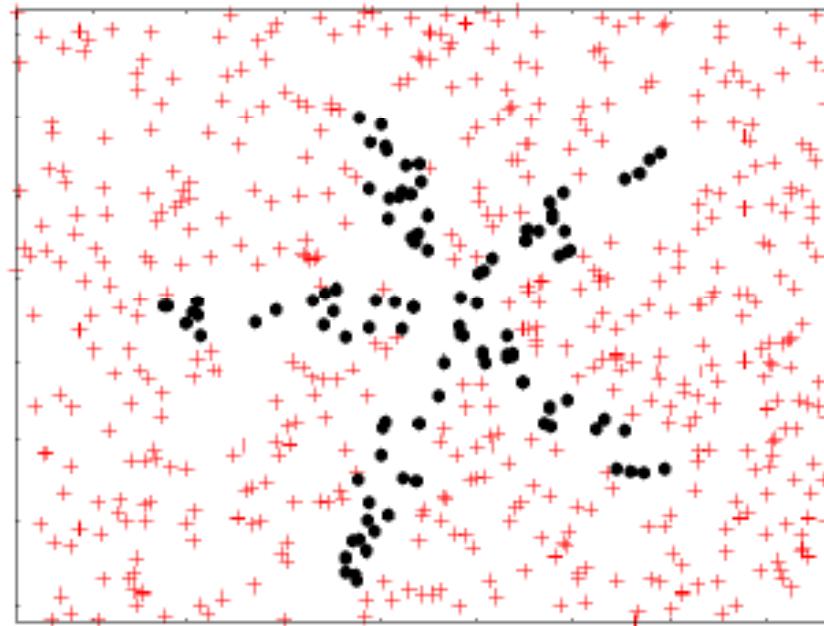


Subclus data

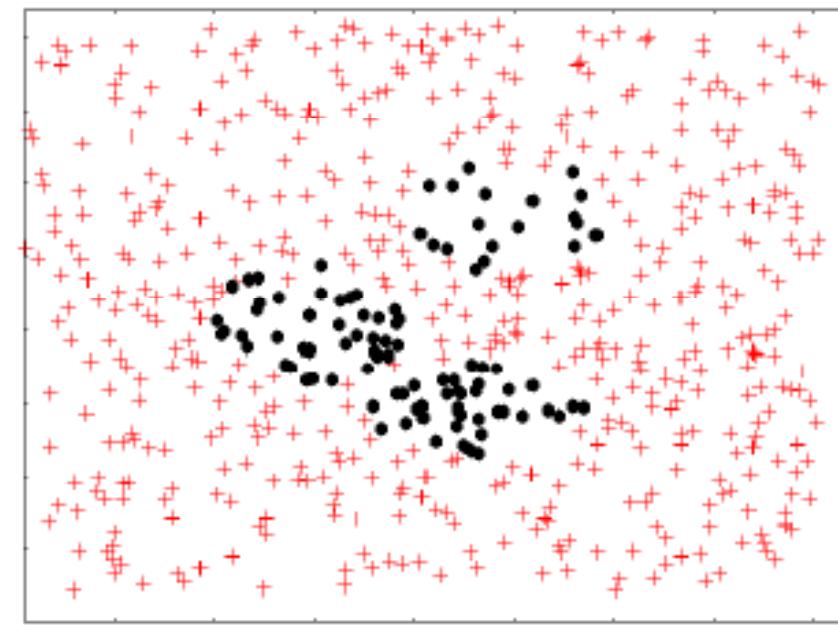
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Borderline and Noise data



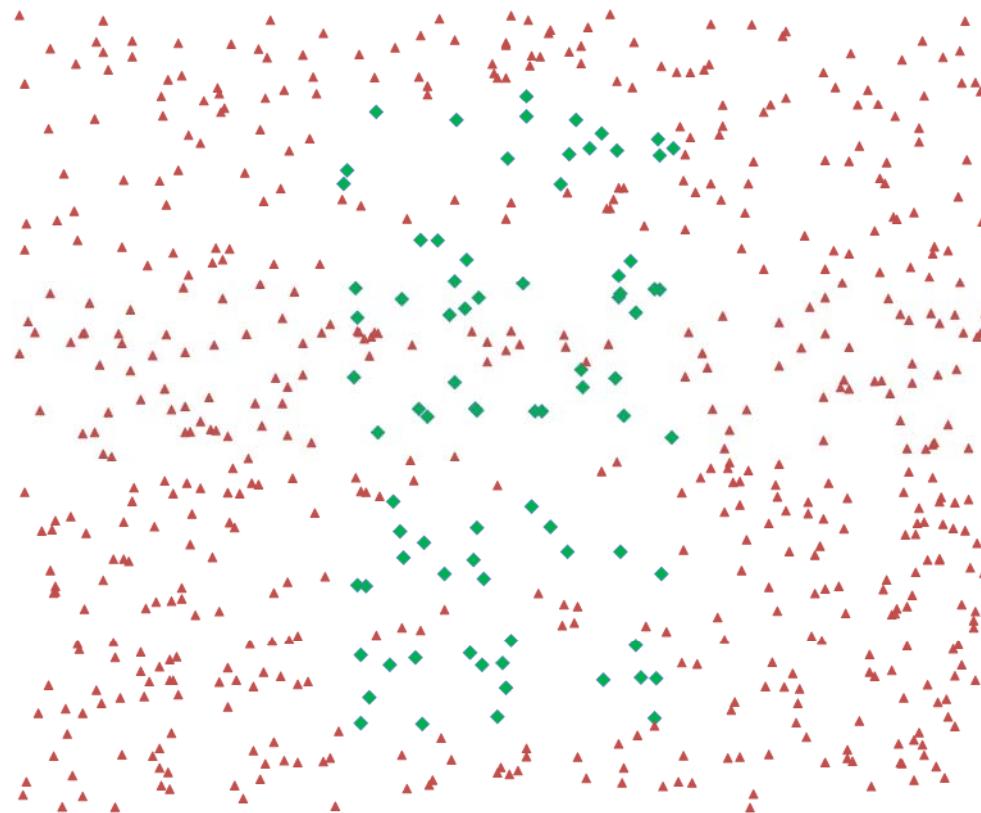
Clover data



Paw data

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

Borderline and Noise data

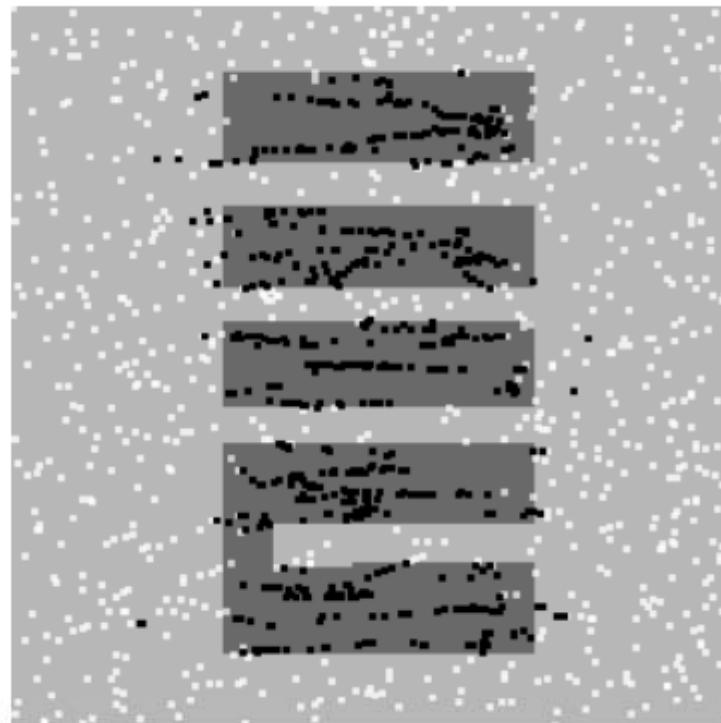


Subclus data

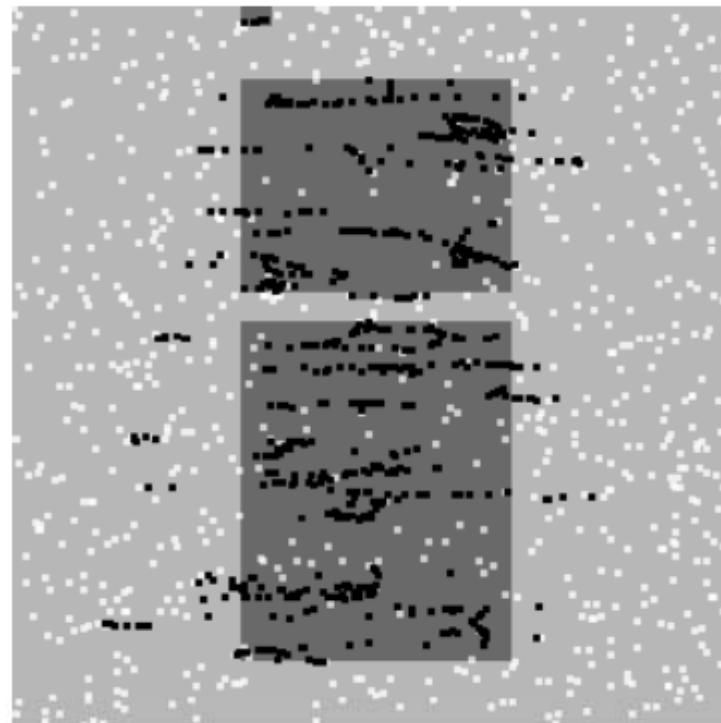
Why is difficult to learn in imbalanced domains? Intrinsic data characteristics

Borderline and Noise data

Subclus data



(a) Original problem and decision functions



(b) Noisy instances and new undesirable
decision functions
20% gaussian noise

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

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Borderline 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

Borderline and Noise data

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

Borderline and Noise data

Small disjunct and Noise data

Borderline and Noise data

Borderline 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

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

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Overlapping

Small disjuncts/rare data sets

Density: Lack of data

Borderline and Noise data

Dataset shift

**Three
connected
problems**

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 rarity.

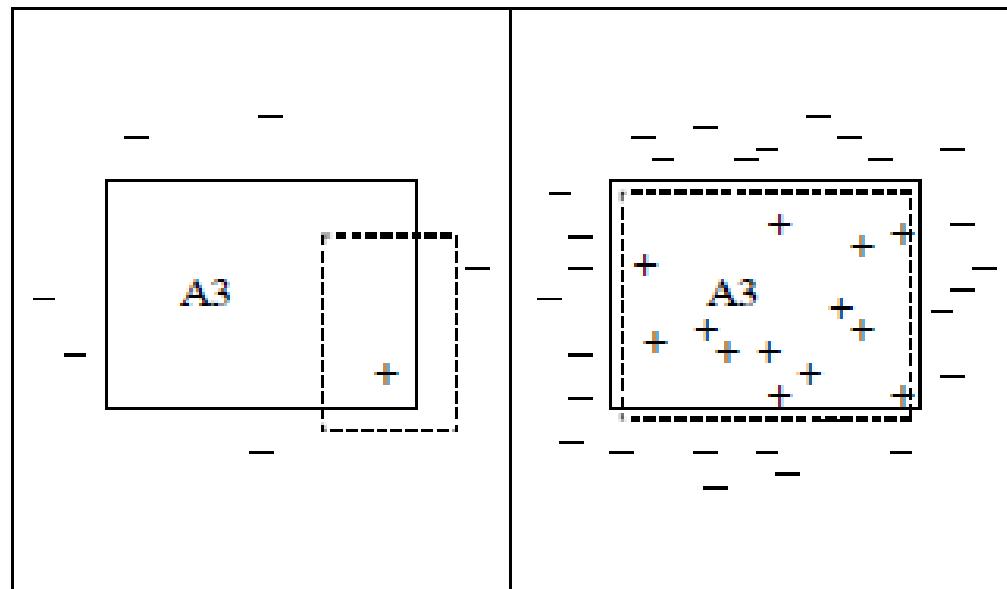


Figure 2: The impact of an “absolute” lack of data

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.

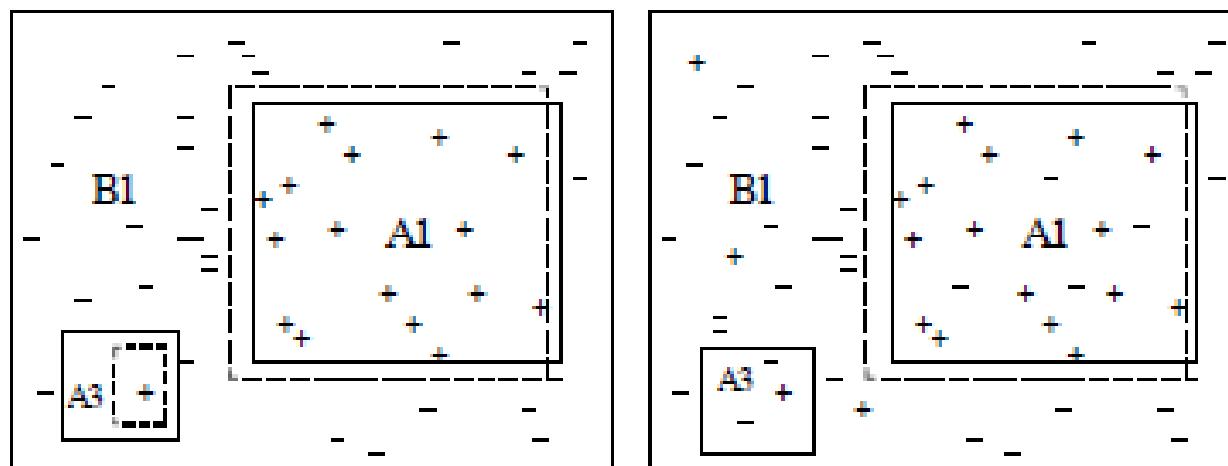


Figure 3: The effect of noise on rare cases

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

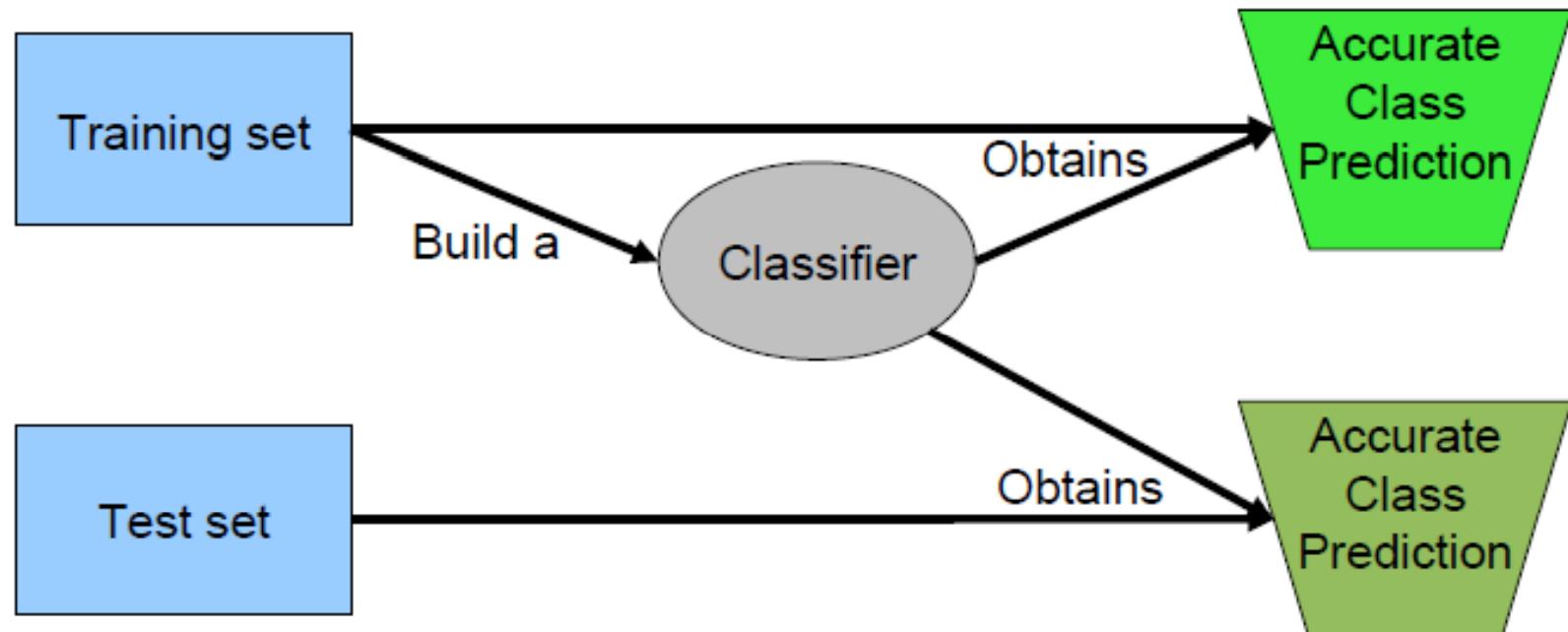
Dataset shift

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

- Basic assumption in classification:

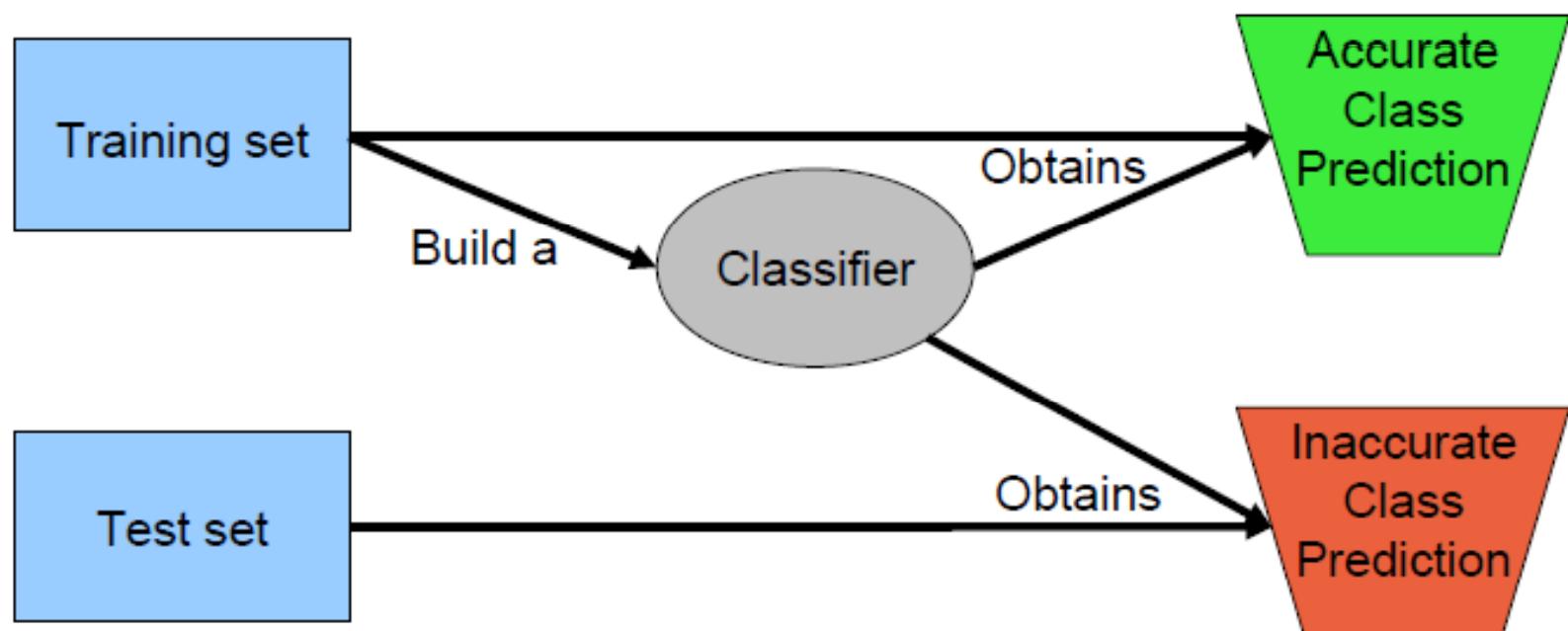


Why is difficult to learn in imbalanced domains?

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)?.

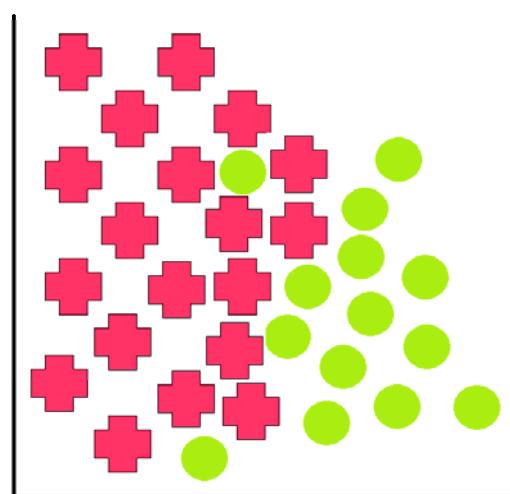
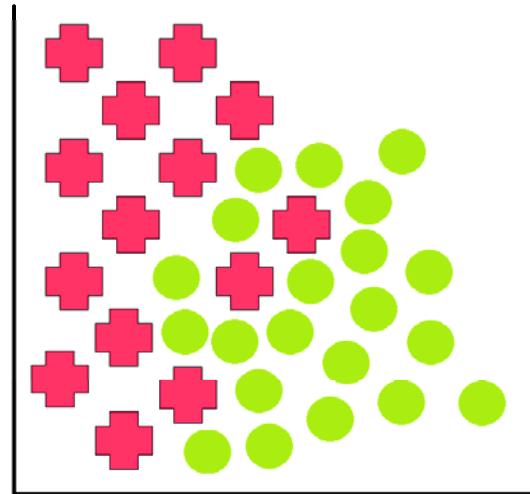


The Problem of Dataset Shift

- **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

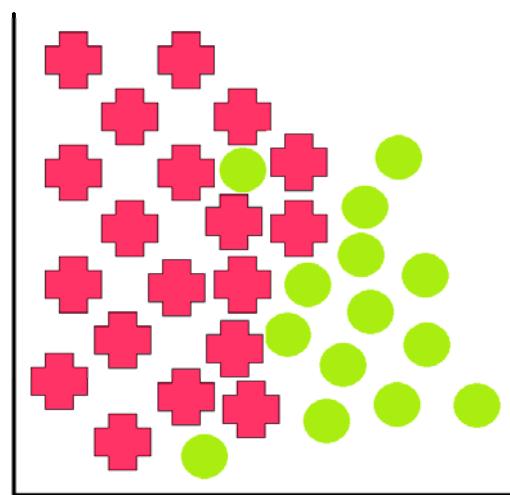
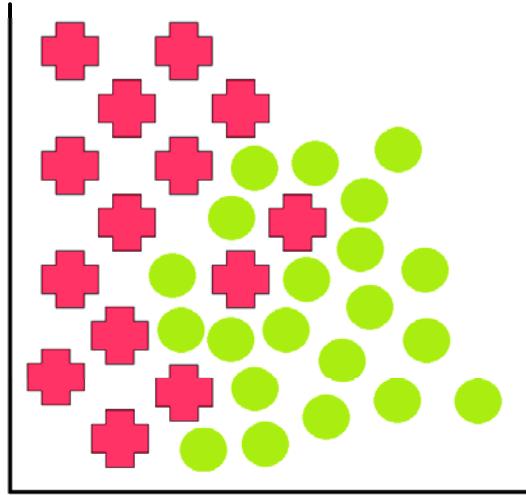
The problem of data-set shift is defined as the case where training and test data follow different distributions.



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.

The Problem of Dataset Shift

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



Covariate Shift: The inputs of the problem differ in training and test sets.

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, due to the minority class examples.**

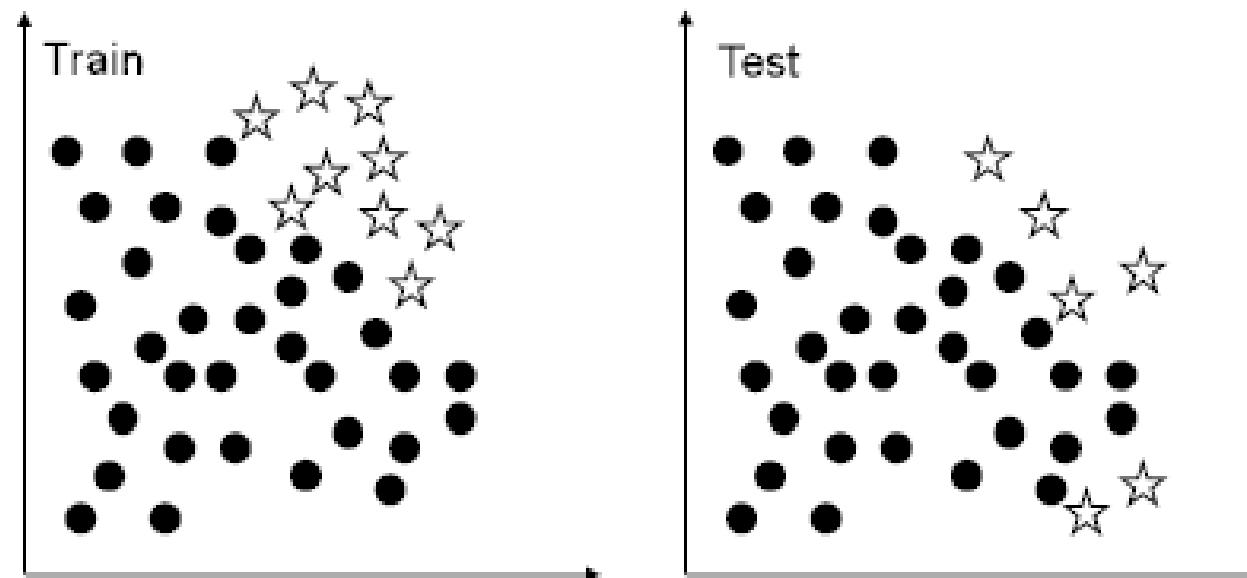


Figure 18: Example of the impact of data-set shift in imbalanced domains.

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:

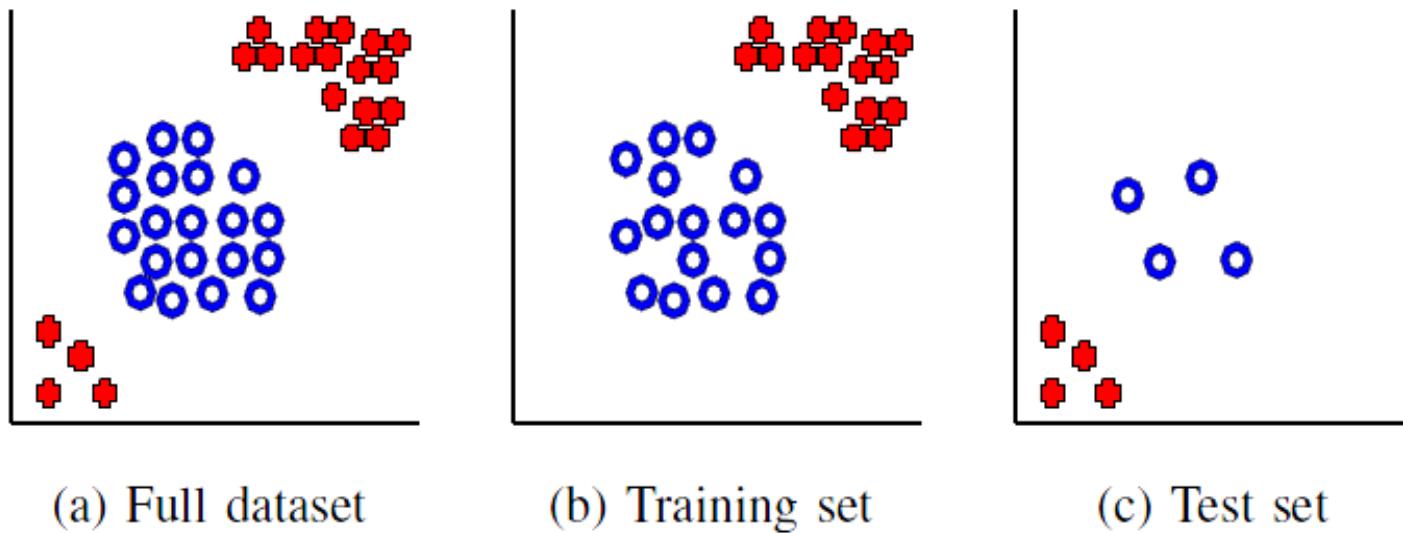


Fig. 1: Extreme example of partition-based covariate shift. Note how the examples on the bottom left of the “cross” class will be wrongly classified due to covariate shift.

Causes of Dataset Shift

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 all datasets within an iteration.**

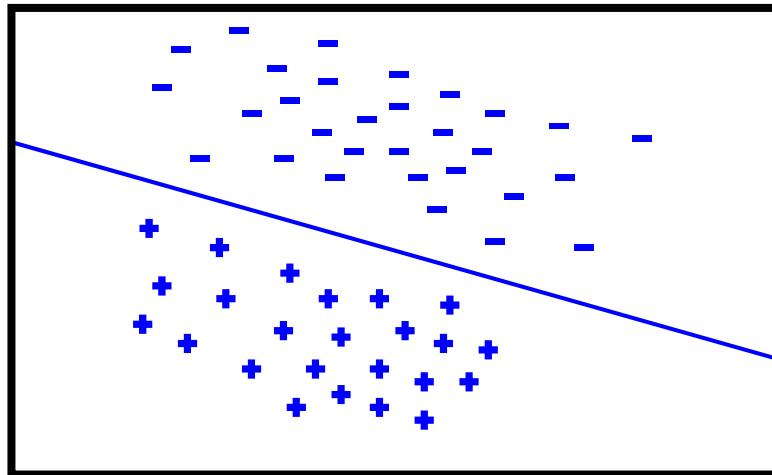
[T. Raeder, T. R. Hoens, and N. V. Chawla, “Consequences of variability in classifier performance estimates,” Proceedings of the 2010 IEEE International Conference on Data Mining, 2010, pp. 421–430.](#)

Wilcoxon test: Clear differences for both algorithms

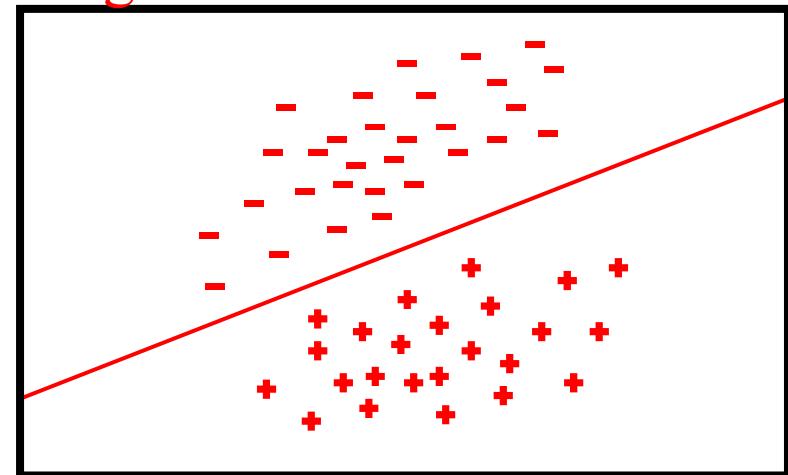
Causes of Dataset Shift

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

source domain



target domain

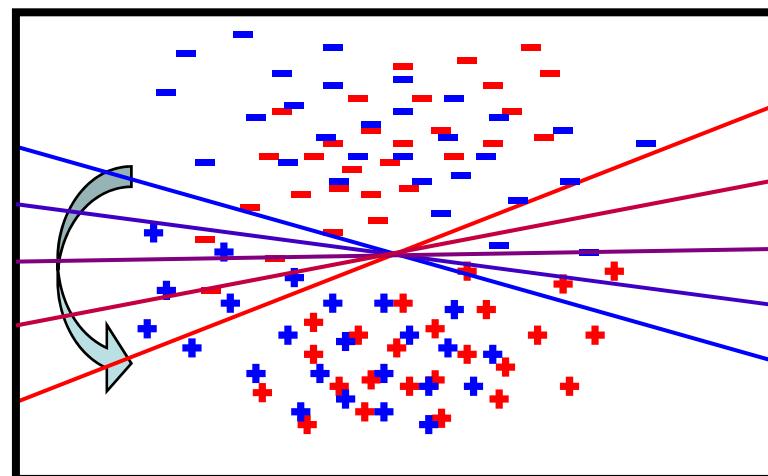


Causes of Dataset Shift

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

source domain

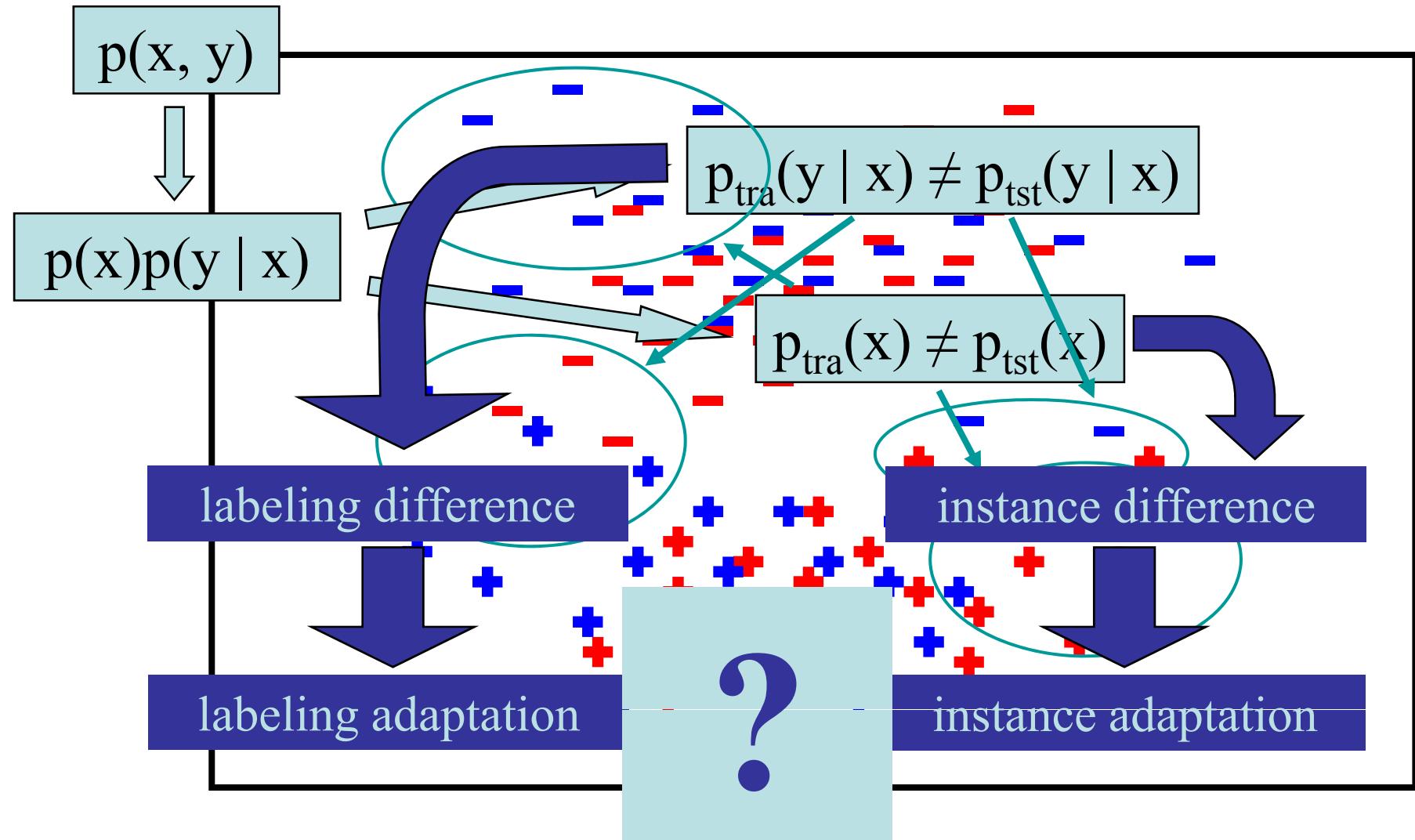
target domain



Causes of Dataset Shift

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

Where Does the Difference Come from?



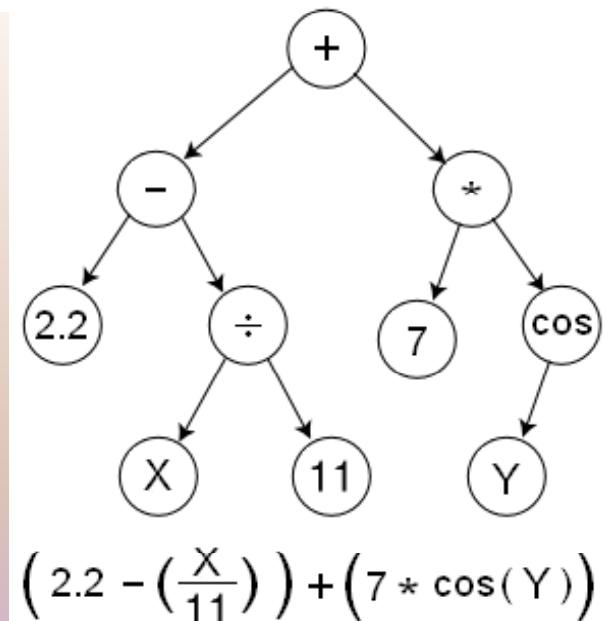
Why is difficult to learn in imbalanced domains?

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.

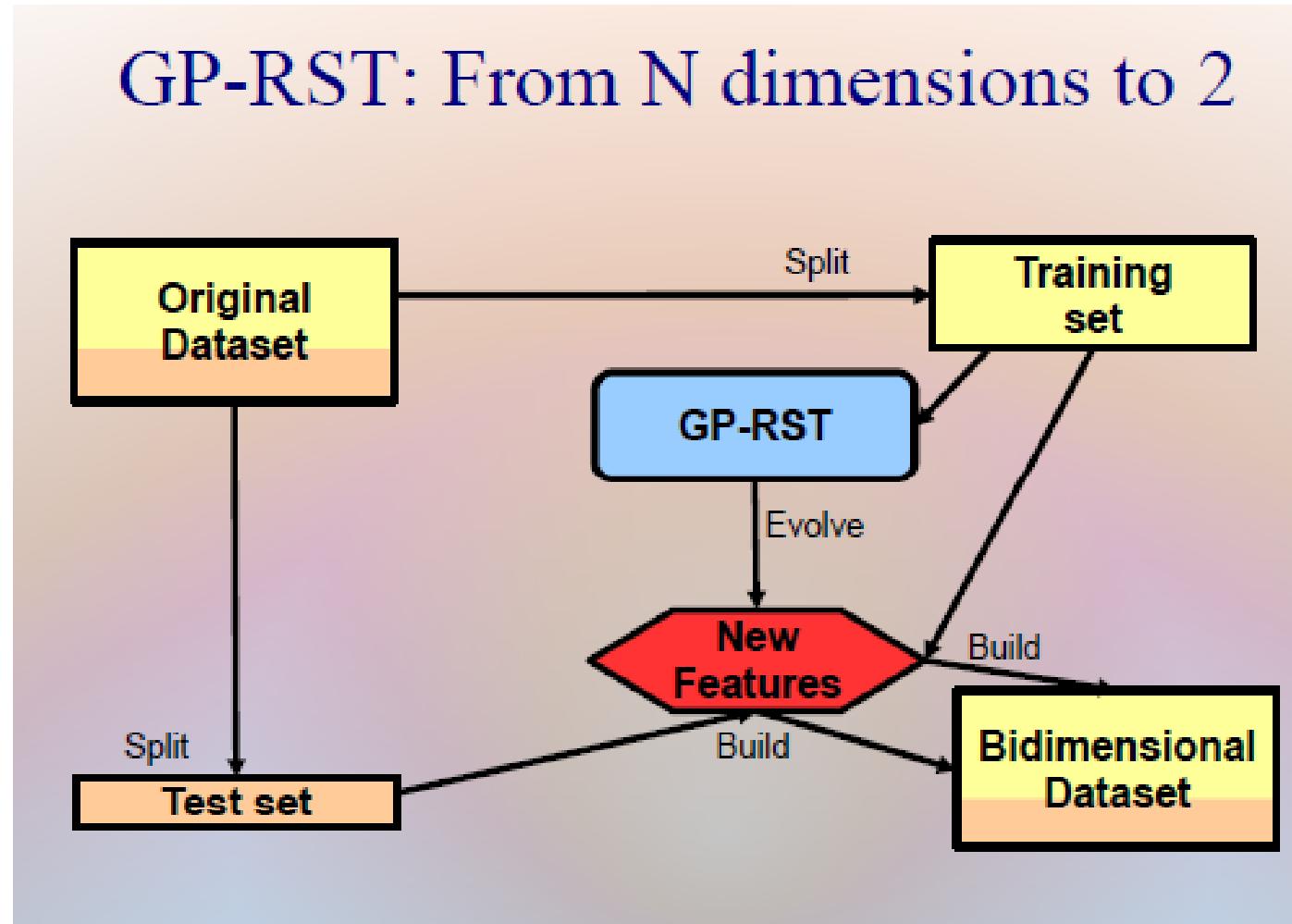


Moreno-Torres, J. G., & Herrera, F. (2010). A preliminary study on overlapping and data fracture in imbalanced domains by means of genetic programming-based feature extraction. In *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA 2010)* (pp. 501–506).

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

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

rough set that belong to its lower approximation.

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

Algorithm 1 Fitness evaluation procedure

1. Obtain $E' = \{e'^h = (f_1(e^h), f_2(e^h), C^h) / h = 1, \dots, n_e\}$, 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 $C_i \in C : i = 1, \dots, n_c$,
 - 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.

Example of good behavior

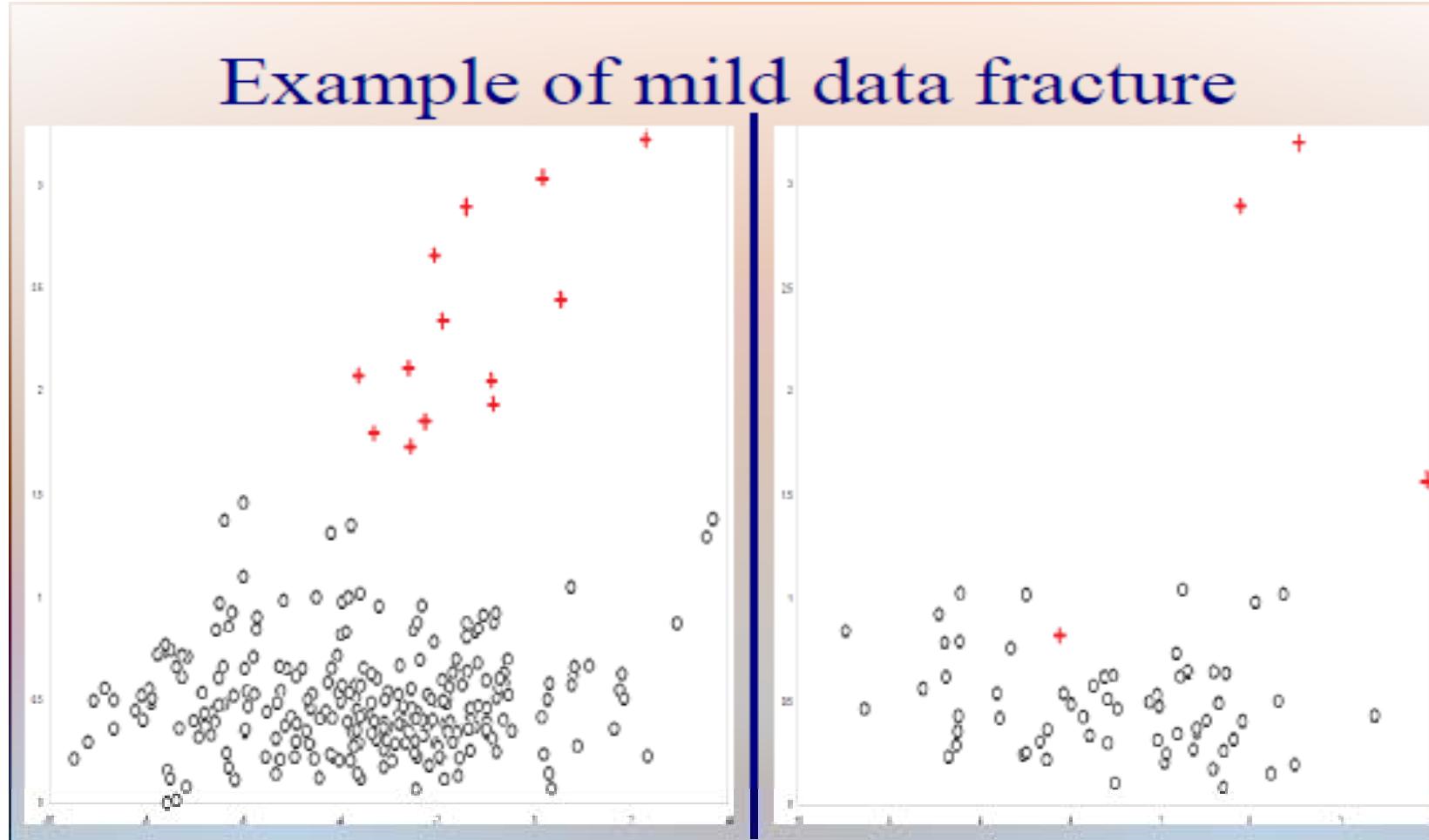


(a) Training set (1.0000)

(b) Test set (1.0000)

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

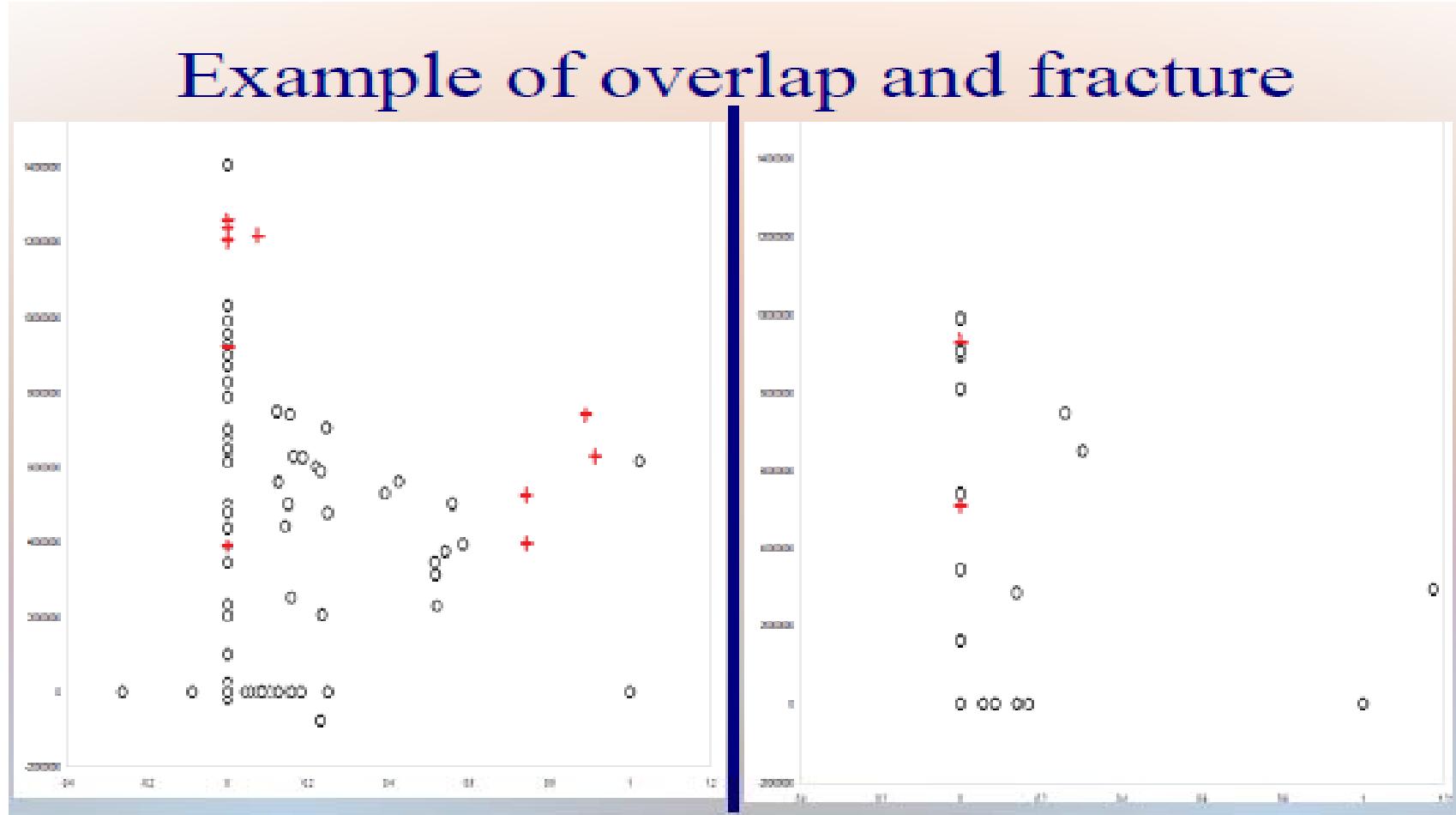
Dataset shift. ecoli 4, 1st partition.



(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.

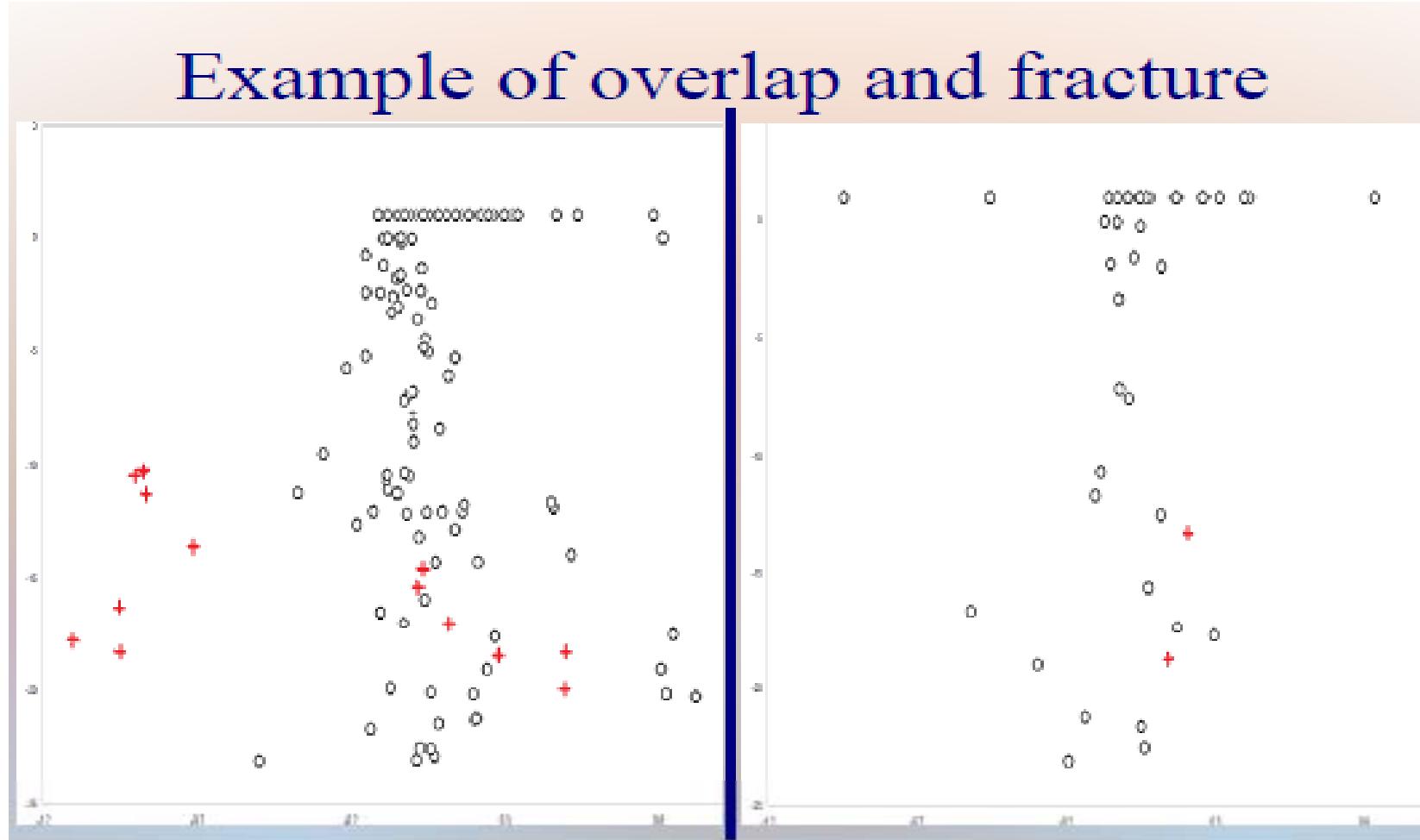


(a) Training set (0.3779)

(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 dataset shift in imbalanced domains.

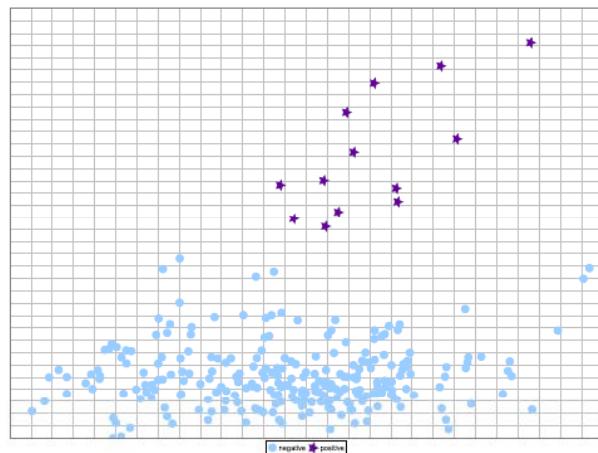
- The first one focuses on intrinsic dataset shift, that is, the data of interest includes some degree of shift that is producing a relevant drop in performance. In this case, we need to:
 - Develop techniques to discover and measure the presence of data-set shift adapting them to minority classes.
 - 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

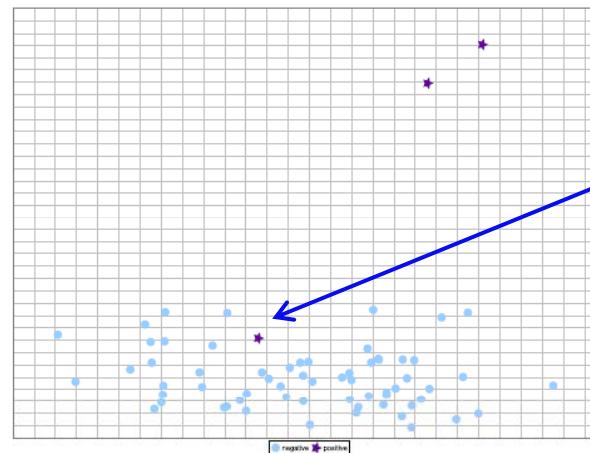
Dataset shift

- 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) Training data. AUC = 1.000

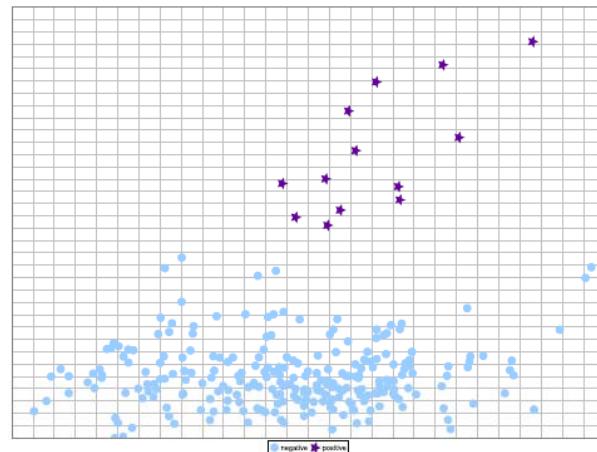


(b) Test data. AUC = .8750

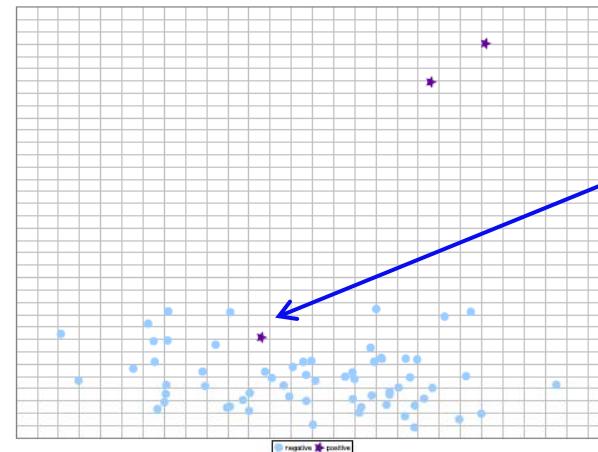
Example:
Training and
test data follow
different
distributions.

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

Dataset shift



(a) Training data. AUC = 1.000



(b) Test data. AUC = .8750

Example:
Training and
test data follow
different
distributions.

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

Proposed solution: DOB-SCV

A more sophisticated technique, known as **DOB-SCV**, is considered:

- **Assigning close-by examples to different folds, so that representative examples for the different regions of the problem will be represented among them.**

Algorithm 1 DOB-SCV Partitioning Method

```
for each class  $c_j \in C$  do
    while  $\text{count}(c_j) > 0$  do
         $e_0 \leftarrow$  randomly select an example of class  $c_j$  from  $D$ 
         $e_i \leftarrow$   $i$ th closest example to  $e_0$  of class  $c_j$  from  $D$  ( $i = 1, \dots, k - 1$ )
         $F_i \leftarrow F_i \cup e_i$  ( $i = 0, \dots, k - 1$ )
         $D \leftarrow D \setminus e_i$  ( $i = 0, \dots, k - 1$ )
    end while
end for
```

J. G. Moreno-Torres, J. A. Sáez, and F. Herrera, “Study on the impact of partition-induced dataset shift on k-fold cross-validation,” IEEE Transactions On Neural Networks And Learning Systems, vol. 23, no. 8, pp. 1304–1313, 2012.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

Proposed solution: DOB-SCV

This technique aims at carrying out an heterogeneous organization of the instances of the classes among the different folds.

Algorithm 1 DOB-SCV Partitioning Method

```
for each class  $c_j \in C$  do
    while  $\text{count}(c_j) > 0$  do
         $e_0 \leftarrow$  randomly select an example of class  $c_j$  from  $D$ 
         $e_i \leftarrow$   $i$ th closest example to  $e_0$  of class  $c_j$  from  $D$  ( $i = 1, \dots, k - 1$ )
         $F_i \leftarrow F_i \cup e_i$  ( $i = 0, \dots, k - 1$ )
         $D \leftarrow D \setminus e_i$  ( $i = 0, \dots, k - 1$ )
    end while
end for
```

J. G. Moreno-Torres, J. A. Sáez, and F. Herrera, "Study on the impact of partition-induced dataset shift on k-fold cross-validation," IEEE Transactions On Neural Networks And Learning Systems, vol. 23, no. 8, pp. 1304–1313, 2012.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

Proposed solution: DOB-SCV

Table 3: Average test results with AUC metric and percentage differences for the SCV and DOB-SCV techniques.

Algorithm	IR < 9			IR > 9			All		
	SCV	DOB-SCV	% Diff	SCV	DOB-SCV	% Diff	SCV	DOB-SCV	% Diff
C4.5	.8597 ± .0357	.8698 ± .0393	1.28	.8133 ± .0844	.8309 ± .0751	2.83	.8288 ± .0681	.8439 ± .0632	2.32
Chi	.8151 ± .0352	.8187 ± .0380	0.51	.7698 ± .1041	.7781 ± .0909	1.24	.7849 ± .0811	.7916 ± .0733	1.00
k-NN	.8478 ± .0342	.8616 ± .0340	1.96	.8272 ± .0937	.8395 ± .0855	1.74	.8341 ± .0739	.8468 ± .0683	1.81
SMO	.8573 ± .0317	.8644 ± .0253	0.96	.8425 ± .0695	.8427 ± .0606	0.23	.8474 ± .0569	.8500 ± .0488	0.47
PDFC	.8877 ± .0293	.8901 ± .0263	0.34	.8608 ± .0819	.8672 ± .0708	0.86	.8698 ± .0644	.8749 ± .0560	0.69

Comparison	R ⁺	R ⁻	p-value
C4.5[DOB-SCV] vs C4.5[SCV]	1391	754	0.0371
Chi[DOB-SCV] vs Chi[SCV]	1411	734	0.0267
k-NN[DOB-SCV] vs k-NN[SCV]	1536	609	0.0024
SMO[DOB-SCV] vs SMO[SCV]	1395	816	0.0639
PDFC[DOB-SCV] vs PDFC[SCV]	1366	845	0.0955

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

Proposed solution: DOB-SCV

DOB-SCV validation technique achieves a higher estimation of the performance for most datasets. It is more robust for analyzing the quality of the models learned in imbalanced data.

- The higher the IR is, the greater the differences between the DOB-SCV and the standard SCV.
- The lower the number of positive instances, the more significant is to maintain the data distribution to avoid the gap in performance between training and test.

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.
- ❑ Dataset shift 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 provoke additional problems.

- The data partition provokes data fracture: Dataset shift.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

What domain characteristics aggravate the problem?

There is a current need to study the aforementioned intrinsic characteristics of the data.

So that future research on classification with imbalanced data should focus on detecting and measuring the most significant data properties, in order to be able to define good solutions as well as alternatives to overcome the problems.

Contents

SESSION 3:

VI. Ensembles to address class imbalance

VII. Multiple class imbalanced data-sets: A pairwise learning approach

VIII. Some learning algorithms for imbalanced data sets

IX. Imbalanced Big Data

X. Class imbalance: Data sets, implementations, ...

XI. Class imbalance: Trends and final comments

Contents

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance**
- VII. Multiple class imbalanced data-sets: A pairwise learning approach**
- VIII. Imbalanced Big Data**
- IX. Class imbalance: Data sets, implementations, ...**
- X. Class imbalance: Trends and final comments**

SESSION 1

SESSION 2

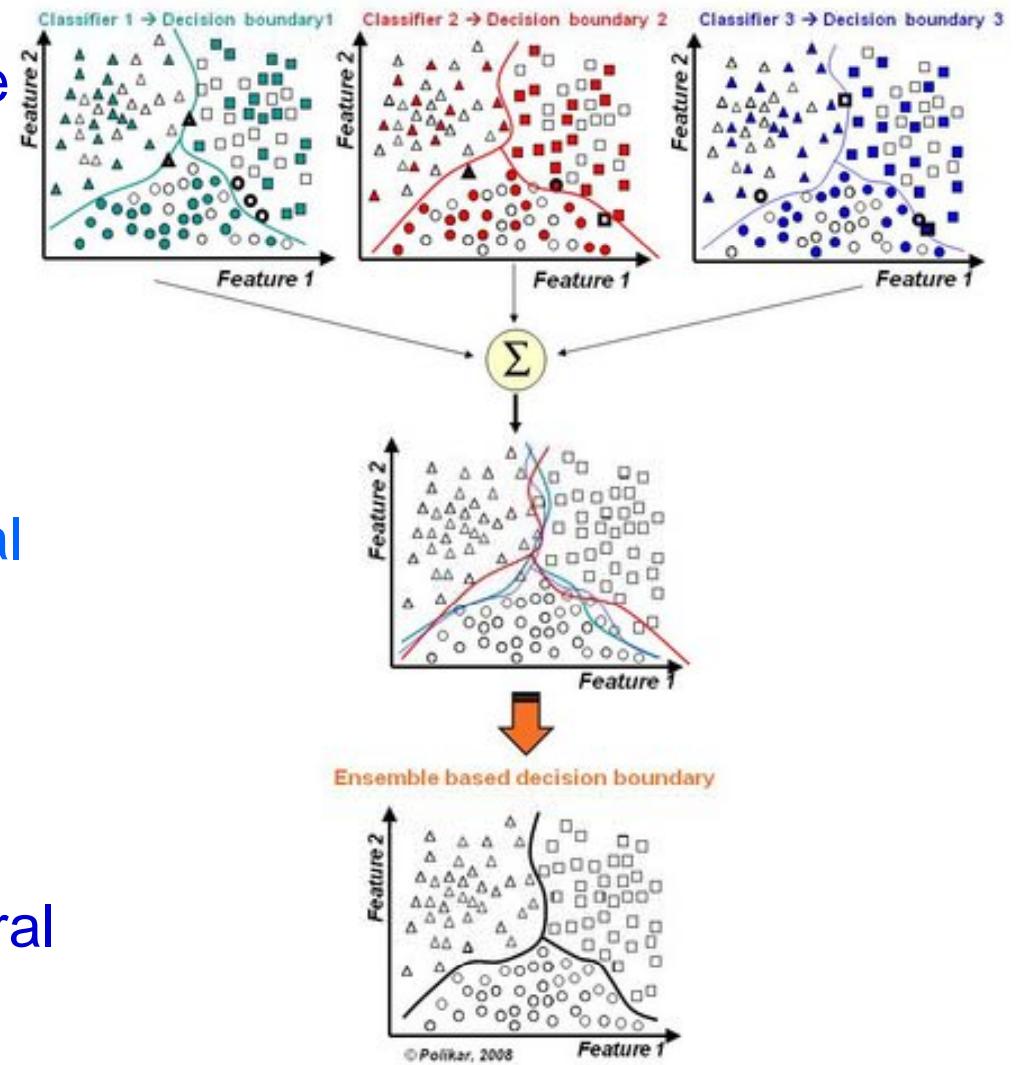
SESSION 3

Ensembles 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.

The basic idea is to construct several classifiers from the original data and then aggregate their predictions when unknown instances are presented.

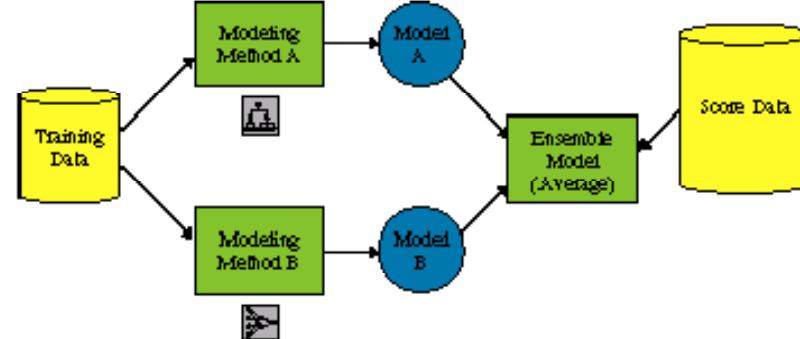
This idea follows human natural behavior which tend to seek several opinions before making any important decision.



Ensembles to address class imbalance

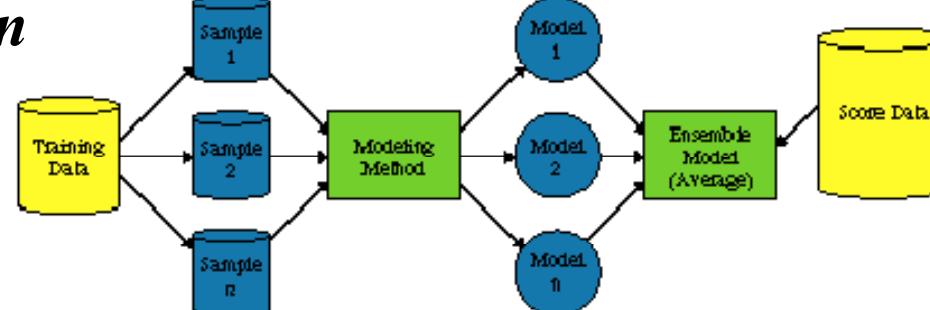
Ensemble methods classification

- *Manipulation with model*
(Model= $M(a)$)



Bagging =
Manipulation with data set

- *Manipulation with data set*



Boosting =
Manipulation with model

Ensembles to address class imbalance

Bagging and Boosting

General idea

Training data

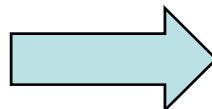
Classification method (CM)



Classifier C

Altered Training data

CM



Classifier C1

Altered Training data

CM



Classifier C2

.....

Aggregation



Classifier C*

Ensembles to address class imbalance

Bagging (Algorithm 1). It consists in training different classifiers with bootstrapped replicas of the original training data-set. That is, **a new data-set is formed to train each classifier by randomly drawing (with replacement) instances from the original data-set** (usually, maintaining the original data-set size). Hence, diversity is obtained by resampling different data subsets. Finally, when an unknown instance is presented to each individual classifier, a majority or weighted vote is used to determine the class.

Algorithm 1 Bagging

Input: S : Training set; T : Number of iterations;

n : Bootstrap size; I : Weak learner

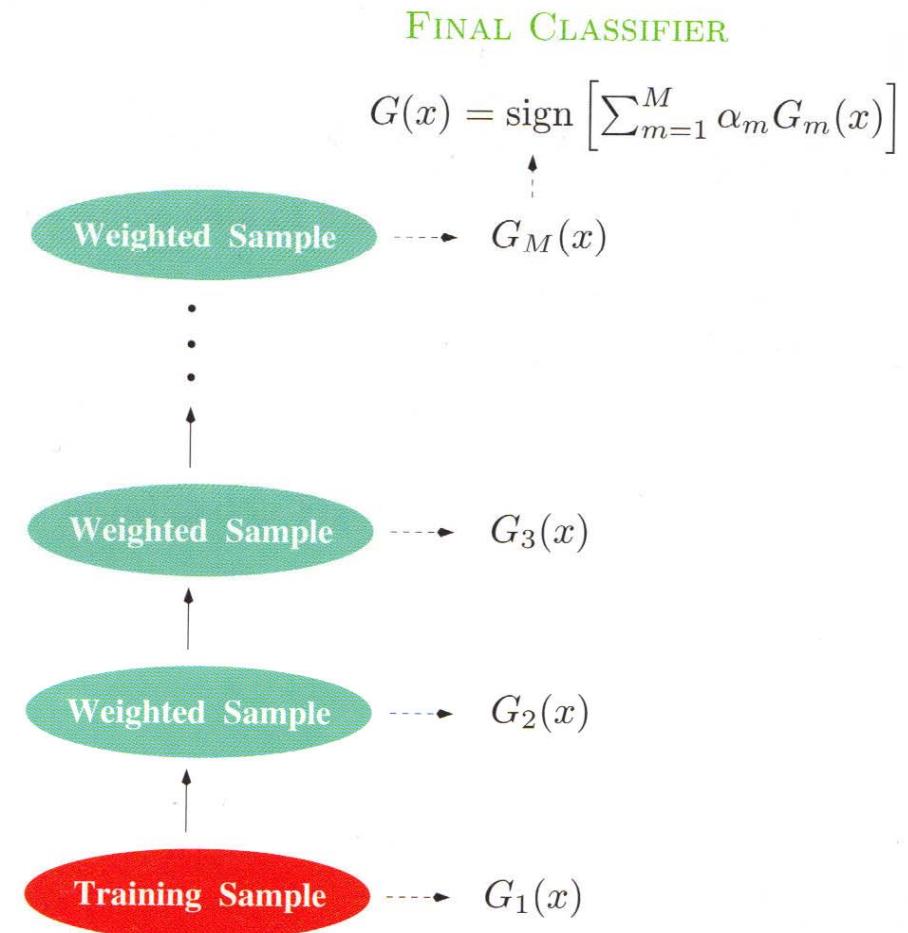
Output: Bagged classifier: $H(x) = \text{sign} \left(\sum_{t=1}^T h_t(x) \right)$ where $h_t \in [-1, 1]$ are the induced classifiers

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

Ensembles to address class imbalance

Boosting

The idea



Ensembles to address class imbalance

Boosting (AdaBoost, Algorithm 3). AdaBoost is the most representative algorithm of Boosting family. AdaBoost uses the whole data-set to train each classifier serially, but after each round, it gives more focus to difficult instances, with the goal of correctly classifying in the following iteration those examples that were incorrectly classified during the current one. **After each iteration, the weights of misclassified instances are increased; on the contrary, the weights of correctly classified instances are decreased.**

Furthermore, each individual classifier is assigned a weight depending on its overall accuracy (the weight is then used in test phase); more confidence is given to more accurate classifiers.

Algorithm 3 AdaBoost

Input: Training set $S = \{\mathbf{x}_i, y_i\}$, $i = 1, \dots, N$; and $y_i \in \{-1, +1\}$; T : Number of iterations; I : Weak learner

Output: Boosted classifier: $H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$ where h_t, α_t are the induced classifiers (with $h_t(x) \in \{-1, 1\}$) and their assigned weights, respectively

- 1: $D_1(i) \leftarrow 1/N$ for $i = 1, \dots, N$
 - 2: **for** $t = 1$ to T **do**
 - 3: $h_t \leftarrow I(S, D_t)$
 - 4: $\varepsilon_t \leftarrow \sum_{i, y_i \neq h_t(\mathbf{x}_i)} D_t(i)$
 - 5: **if** $\varepsilon_t > 0.5$ **then**
 - 6: $T \leftarrow t - 1$
 - 7: **return**
 - 8: **end if**
 - 9: $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\varepsilon_t}{\varepsilon_t} \right)$
 - 10: $D_{t+1}(i) = D_t(i) \cdot e^{(-\alpha_t h_t(\mathbf{x}_i) y_i)}$ for $i = 1, \dots, N$
 - 11: Normalize D_{t+1} to be a proper distribution
 - 12: **end for**
-

Ensembles to address class imbalance

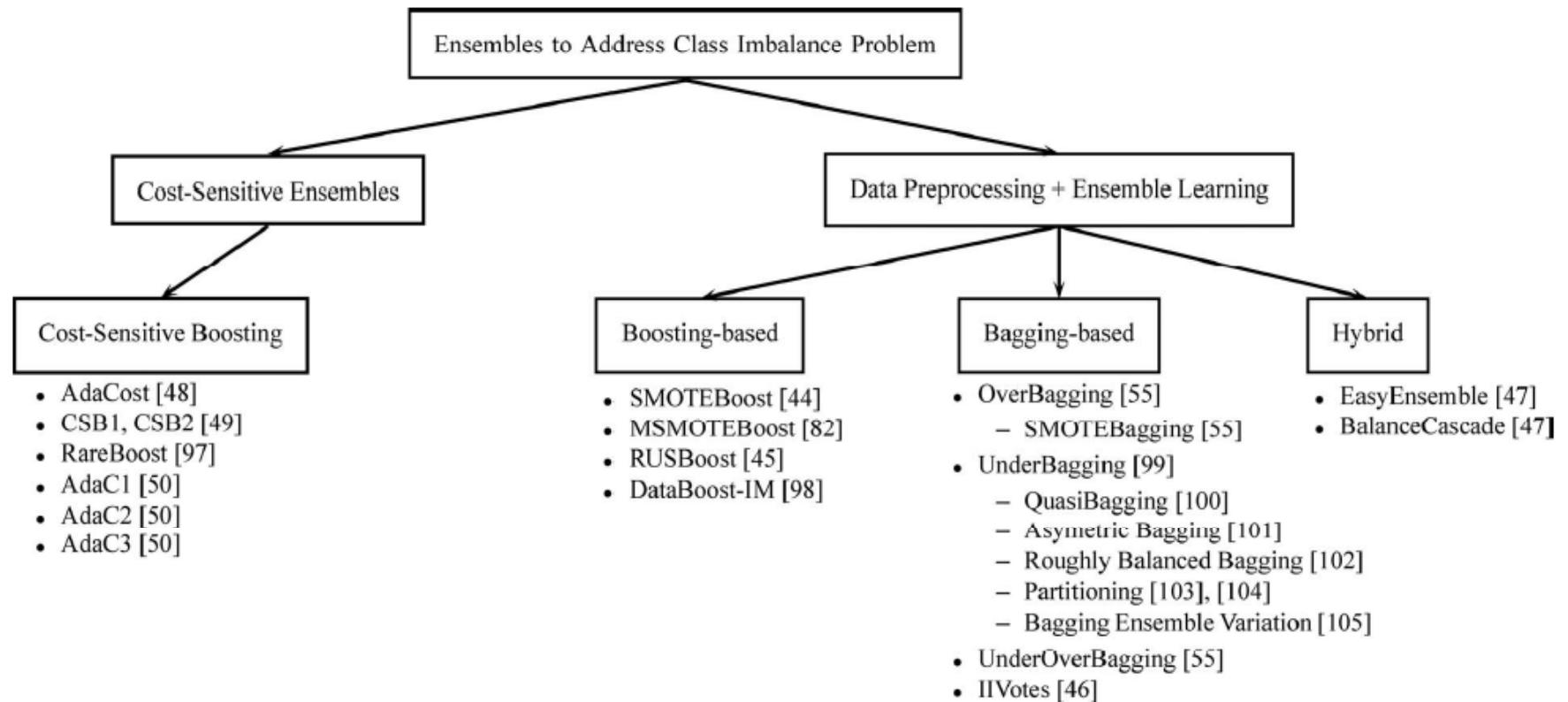


Fig. 3. Proposed taxonomy for ensembles to address the class imbalance problem.

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

Ensembles to address class imbalance

TABLE XV
REPRESENTATIVE METHODS SELECTED FOR EACH FAMILY

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

TABLE XVIII
SHAFFER TESTS FOR INTERFAMILY COMPARISON

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

SBAG SMOTEBagging Bagging where each bag's SMOTE quantity varies ($T=40$)

RusBoost ($T=10$) removes instances from the majority class by random undersampling the dataset in each iteration.

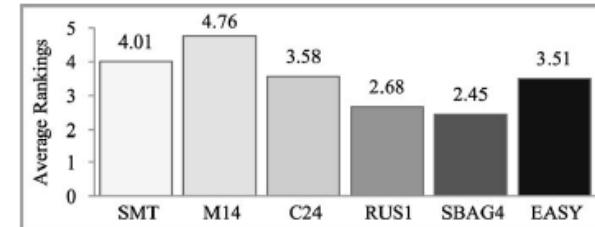


fig. 9. Average rankings of the representatives of each family.

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.

Ensembles 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 Boosting/Bagging ensemble learning algorithm.

Ensembles to address class imbalance

Our proposal recent proposal: **EUSBoost**

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.

M. Galar, A. Fernandez, E. Barrenechea, F. Herrera, **EUSBoost: Enhancing Ensembles for Highly Imbalanced Data-sets by Evolutionary Undersampling.** *Pattern Recognition*, [doi: j.patcog.2013.05.006](https://doi.org/10.1016/j.patcog.2013.05.006), *in press* (2013)

Ensembles to address class imbalance

EUSBoost: ENHANCING ENSEMBLES FOR HIGHLY IMBALANCED DATA-SETS

MOTIVATION

- **RUSBoost** is highly **competitive**
 - Good trade-off: performance/complexity
- **Ensembles and random sampling** obtained great results
 - Competitive despite its **randomness**
 - **Accuracy-diversity** relation
- Such a randomness might be improved
 - It could discard potentially useful instances
 - More probable as the IR increases
- “*Accurate base classifiers lead to better ensembles than much diverse ones*” [\[Kuncheva12\]](#)

[KUNCHEVA12](#) L. Kuncheva. A bound on kappa-error diagrams for analysis of classifier ensembles, IEEE Transactions on Knowledge Data Engineering, doi: 10.1109/TKDE.2011.234, 2012 in press.

M. Galar, A. Fernandez, E. Barrenechea, F. Herrera, **EUSBoost: Enhancing Ensembles for Highly Imbalanced Data-sets by Evolutionary Undersampling**. *Pattern Recognition*, doi: j.patcog.2013.05.006, *in press* (2013)

Ensembles to address class imbalance

EUSBoost

- EUSBoost
- Supervised undersampling:
 - Evolutionary Undersampling (EUS)
 - EUS outperforms random undersampling on highly imbalanced data-sets
- Diversity promotion mechanism

M. Galar, A. Fernandez, E. Barrenechea, F. Herrera, **EUSBoost: Enhancing Ensembles for Highly Imbalanced Data-sets by Evolutionary Undersampling**. *Pattern Recognition*, [doi: j.patcog.2013.05.006](https://doi.org/10.1016/j.patcog.2013.05.006), *in press* (2013)

Ensembles to address class imbalance

EUSBoost: ENHANCING ENSEMBLES FOR HIGHLY IMBALANCED DATA-SETS

PROMOTING DIVERSITY

- **Diversity** is crucial
- **No direct relation** between diversity and accuracy
- In imbalanced domains, diversity is a key factor^[Wang12]
 - Relation between diversity and single-class measures
 - It positively affects AUC and GM
- EUSBoost: seeking for accuracy causes a **loss of diversity**
 - **Solution:** promote diversity in the fitness function
 - Favor chromosomes better combining accuracy and diversity
 - Forcing the usage of **different data-subsets**

^{WANG12} S. Wang and X. Yao, Relationships between diversity of classification ensembles and single-class performance measures, IEEE Transactions on Knowledge Data Engineering, doi: 10.1109/TKDE.2011.207, 2012 in press.

Ensembles to address class imbalance

EUSBoost: ENHANCING ENSEMBLES FOR HIGHLY IMBALANCED DATA-SETS

PROMOTING DIVERSITY IN THE EVOLUTIONARY MODEL

- Performance (Perf): AUC or GM, 1NN hold-one-out

$$GM = \sqrt{TP_{rate} \cdot TN_{rate}}$$

- **Fitness function:**

$$\text{Fitness}_{\text{EUS}} = \begin{cases} \text{Perf} - \left| 1 - \frac{n^+}{N^-} \cdot P \right| & \text{if } N^- > 0 \\ \text{Perf} - P & \text{if } N^- = 0 \end{cases}$$

- N^- no. of majority class instances selected
- $P = 0.2$ importance of the balance

- We use $\text{Perf} = \text{GM}$ [García09]

- $\text{Fitness}_{\text{EUS}}$ original fitness function

- Weighting factor β

$$\text{Fitness}_{\text{EUS}_Q} = \text{Fitness}_{\text{EUS}} \cdot \frac{1.0}{\beta} \cdot \frac{10.0}{\text{IR}} - Q \cdot \beta$$

$$\beta = \frac{N - t - 1}{N}$$

- $t = 1$ original EUS

Ensembles to address class imbalance

EUSBoost: ENHANCING ENSEMBLES FOR HIGHLY IMBALANCED DATA-SETS

PROMOTING DIVERSITY IN THE EVOLUTIONARY MODEL

- Several measures^[Kuncheva03]
- **Q -statistic** of (V_i, V_j)

$$Q_{i,j} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \in [-1, 1]$$

- N^{ab} no. of elements with value a in V_i and b in V_j
- Lower values indicates greater diversity
- Q is a **pairwise measure**, but we compare **several solutions**
 - The **maximum** of all pairwise $Q_{i,j}$

$$Q = \max_{i=1, \dots, t} Q_{i,j} \quad V_i, \quad i = 1, \dots, t$$

- V_j is the candidate solution, t is the current iteration

KUNCHEVA03 L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine Learning 51:181-207, 2003.

M. Galar, A. Fernandez, E. Barrenechea, F. Herrera, **EUSBoost: Enhancing Ensembles for Highly Imbalanced Data-sets by Evolutionary Undersampling**. *Pattern Recognition*, doi: j.patcog.2013.05.006, **in press (2013)**

Ensembles to address class imbalance

- 10 classifiers for Boosting-based
- 40 classifiers for Bagging-based
- The 33 most imbalanced data-sets from KEEL repository

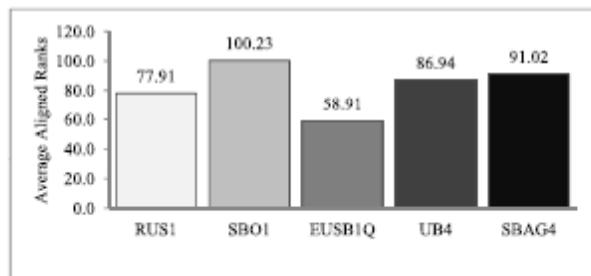


FIGURE : Average aligned ranks.

TABLE : Holm test, EUSB1_Q vs. state-of-the-art.

Control method: EUSB _Q (58.91), AF p-value 0.00001					
i	Algorithm (Rank)	Z	p-value	Holm	Hypothesis ($\alpha = 0.05$)
4	SBO1 (100.23)	3.51300	0.00044	0.0125	Rejected for EUSB _Q
3	SBAG4 (91.02)	2.72976	0.00634	0.01667	Rejected for EUSB _Q
2	UB4 (86.94)	2.38322	0.01716	0.025	Rejected for EUSB _Q
1	RUS1 (77.91)	1.61544	0.10622	0.05	Not Rejected

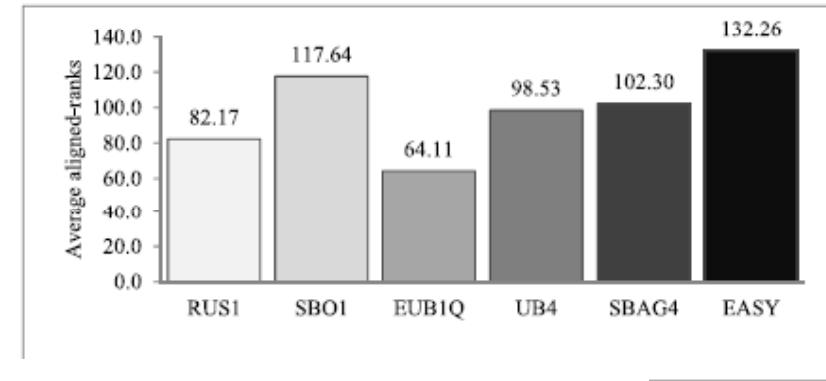
TABLE : Wilcoxon tests to compare EUSB1_Q with RUS1. R^+ corresponds to EUSB1_Q and R^- to RUS1.

Comparison	R^+	R^-	Hypothesis ($\alpha = 0.1$)	p-value
EUSB1 _Q vs. RUS1	378.5	182.5	Rejected for EUSB1 _Q	0.06432

Ensembles to address class imbalance

Summary: EUSBoost

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



- A novel approach to enhance ensembles in highly imbalanced scenarios
 - **EUS** instead of random undersampling
 - **Diversity promotion** mechanism
- **EUSBoost outstands** vs. the state-of-the-art methods
- Adaptation of **kappa-error diagrams**
 - **Analyze** the advantages and disadvantages of **EUSBoost**
 - The **importance of the individual accuracy** has been shown

Contents

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance
- VII. **Multiple class imbalanced data-sets: A pairwise learning approach**
- VIII. **Imbalanced Big Data**
- IX. **Class imbalance: Data sets, implementations, ...**
- X. **Class imbalance: Trends and final comments**

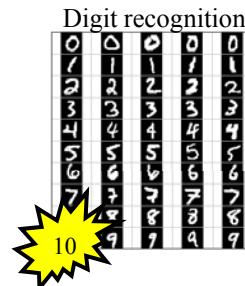
SESSION 1

SESSION 2

SESSION 3

Multiple class imbalanced data sets: A pairwise learning approach

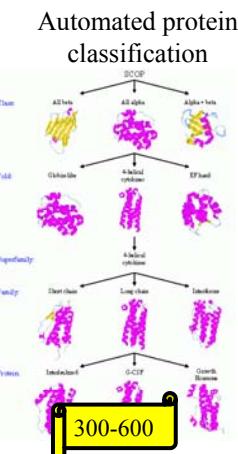
- Multiple classes imply an additional difficulty with high imbalance ratio between classes.
- Imbalanced problems: the proposed solutions for the binary case could not be directly applicable or could obtain a lower performance than expected.
 - Increment of the search space in solutions at the data level
 - Difficulty in adapting the solutions at the algorithm level



Phoneme recognition

I:	ɛ	ʊ	ʌ	eɪ	ɛɪ	ɪə	ɛə
READ	AMERICA	WEIRD	BORN	TOUR	BOY	GO	HOW
æ	A:	a:	d	ee	ai	o	u:
CAT	BUT	PART	HOT	WEAR	MY	HOW	UH
p	b	t	d	f	θ	k	g
RED	TIM	DO	CHIRP	SUPER	KILO	GO	WHO
θ	ð	s	z	ʃ	ʒ	χ	χ
THINK	THE	SIX	ZOO	SHIRT	JOINT	CASUAL	THEIR
ŋ	h	l	r	w	j	ɹ	v
FIND	BELLO	LIVE	READ	WINDOW	ZEAL	WEE	VEE

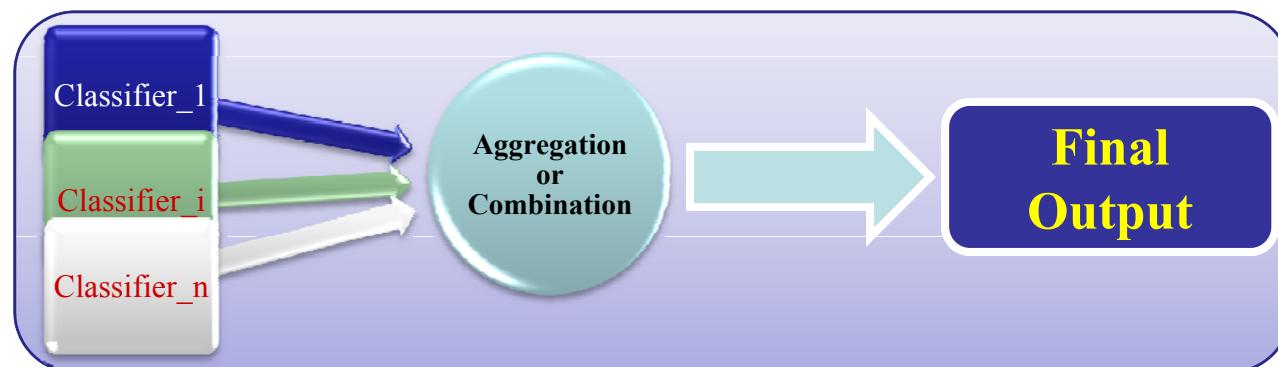
50



Multiple class imbalanced data sets: A pairwise learning approach

Pairwise Learning: Binarization

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

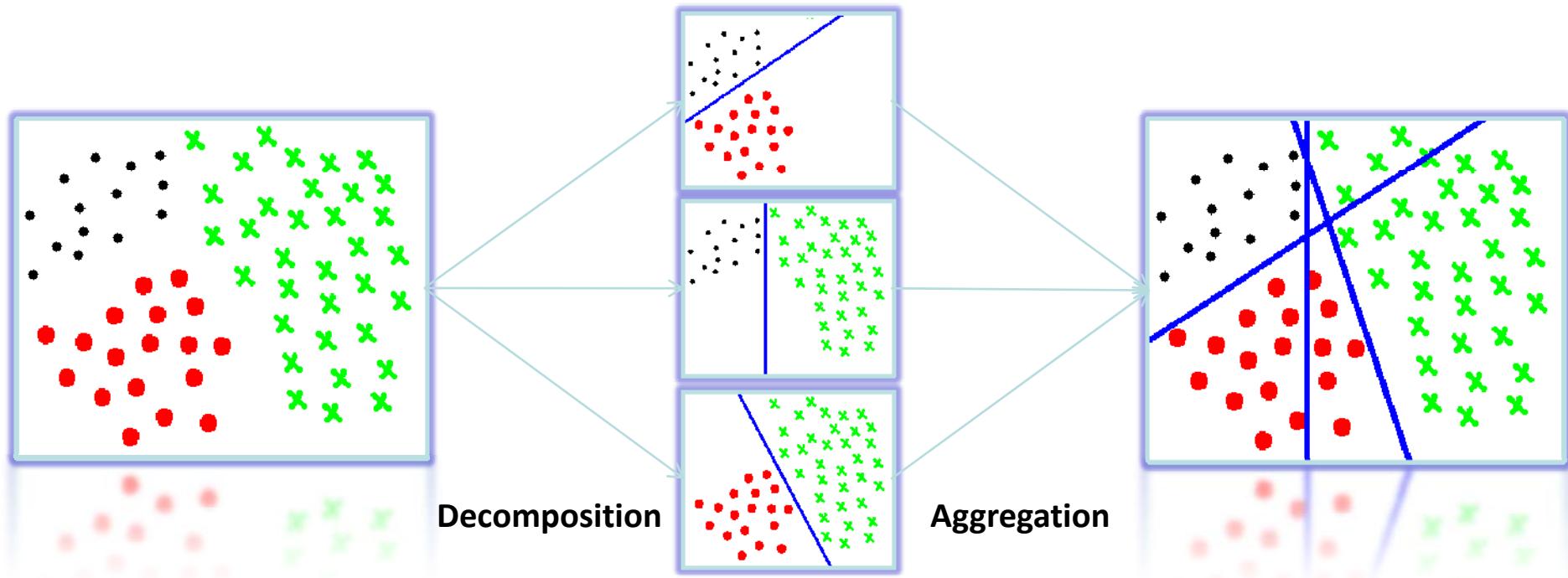


Multiple class imbalanced data sets: A pairwise learning approach

Pairwise Learning: Decomposition Strategies

“One-vs-One strategy” (OVO)

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



Multiple class imbalanced data sets: A pairwise learning approach

Pairwise Learning: Decomposition Strategies

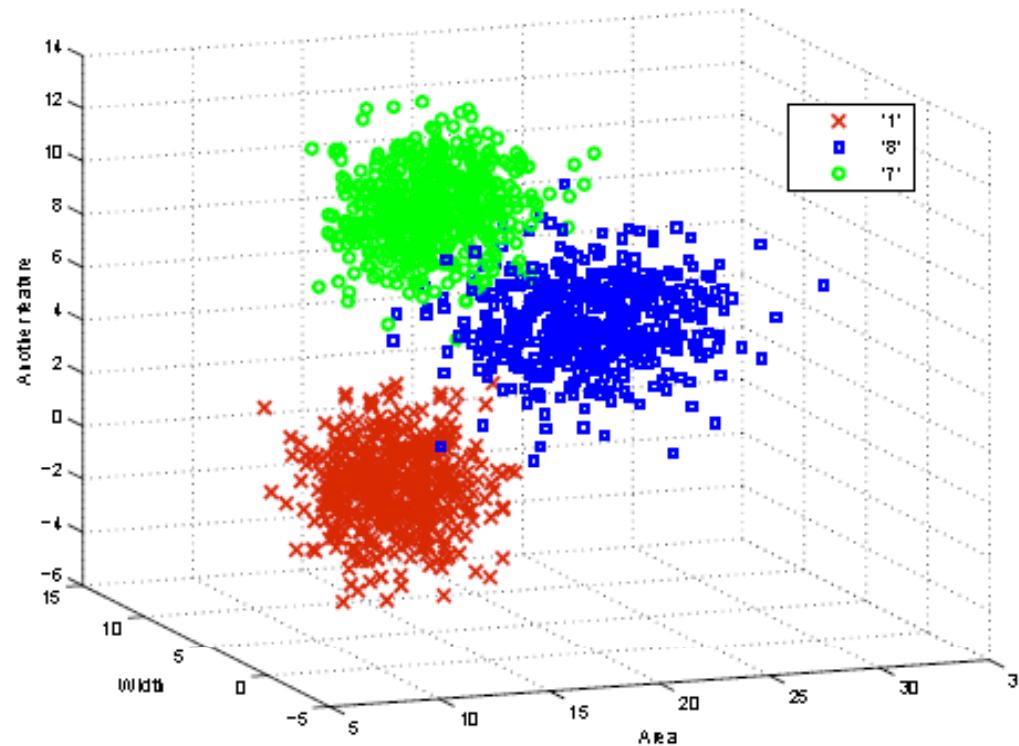


FIGURE : A multi-class problem, a new feature is needed

Multiple class imbalanced data sets: A pairwise learning approach

Pairwise Learning: Decomposition Strategies

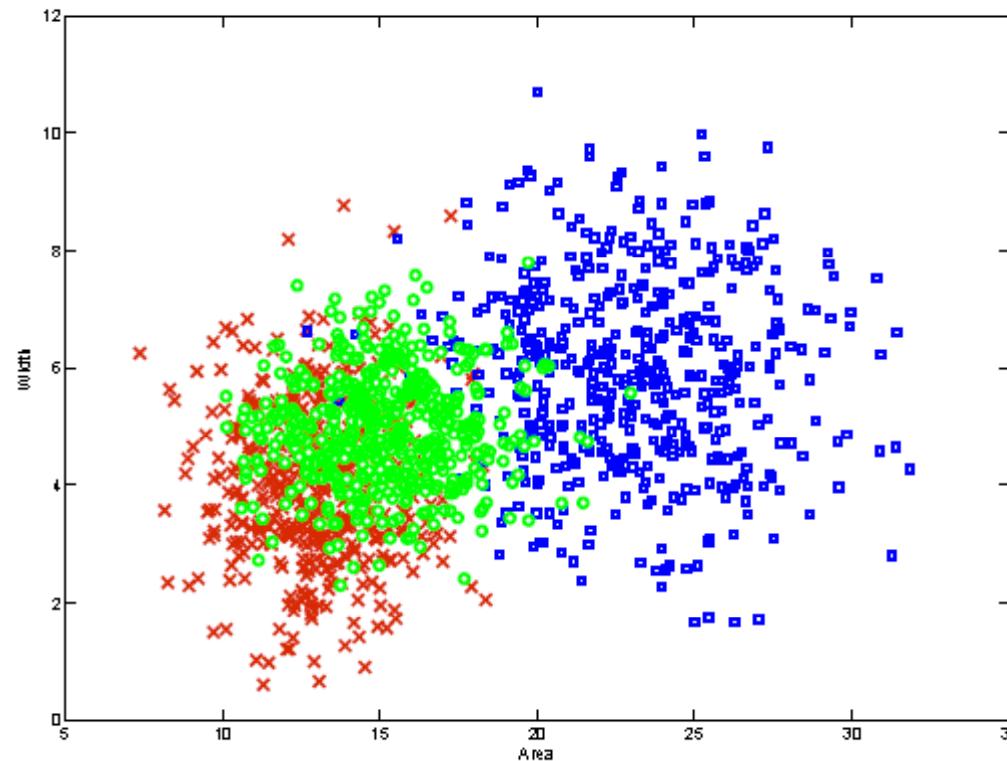
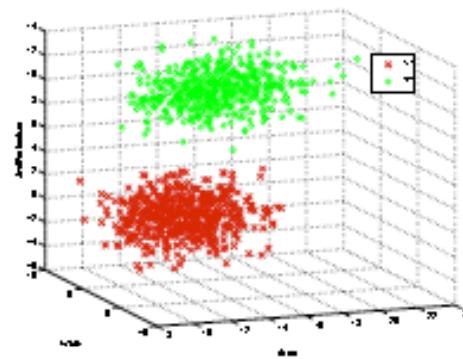


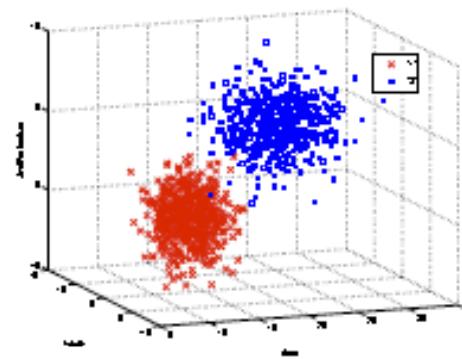
FIGURE : A multi-class problem

Multiple class imbalanced data sets: A pairwise learning approach

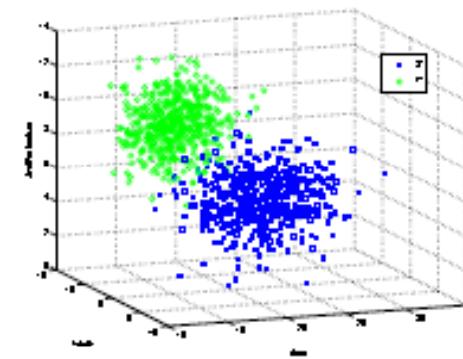
Pairwise Learning: Decomposition Strategies



(a) '1' vs. '7'



(b) '1' vs. '8'

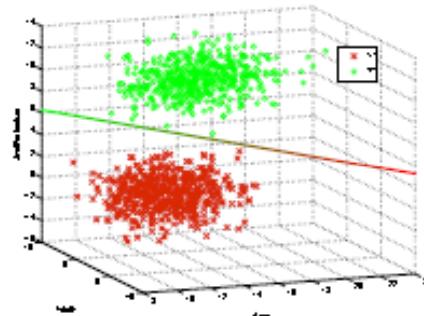


(c) '8' vs. '7'

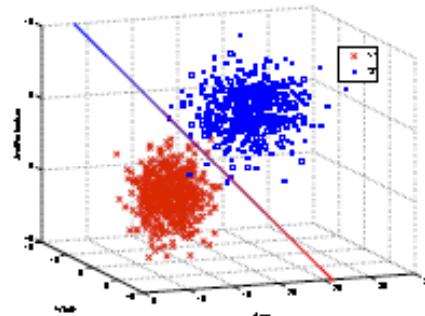
FIGURE: One-vs-One scheme

Multiple class imbalanced data sets: A pairwise learning approach

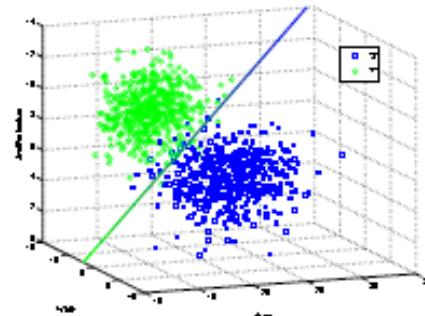
Pairwise Learning: Decomposition Strategies



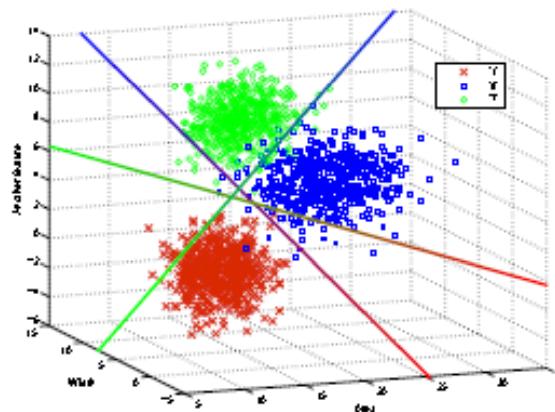
(a) '1' vs. '7'



(b) '1' vs. '8'



(c) '8' vs. '7'

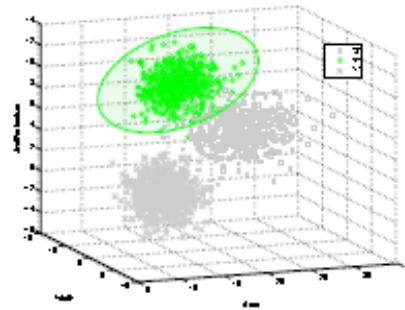


(d) Aggregation

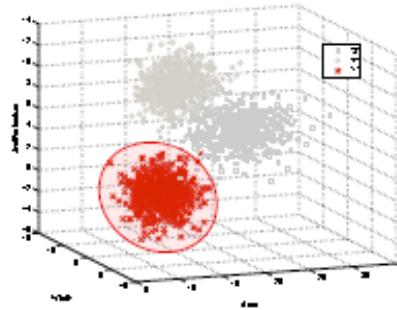
FIGURE : One-vs-One scheme

Multiple class imbalanced data sets: A pairwise learning approach

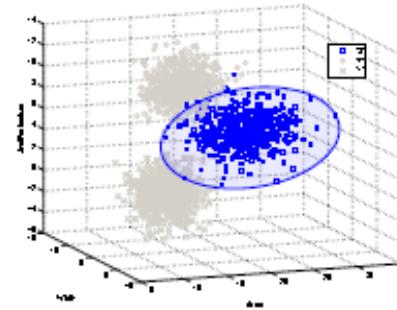
Other Decomposition Strategies: One vs All



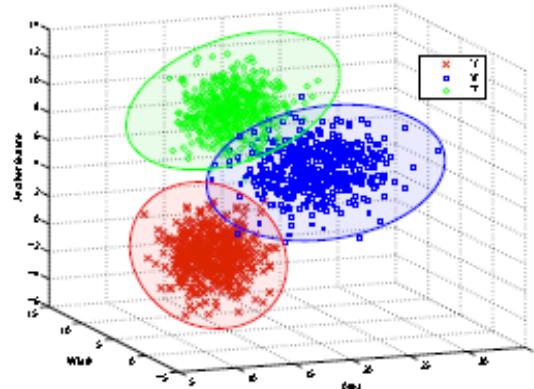
(a) '7' vs. '1' and '8'



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



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



(d) Aggregation

FIGURE : One-vs-All scheme

Multiple class imbalanced data sets: A pairwise learning approach

Pairwise Learning: Decomposition Strategies

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

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

Multiple class imbalanced data sets: A pairwise learning approach

Pairwise Learning: Combination of the outputs

- Aggregation phase
 - *The way in which the outputs of the base classifiers are combined to obtain the final output.*
- Starting from the score-matrix

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

- r_{ij} = confidence of classifier in favor of class i
- r_{ji} = confidence of classifier in favor of class j
 - Usually: $r_{ji} = 1 - r_{ij}$ (required for probability estimates)

Multiple class imbalanced data sets: A pairwise learning approach

Pairwise Learning: Combination of the outputs

- Non-Dominance Criterion (ND) [Fernandez10]
 - Decision making and preference modeling [Orlovsky78]
 - Score-Matrix = preference relation
 - $r_{ji} = 1 - r_{ij}$, if not → normalize $\bar{r}_{ij} = \frac{r_{ij}}{r_{ij} + r_{ji}}$
 - Compute the maximal non-dominated elements
 - Construct the strict preference relation $r'_{ij} = \begin{cases} \bar{r}_{ij} - \bar{r}_{ji}, & \text{when } \bar{r}_{ij} > \bar{r}_{ji} \\ 0, & \text{otherwise.} \end{cases}$
 - Compute the non-dominance degree $ND_i = 1 - \sup_{j \in C} [r'_{ji}]$
 - » *the degree to which the class i is dominated by no one of the remaining classes*
 - Output

$$\text{Class} = \arg \max_{i=1, \dots, m} \{ND_i\}$$

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

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

Multiple class imbalanced data sets: A pairwise learning approach

Our Proposal

A two stage methodology is proposed:

1. Simplifying the initial problem in several binary sets.

The “OVO” technique is employed:

- More **precise** for rule learning algorithms [Fürnkranz02].
2. Less biased to obtain **imbalanced** training subsets. The **SMOTE** algorithm [Chawla02] is applied to those subsets with a significative Imbalance Ratio (IR). IR threshold = 1.5 (60-40% distribution).

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

A. Fernandez, M.J. del Jesus, F. Herrera, **Multi-class Imbalanced Data-Sets with Linguistic Fuzzy Rule Based Classification Systems Based on Pairwise Learning**. *13th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU2010)* Dortmund (Germany), LNAI 6178 pp 89-98, 28 June - 02 July 2010.

Multiple class imbalanced data sets: A pairwise learning approach

Experimental framework

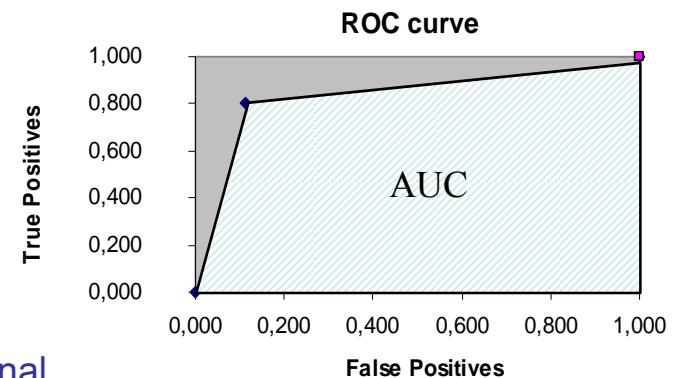
Performance measures in this domain:

The use of common metrics like accuracy rate may lead to erroneous conclusions.

Confusion Matrix for a binary problem:

		Prediction	
		Pos. Class	Neg. Class
Real	Positive Class	TP	FN
	Negative Class	FP	TN

True Diagonal



$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \quad PAUC = \frac{1}{C(C-1)} \sum_{j=1}^C \sum_{k \neq j}^C AUC(j, k)$$

Multiple class imbalanced data sets: A pairwise learning approach

Experimental framework

- **16 real-world Data-sets**
- **5 fold-cross validation**

id	Data-set	#Ex.	#Atts.	#Num.	#Nom.	#Cl.	IR
aut	autos	159	25	15	10	6	16.00
bal	balance scale	625	4	4	0	3	5.88
cle	cleveland	297	13	6	7	5	13.42
con	contraceptive method choice	1,473	9	6	3	3	1.89
der	dermatology	366	33	1	32	6	5.55
eco	ecoli	336	7	7	0	8	71.50
gla	glass identification	214	9	9	0	6	8.44
hay	hayes-roth	132	4	4	0	3	1.70
lym	lymphography	148	18	3	15	4	40.50
new	new-thyroid	215	5	5	0	3	4.84
pag	page-blocks	548	10	10	0	5	164.00
pen	pen-based recognition	1,099	16	16	0	10	1.95
shu	shuttle	2,175	9	9	0	5	853.00
thy	thyroid	720	21	6	15	3	36.94
win	wine	178	13	13	0	3	1.5
yea	yeast	1,484	8	8	0	10	23.15

Multiple class imbalanced data sets: A pairwise learning approach

Experimental framework

- FH-GBML Algorithm [Ishibuchi05]
 - Default configuration KEEL software tool (<http://www.keel.es>)
 - Number of fuzzy rules: $5 \cdot d$ rules (max. 50 rules).
 - Number of rule sets: 200 rule sets.
 - Crossover probability: 0.9.
 - Mutation probability: $1/d$.
 - Number of replaced rules: All rules except the best-one (Pittsburgh-part, elitist approach), number of rules / 5 (GCCL-part).
 - Total number of generations: 1,000 generations.
 - Don't care probability: 0.5.
 - Probability of the application of the GCCL iteration: 0.5.

[Ishibuchi05] Ishibuchi, H., Yamamoto, T., Nakashima, T.: Hybridization of fuzzy GBML approaches for pattern classification problems. IEEE Transactions on System, Man and Cybernetics B 35(2) (2005) 359–365

Multiple class imbalanced data sets: A pairwise learning approach

Experimental study

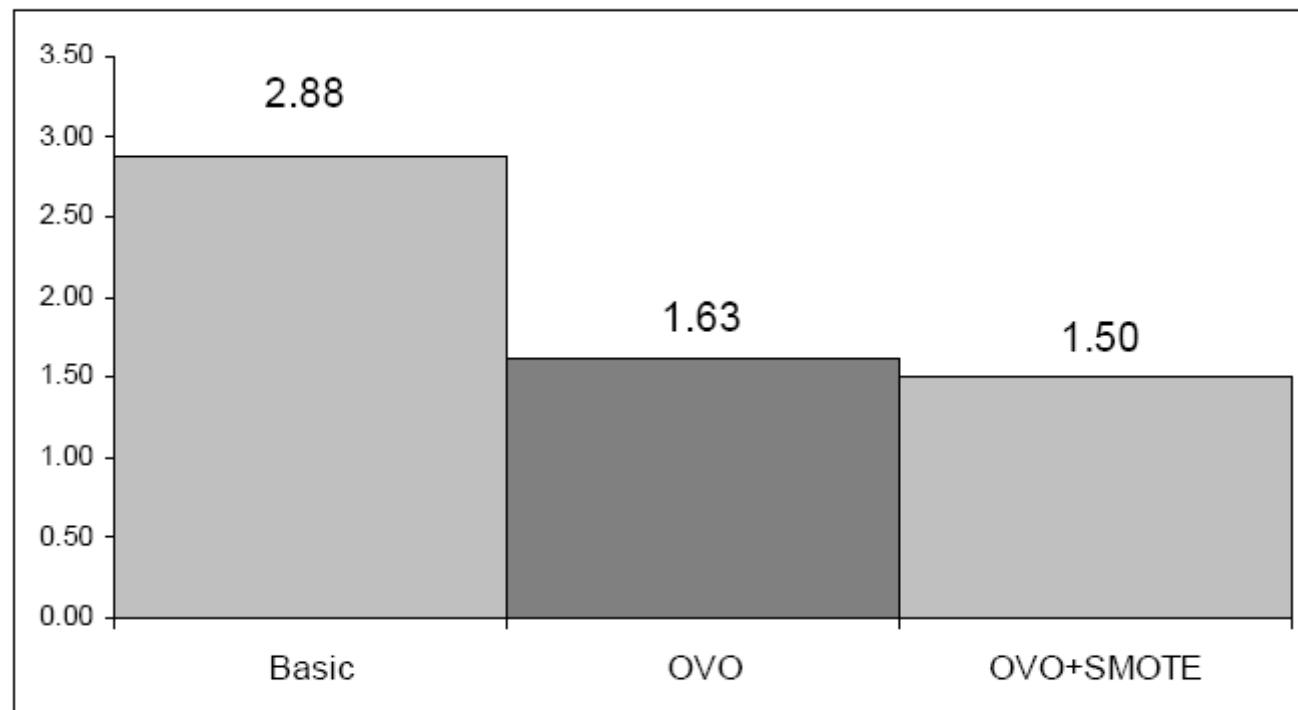
- Results for the FH-GBML algorithm with the different classification approaches

Data-set	Base		OVO		OVO+SMOTE	
	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
aut	.7395	.6591	.8757	.6910	.8032	.6829
bal	.7178	.7008	.7307	.7109	.7992	.7296
cle	.6395	.5577	.7366	.5664	.7949	.5584
con	.5852	.5623	.6468	.6201	.6683	.6294
der	.7169	.6862	.9746	.9084	.9614	.8716
eco	.7564	.7811	.9269	.8201	.9578	.8321
gla	.7426	.6920	.8691	.7444	.9375	.8207
lym	.8590	.7626	.9349	.8397	.9284	.8689
hay	.7979	.6954	.9597	.6656	.9663	.6456
new	.9490	.8861	.9967	.9564	.9850	.9457
pag	.7317	.6929	.9472	.7862	.9696	.8552
pen	.8460	.8340	.9798	.9508	.9740	.9387
shu	.7253	.7709	.9319	.8635	.9950	.9516
thy	.5198	.4992	.5304	.4993	.9193	.8763
win	.9847	.9501	1.000	.9710	.9974	.9519
yea	.6456	.6272	.8042	.7438	.8365	.7442
Mean	.7473	.7099	.8653	.7711	.9075	.8064

Multiple class imbalanced data sets: A pairwise learning approach

Experimental study

- Average ranking for the FH-GBML method with the different classification schemes



Multiple class imbalanced data sets: A pairwise learning approach

Experimental study

- Statistical analysis
 - Wilcoxon signed-ranks test. R^+ corresponds to the sum of the ranks for the OVO+SMOTE method and R^- to the Basic and OVO classification schemes

Comparison	R^+	R^-	p-value	Hypothesis ($\alpha = 0.05$)
OVO+SMOTE vs. Basic	131.0	5.0	0.001	Rejected for OVO+SMOTE
OVO+SMOTE vs. OVO	88.0	48.0	0.301	Not Rejected

- The methodology is actually better suited for imbalanced data-sets with multiple classes than the basic learning algorithm.
- The application of the oversampling step enables the achievement of better results than applying the binarization scheme directly over the original training data.

Multiple class imbalanced data sets: A pairwise learning approach

Final comments on the pairwise learning approach:

- Goodness of using binarization for imbalanced data-sets.
- Improvement by means of the application of preprocessing for each binary subset.
- Different Machine Learning algorithms to analyse the robustness of the methodology are used with similar results (C4.5, SVM, PDPC).

A. Fernández, V. López, M. Galar, M.J. del Jesus, F. Herrera. Analysing the Classification of Imbalanced Data-sets with Multiple Classes: Binarization Techniques and Ad-Hoc Approaches. *Knowledge-Based Systems* 42 (2013) 97-110.

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](https://doi.org/10.1016/j.patcog.2011.01.017).

Multiple class imbalanced data sets: A pairwise learning approach

Random oversampling + AdaBoost.NC

This approach is based on AdaBoost algorithm in combination with negative correlation learning.

The main procedure is quite similar to any boosting approach, in which the weights of the examples are updated with an ad hoc formula depending on the classification or misclassification given by both the classifier learned in the current iteration, and the global ensemble.

Initial weights in this boosting approach are assigned in inverse proportion to the number of instances in the corresponding class.

Multiple class imbalanced data sets: A pairwise learning approach

Table 3

Number of instances per class.

Data	Examples	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
aut	159	46	13	48	29	20	3	-	-	-	-
bal	625	49	288	288	-	-	-	-	-	-	-
cle	467	164	36	35	55	13	164	-	-	-	-
con	1473	629	333	511	-	-	-	-	-	-	-
der	358	60	111	71	48	48	20	-	-	-	-
eco	336	143	77	2	2	35	20	5	42	-	-
fla	1066	331	239	211	147	95	43	-	-	-	-
gla	214	70	76	17	13	9	29	-	-	-	-
hay	160	65	64	31	-	-	-	-	-	-	-
led	500	45	37	51	57	52	52	47	57	53	49
tym	148	61	81	4	2	-	-	-	-	-	-
new	215	150	35	30	-	-	-	-	-	-	-
nur	12690	2	4320	4266	328	4044	-	-	-	-	-
pag	5472	4913	329	87	115	28	-	-	-	-	-
pos	87	62	24	1	-	-	-	-	-	-	-
sat	6435	1358	626	707	1508	703	1533	-	-	-	-
shu	57999	8903	45586	3267	49	171	13	10	-	-	-
spl	3190	767	768	1655	-	-	-	-	-	-	-
thy	7200	6666	368	166	-	-	-	-	-	-	-
win	178	59	71	48	-	-	-	-	-	-	-
wre	1599	681	638	199	53	18	10	-	-	-	-
wwh	4898	2198	1457	880	175	163	20	5	-	-	-
yea	1484	244	429	463	44	35	51	163	30	20	5
zoo	101	41	13	10	20	8	5	4	-	-	-

A. Fernandez, V. López, M. Galar, M.J. del Jesus, F. Herrera, **Analysing the Classification of Imbalanced Data-sets with Multiple Classes: Binarization Techniques and Ad-Hoc Approaches.** *Knowledge-Based Systems* 42 (2013) 97-110, [doi: 10.1016/j.knosys.2013.01.018](https://doi.org/10.1016/j.knosys.2013.01.018).

Multiple class imbalanced data sets: A pairwise learning approach

”Random oversampling + AdaBoost.NC”

Method	Adaptation	C4.5		SVM	
		Avg-Acc		Avg-Acc	
Std	Base	71.28		-	
	Global-CS	72.25		73.04	
	Static-SMT	70.18		70.53	
	AdaBoost.NC	74.03		71.70	
OVO	Std-OVO	69.97		69.14	
	ROS	72.35		72.41	
	SL-SMT	72.35		72.32	
	SMT-ENN	70.84		71.73	
	SMT	72.74		72.58	
	CS	71.95		72.70	

A. Fernandez, V. López, M. Galar, M.J. del Jesus, F. Herrera, **Analysing the Classification of Imbalanced Data-sets with Multiple Classes: Binarization Techniques and Ad-Hoc Approaches.** *Knowledge-Based Systems* 42 (2013) 97-110, [doi: 10.1016/j.knosys.2013.01.018](https://doi.org/10.1016/j.knosys.2013.01.018).

Multiple class imbalanced data sets: A pairwise learning approach

Multiple class imbalanced data sets: Some Comments

- The developed approaches have not considered the intrinsic data characteristics.
- In the near future, it would have interest to analyze them in order to develop algorithms according to the specific problems.
- It would have interest to consider the use of several pre-processing methods in ensembles.

A. Fernandez, V. López, M. Galar, M.J. del Jesus, F. Herrera, **Analysing the Classification of Imbalanced Data-sets with Multiple Classes: Binarization Techniques and Ad-Hoc Approaches.** *Knowledge-Based Systems* 42 (2013) 97-110, [doi: 10.1016/j.knosys.2013.01.018](https://doi.org/10.1016/j.knosys.2013.01.018).

Contents

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance
- VII. Multiple class imbalanced data-sets: A pairwise learning approach
- VIII. Imbalanced Big Data**
- IX. Class imbalance: Data sets, implementations, ...**
- X. Class imbalance: Trends and final comments

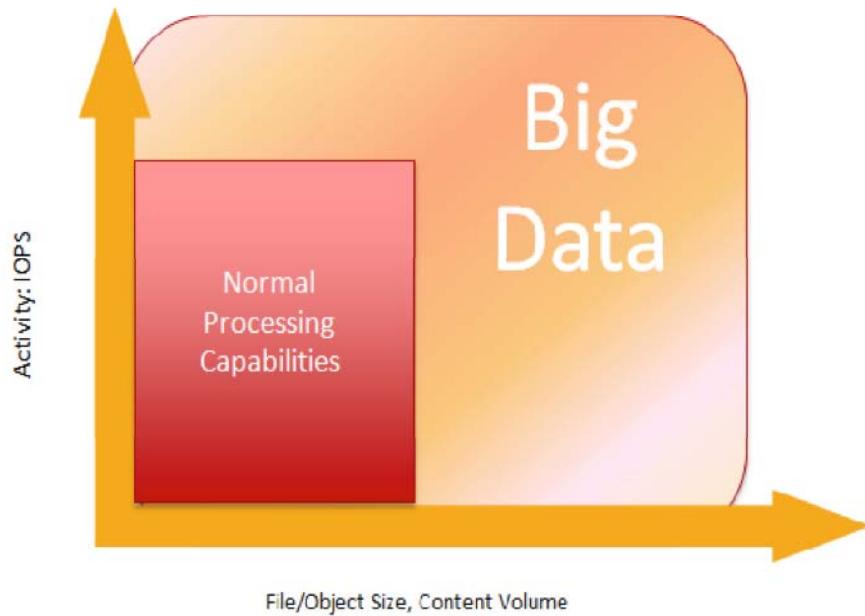
SESSION 1

SESSION 2

SESSION 3

Imbalanced big data

“Extremely
Imbalanced
Big Data
Problems”

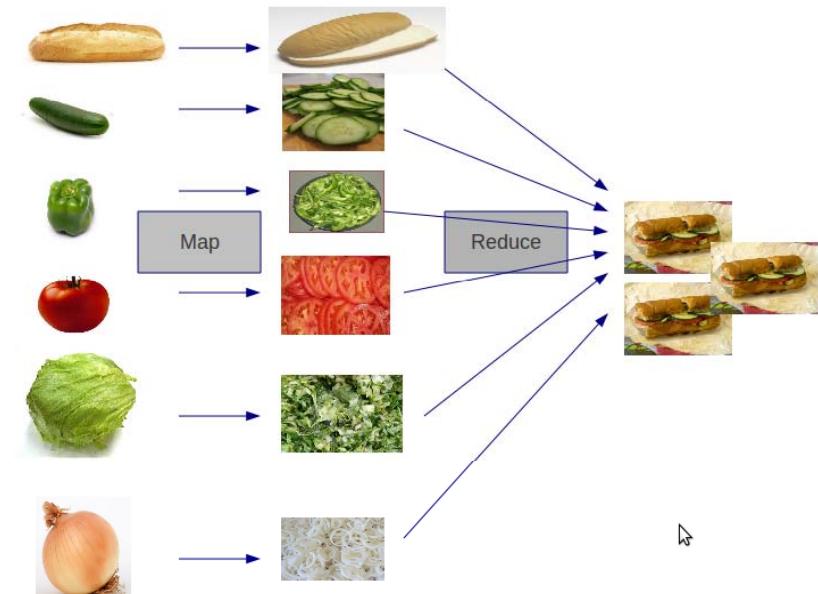
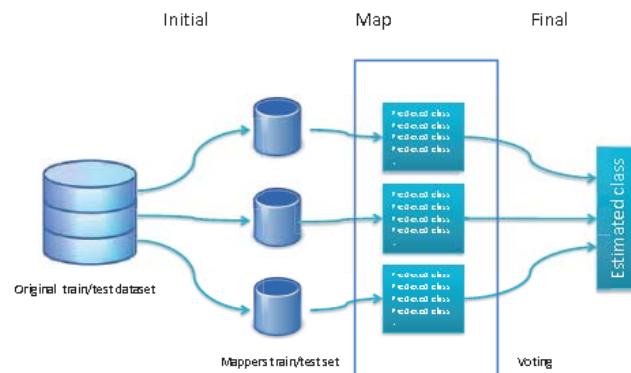


Imbalanced big data

“Extremely Imbalanced Big Data Problems”

How to tackle them?

MapReduce framework



Contents

VIII. Imbalanced Big Data

What is Big Data?

Big Data. MapReduce

Hadoop and Mahout

Extremely Imbalanced Big Data:

A case of study

What is Big Data?

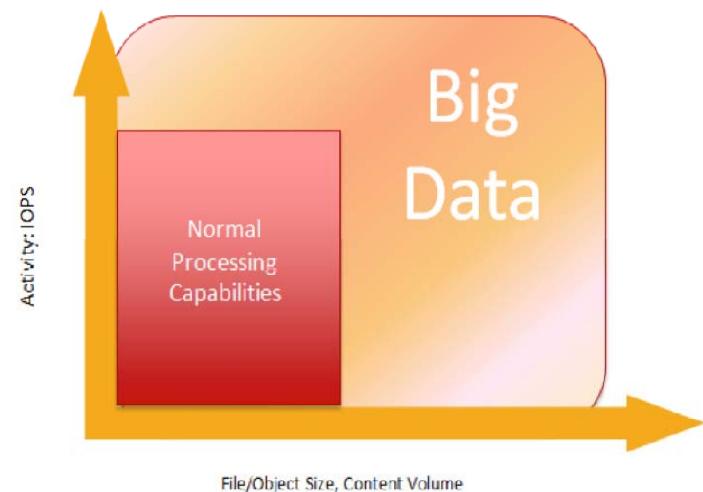
- No single standard definition



Big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications.

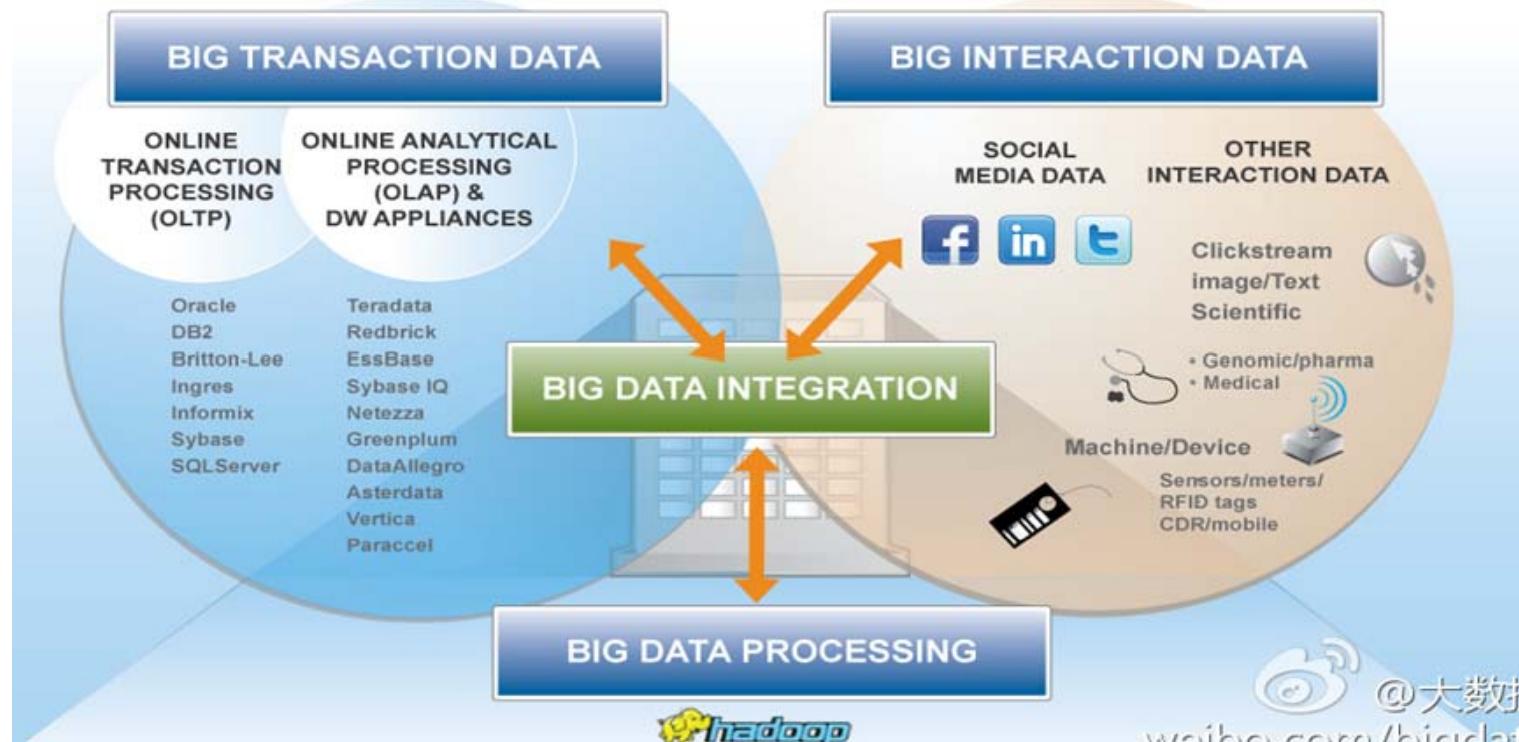


“Big Data” is data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it...



What is Big Data?

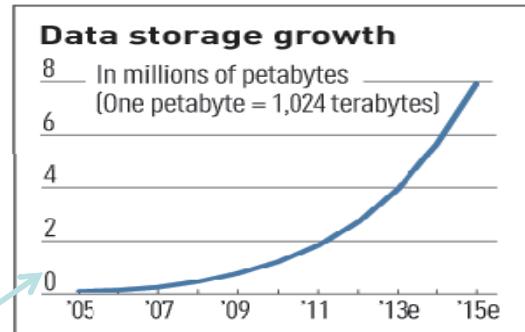
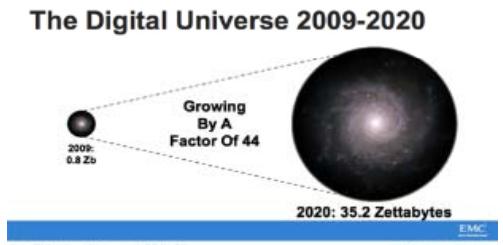
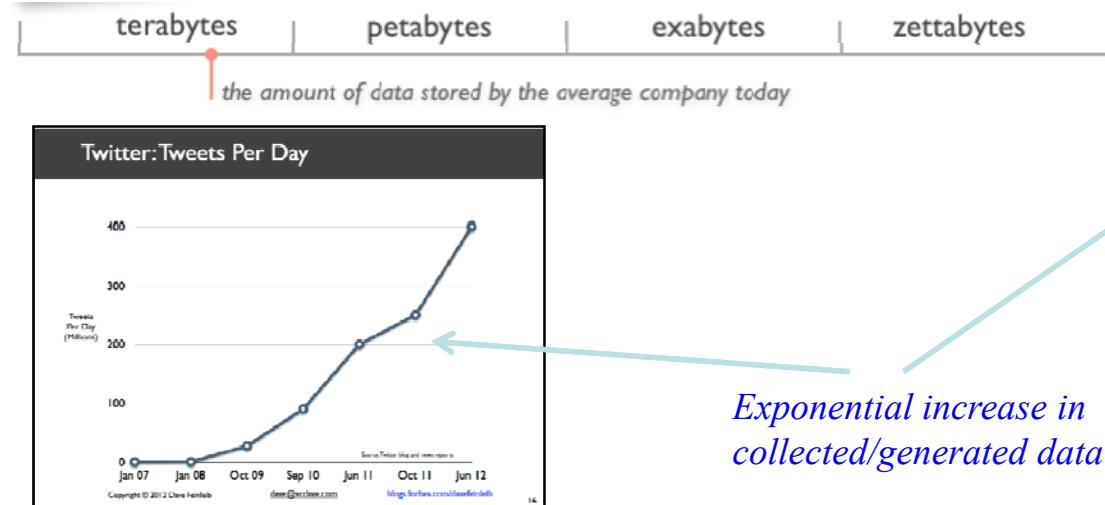
Big data is the confluence of the three trends consisting of Big Transaction Data, Big Interaction and Big Data Processing.



What is Big Data? 3 Vs of Big Data

1-Scale (Volume)

- **Data Volume**
 - 44x increase from 2009 2020
 - From 0.8 zettabytes to 35zb
- **Data volume is increasing exponentially**



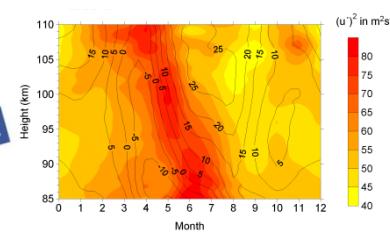
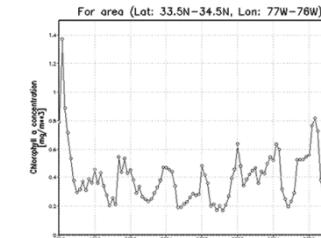
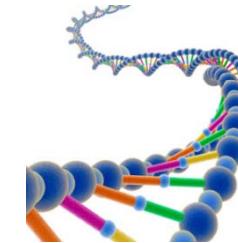
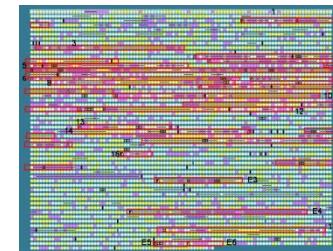
Exponential increase in collected/generated data

What is Big Data? 3 Vs of Big Data

2-Complexity (Variety)

- **Various formats, types, and structures**
- **Text, numerical, images, audio, video, sequences, time series, social media data, multi-dim arrays, etc...**
- **Static data vs. streaming data**
- **A single application can be generating/collecting many types of data**

To extract knowledge → all these types of data need to linked together



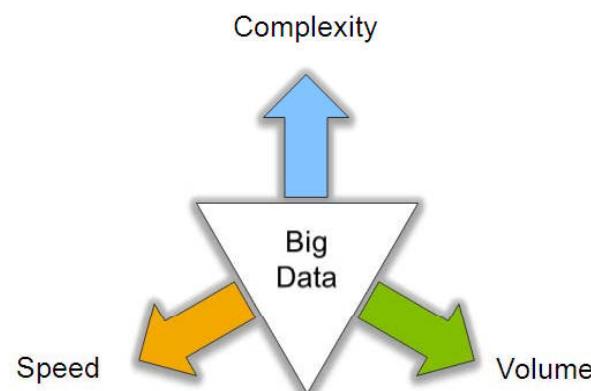
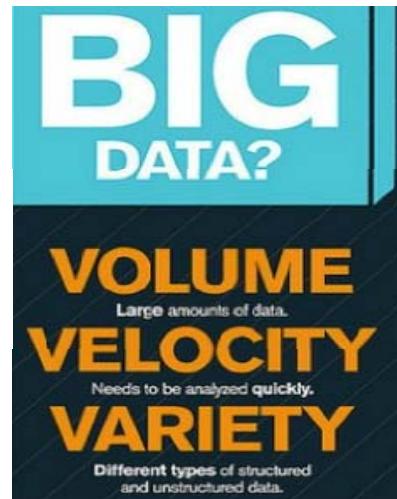
What is Big Data? 3 Vs of Big Data

3-Speed (Velocity)

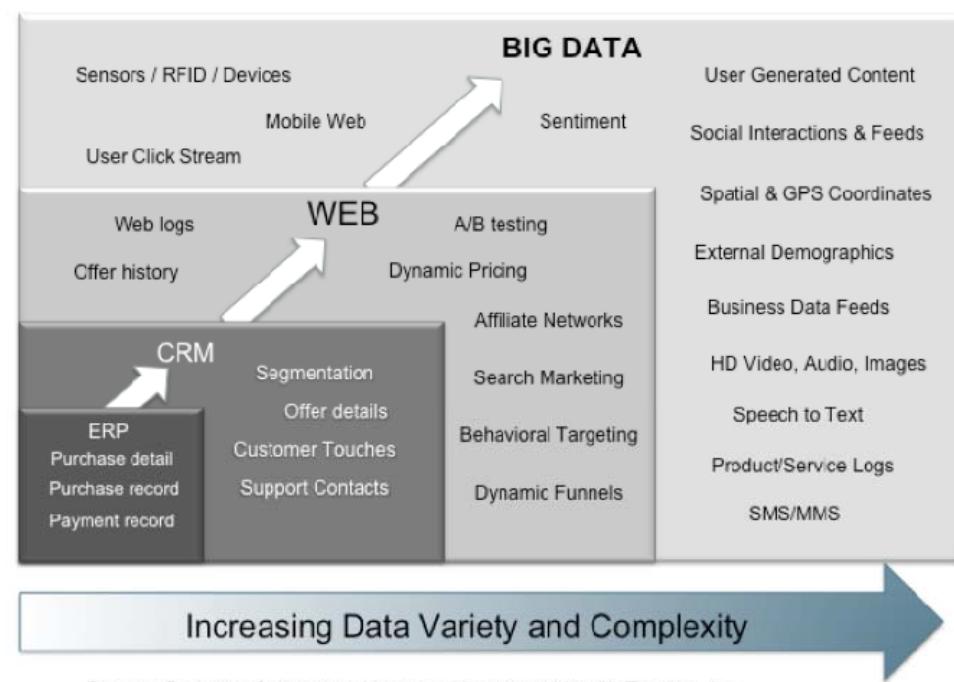
- **Data is generated fast and need to be processed fast**
- **Online Data Analytics**
- **Late decisions → missing opportunities**
- **Examples**
 - **E-Promotions:** Based on your current location, your purchase history, what you like → send promotions right now for store next to you
 - **Healthcare monitoring:** sensors monitoring your activities and body → any abnormal measurements require immediate reaction



What is Big Data? 3 Vs of Big Data



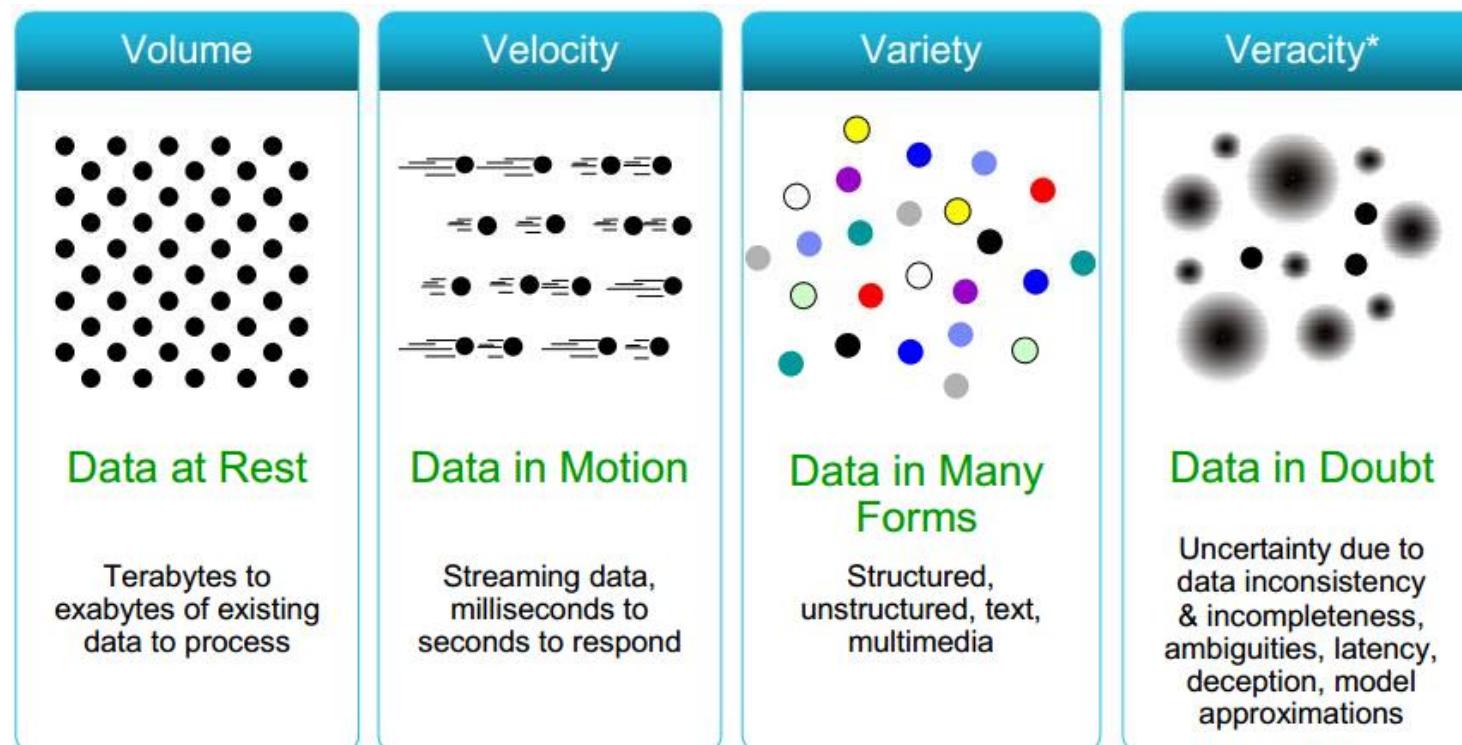
Big Data = Transactions + Interactions + Observations



Source: Contents of above graphic created in partnership with Teradata, Inc.

What is Big Data? 3 Vs of Big Data

Some Make it 4V's





What is Big Data?

Our world revolves around the data

- **Science**
 - Data bases from astronomy, genomics, environmental data, transportation data, ...
- **Humanities and Social Sciences**
 - Scanned books, historical documents, social interactions data, ...
- **Business & Commerce**
 - Corporate sales, stock market transactions, census, airline traffic, ...
- **Entertainment**
 - Internet images, Hollywood movies, MP3 files, ...
- **Medicine**
 - MRI & CT scans, patient records, ...
- **Industry, Energy, ...**
 - Sensors, ...



What is Big Data?

Who's Generating Big Data?



Social Media



Social media and networks

(all of us are generating data)



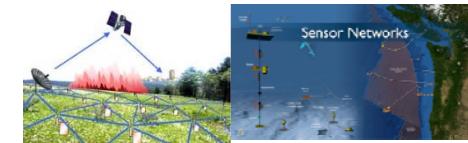
Scientific instruments

(collecting all sorts of data)



Mobile devices

(tracking all objects
all the time)



Sensor technology and networks

(measuring all kinds of data)

Big Data Analysis Example

Astronomy



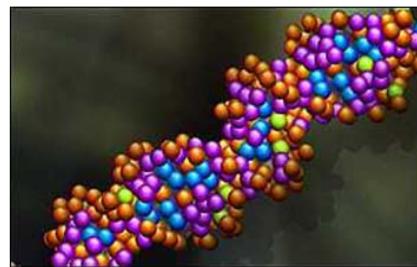
- Astronomical Data Mining
- Consuming habit

Credit card transactions



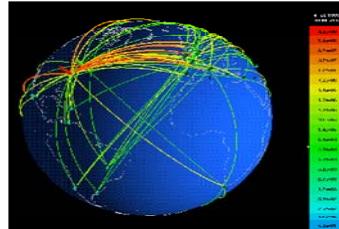
- 47.5 billion transactions in 2005 worldwide
- 115 Terabytes of data transmitted to VisaNet data processing center in 2004

Genomics



- 25,000 genes
- 3 billion base pairs
- 3 Gb

Internet traffic



- Traffic in a typical router:
- 42 kB/second
 - 3.5 Gigabytes/day
 - 1.3 Terabytes/year

Phone call billing records



- 250M calls/day
- 60G calls/year
- 40 bytes/call
- 2.5 Terabytes/year

SNA
Finding communities

The World-Wide Web



- 25 billion pages indexed
- 10kB/Page
- 250 Terabytes of indexed text data
- “Deep web” is supposedly 100 times as large

Contents

VIII. Imbalanced Big Data

What is Big Data?

Big Data. MapReduce

Hadoop and Mahout

Extremely Imbalanced Big Data:

A case of study

Big Data. MapReduce

Programming
Framework

■ Overview:

- Data-parallel programming model
- An associated parallel and distributed implementation for commodity clusters

■ Pioneered by Google

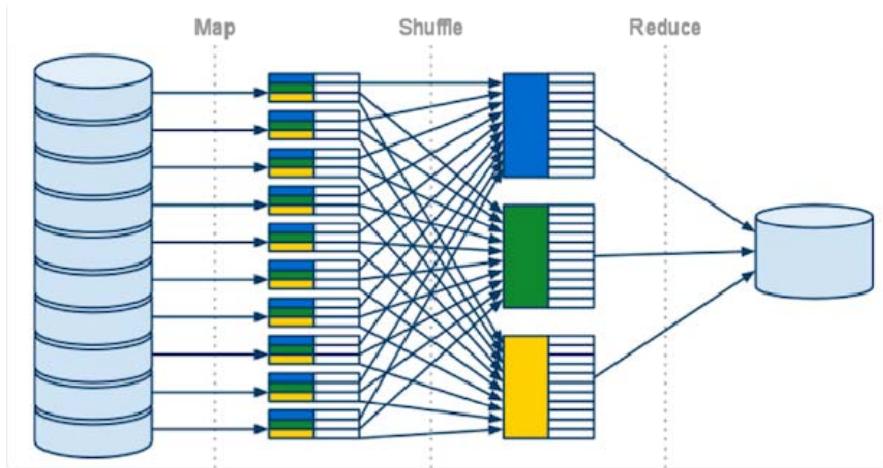
- Processes 20 PB of data per day

■ Popularized by open-source Hadoop project

- Used by Yahoo!, Facebook, Amazon, and the list is growing ...

Big Data. MapReduce

Programming
Framework



Raw Input: <key, value>

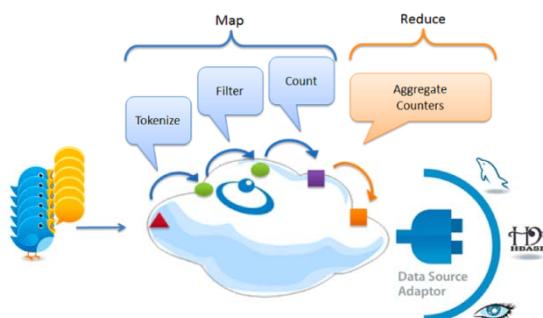
MAP

<K1, V1>

<K2, V2>

<K3, V3>

REDUCE



Big Data. MapReduce



How to process the data?

- **Automatic Parallelization:**
 - Depending on the size of RAW INPUT DATA → instantiate multiple MAP tasks
 - Similarly, depending upon the number of intermediate <key, value> partitions → instantiate multiple REDUCE tasks
- **Run-time:**
 - Data partitioning
 - Task scheduling
 - Handling machine failures
 - Managing inter-machine communication
- **Completely transparent to the programmer/analyst/user**

Big Data. MapReduce



Advantages

- **Scalability to large data volumes:**
 - Scan 100 TB on 1 node @ 50 MB/sec = 23 days
 - Scan on 1000-node cluster = 33 minutes
- ➔ Divide-And-Conquer (i.e., data partitioning)
- **Runs on large commodity clusters:**
 - 1000s to 10,000s of machines
- **Processes many terabytes of data**
- **Easy to use since run-time complexity hidden from the users**
- **Cost-efficiency:**
 - Commodity nodes (cheap, but unreliable)
 - Commodity network
 - Automatic fault-tolerance (fewer administrators)
 - Easy to use (fewer programmers)

Big Data.MapReduce



- MapReduce's data-parallel programming model hides complexity of distribution and fault tolerance
- Key philosophy:
 - *Make it scale*, so you can throw hardware at problems
 - *Make it cheap*, saving hardware, programmer and administration costs (but requiring fault tolerance)
- MapReduce is not suitable for all problems, but when it works, it may save you a lot of time

Contents

VIII. Imbalanced Big Data

What is Big Data?

Big Data. MapReduce

Hadoop and Mahout

Extremely Imbalanced Big Data:

A case of study

Big Data. Hadoop



**Hadoop implements the
computational paradigm
named MapReduce.**

Big Data. Hadoop

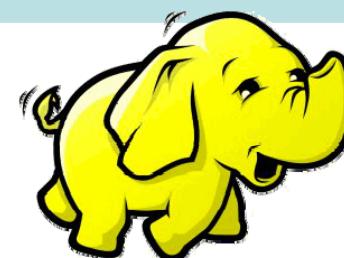
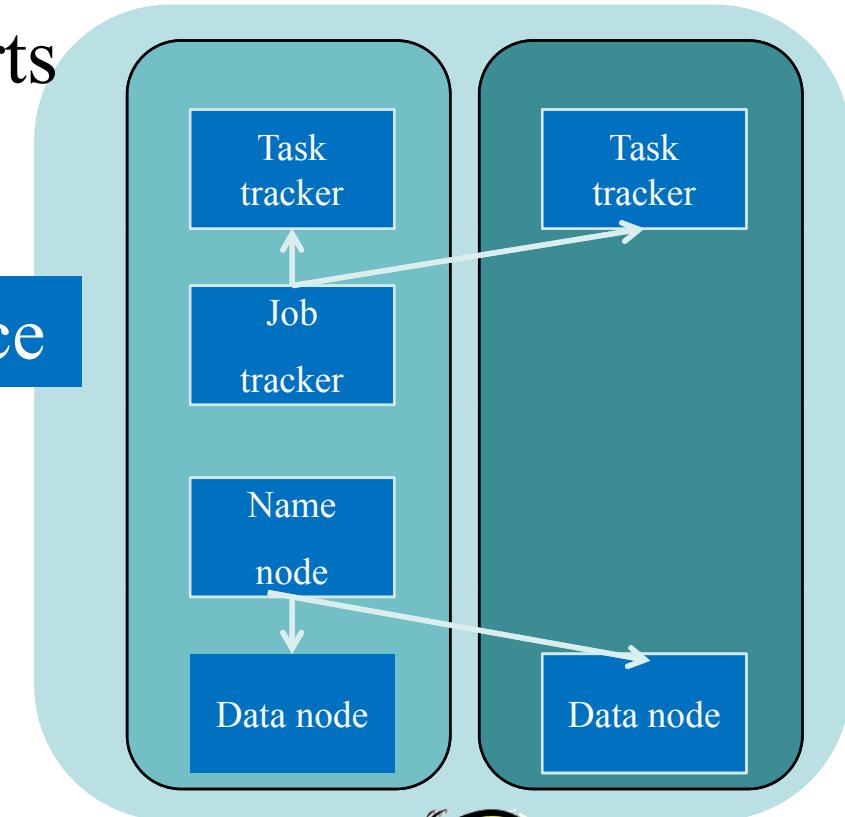


Apache Hadoop is an open-source software framework that supports data-intensive distributed applications, licensed under the Apache v2 license.

Map Reduce

HDFS

Created by **Doug Cutting** (chairman of board of directors of the Apache Software Foundation, 2010)



<http://hadoop.apache.org/>

Big Data. Hadoop



July 2008 - Hadoop Wins Terabyte Sort Benchmark

One of Yahoo's Hadoop clusters sorted 1 terabyte of data in 209 seconds, which beat the previous record of 297 seconds in the annual general purpose (Daytona) [terabyte sort benchmark](#). This is the first time that either a [Java](#) or an open source program has won.

What Is Apache Hadoop?

The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

<http://hadoop.apache.org/>

Big Data. Hadoop



The project

The project includes these modules:

- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.

Other Hadoop-related projects at Apache include:

- [**Avro™**](#): A data serialization system.
- [**Cassandra™**](#): A scalable multi-master database with no single points of failure.
- [**Chukwa™**](#): A data collection system for managing large distributed systems.
- [**HBase™**](#): A scalable, distributed database that supports structured data storage for large tables.
- [**Hive™**](#): A data warehouse infrastructure that provides data summarization and ad hoc querying.
- [**Mahout™**](#): A Scalable machine learning and data mining library.
- [**Pig™**](#): A high-level data-flow language and execution framework for parallel computation.
- [**ZooKeeper™**](#): A high-performance coordination service for distributed applications.

Big Data. Mahout



Scalable machine learning
and data mining



Apache Mahout has implementations of a wide range of machine learning and data mining algorithms:
clustering, classification, collaborative filtering and frequent pattern mining



Mahout currently has

- Collaborative Filtering
- User and Item based recommenders
- K-Means, Fuzzy K-Means clustering
- Mean Shift clustering
- Dirichlet process clustering
- Latent Dirichlet Allocation
- Singular value decomposition

- Parallel Frequent Pattern mining
- Complementary Naive Bayes classifier
- Random forest decision tree based classifier
- High performance [java](#) collections (previously colt collections)
- A vibrant community
- and many more cool stuff to come by this summer thanks to Google summer of code

<http://mahout.apache.org/>

Contents

VIII. Imbalanced Big Data

What is Big Data?

Big Data. MapReduce

Hadoop and Mahout

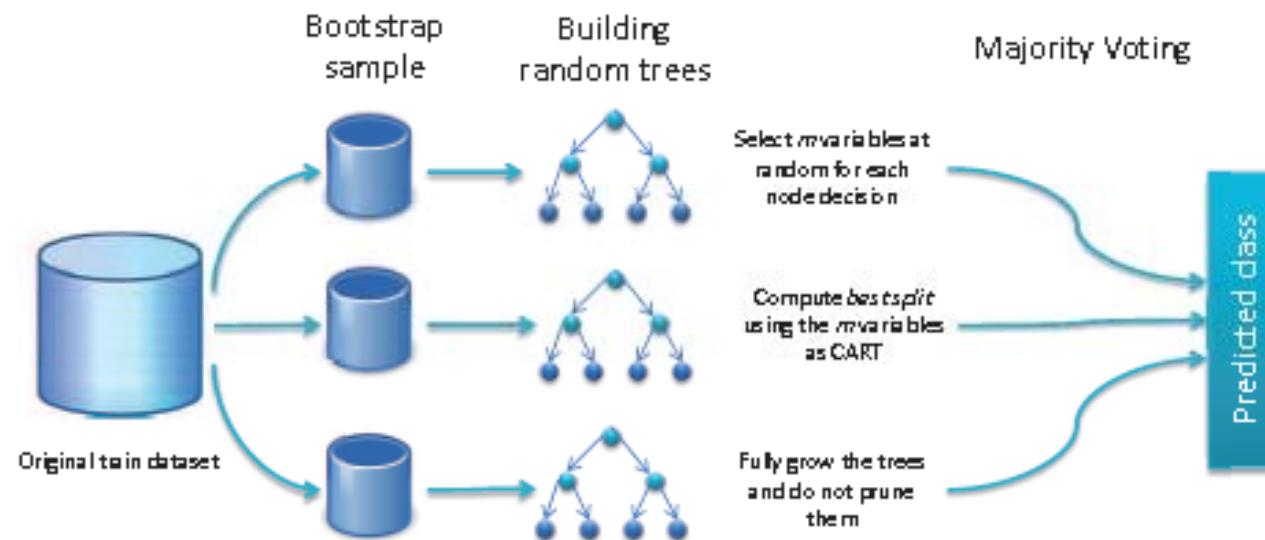
Extremely Imbalanced Big Data:

A case of study

Applying Cost-Sensitive Learning to Enhance Extremely
Imbalanced Big Data Problems using Random Forest

Extremely Imbalanced Big Data

Random Forest (RF): The predicted class is computed by aggregating the predictions of the ensembles through majority voting.

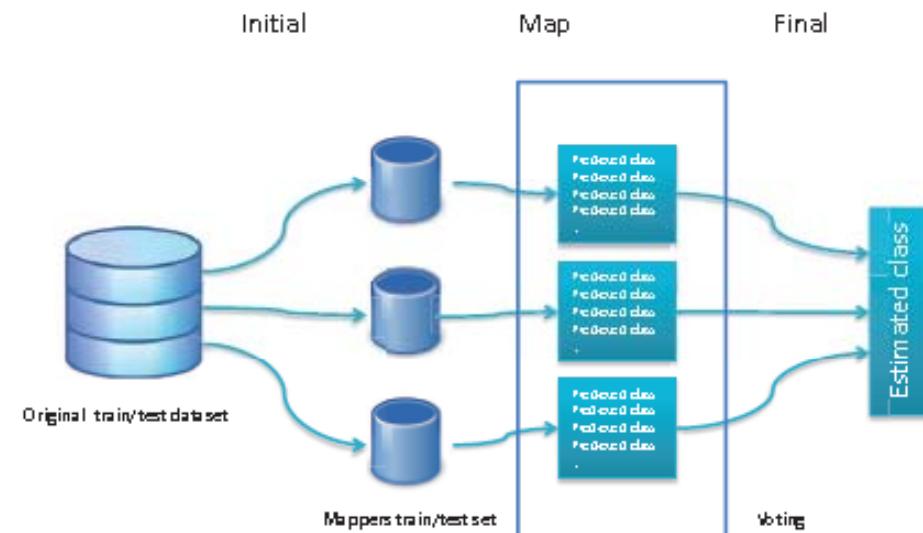
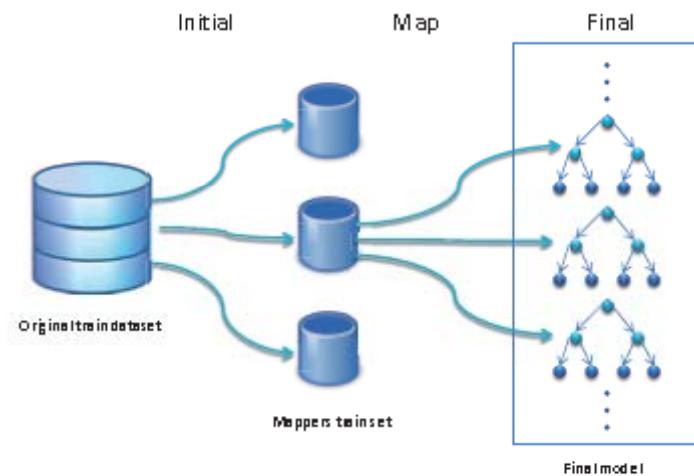


Extremely Imbalanced Big Data

The **RF Mahout Partial implementation** : is an algorithm that builds multiple trees for different portions of the data.

Two phases:

- Building phase
- Classification phase

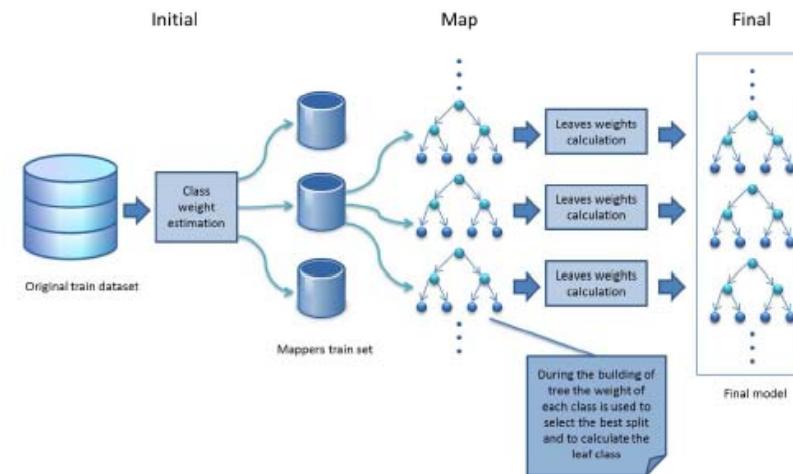


Extremely Imbalanced Big Data

Weighted Random Forest: A cost-sensitive learning based approach to deal with imbalanced data sets using RF.

Determination of the final class by aggregating the weighted vote using the weights of the leaf nodes.

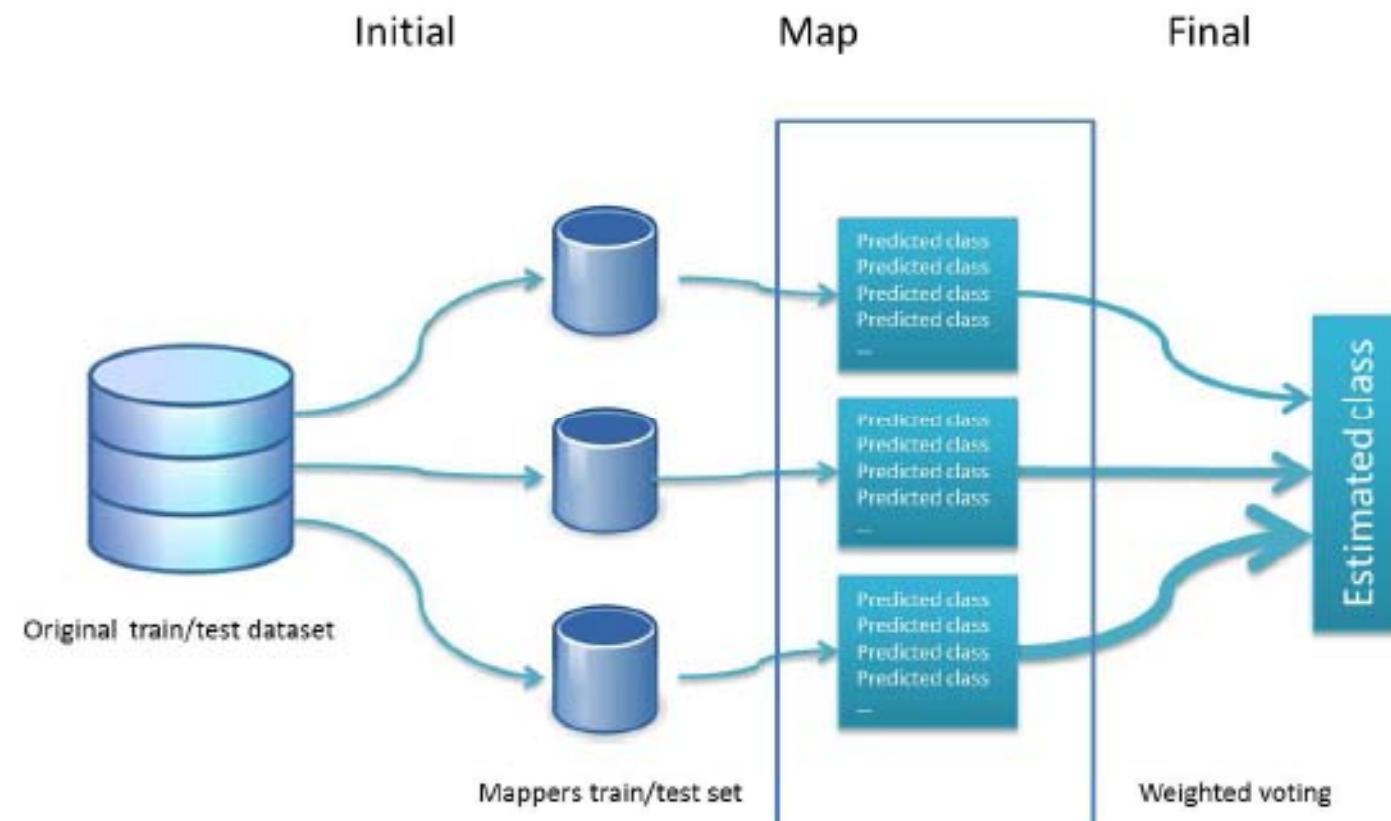
Building phase:



$$weightClass_i = \frac{n_{majorityClass}}{n_j},$$

Extremely Imbalanced Big Data

Classification phase:



Extremely Imbalanced Big Data

- 22 cases of study from two real-world imbalanced problem:
 - Derived from the KDD Cup 1999 Dataset.
 - Record Linkage Comparison Patterns Dataset.
- 10-fold stratified cross-validation model.
- Available at UCI Machine Learning Repository.

Datasets	#Ex.	#Atts.	Class (maj:min)	%Class(maj: min)	IR
kddcup_10_DOS_versus_normal	488736	41	(DOS; normal)	(80.10,19.90)	4.02
kddcup_10_DOS_versus_PRB	395565	41	(DOS; PRB)	(98.96,1.04)	95.31
kddcup_10_DOS_versus_R2L	392577	41	(DOS; R2L)	(99.71,0.29)	349.83
kddcup_10_DOS_versus_U2R	391517	41	(DOS; U2R)	(99.98,0.02)	6634.88
kddcup_10_normal_versus_PRB	101385	41	(normal; PRB)	(95.95,4.05)	23.69
kddcup_10_normal_versus_R2L	98397	41	(normal; R2L)	(98.86,1.14)	86.93
kddcup_10_normal_versus_U2R	97337	41	(normal; U2R)	(99.94,0.06)	1648.78
kddcup_50_DOS_versus_normal	2428075	41	(DOS; normal)	(79.97,20.03)	3.99
kddcup_50_DOS_versus_PRB	1962236	41	(DOS; PRB)	(98.95,1.05)	94.48
kddcup_50_DOS_versus_R2L	1942248	41	(DOS; R2L)	(99.97,0.03)	3448.82
kddcup_50_DOS_versus_U2R	1941711	41	(DOS; U2R)	(99.99,0.01)	74680.19
kddcup_50_normal_versus_PRB	506941	41	(normal; PRB)	(95.95,4.05)	23.67
kddcup_50_normal_versus_R2L	488953	41	(normal; R2L)	(99.88,0.12)	863.93
kddcup_50_normal_versus_U2R	486416	41	(normal; U2R)	(99.99,0.01)	18707.31
kddcup_fullDOS_versus_normal	4856151	41	(DOS; normal)	(79.97,20.03)	3.99
kddcup_fullDOS_versus_PRB	3924472	41	(DOS; PRB)	(98.95,1.05)	94.48
kddcup_fullDOS_versus_R2L	3884496	41	(DOS; R2L)	(99.97,0.03)	3448.82
kddcup_fullDOS_versus_U2R	3888422	41	(DOS; U2R)	(99.99,0.01)	74680.19
kddcup_full_normal_versus_PRB	1619009	41	(normal; PRB)	(95.95,4.05)	23.67
kddcup_full_normal_versus_R2L	973907	41	(normal; R2L)	(99.88,0.12)	863.93
kddcup_full_normal_versus_U2R	972833	41	(normal; U2R)	(99.99,0.01)	18707.33
RLCP	5749132	4	(TRUE; FALSE)	(99.64,0.36)	273.67

Extremely Imbalanced Big Data

- Algorithms:
 - **Random Forest** (RF): available on the Weka Data Mining Tool.
 - **Cost-Sensitive Random Forest** (RF-CS): adapted version using the previous algorithm as base.
 - **Big Data Random Forest** (RF-BigData): adapted version of the original Random Forest algorithm available on the Mahout library.
- Parameter specification:

Algorithm	Parameters
RF	maxDepth = unlimited, numFeatures = $\log_2(N_{vars}) + 1$, numTrees = 100
RF-CS	maxDepth = unlimited, numFeatures = $\log_2(N_{vars}) + 1$, numTrees = 100, costEstimation = weightBased
RF-BigData	maxDepth = unlimited, numFeatures = $\log_2(N_{vars}) + 1$, numPartitions = 5/10/20 numTrees = 20/10/5 (respectively)
RF-BigDataCS	maxDepth = unlimited, numFeatures = $\log_2(N_{vars}) + 1$, numPartitions = 5/10/20 numTrees = 20/10/5 (respectively), costEstimation = weightBased

Extremely Imbalanced Big Data

Average results using the GM measure:

Datasets	kddcup_full_DOS_versus_U2R		kddcup_full_normal_versus_R2L		kddcup_full_normal_versus_U2R	
	GM _{tr}	GM _{tst}	GM _{tr}	GM _{tst}	GM _{tr}	GM _{tst}
Sequential versions						
RF	NC	NC	1.0000	0.9832	0.9999	0.6836
RF-CS	NC	NC	0.9998	0.9976	0.9999	0.9813
Big data versions						
RF-BigData – 5 parts	0.7610	0.6731	0.9482	0.9376	0.3565	0.3333
RF-BigDataCS – 5 parts	0.8278	0.9608	0.9812	0.9835	0.9433	0.9960
RF-BigData - 10 parts	0.7045	0.6756	0.9274	0.9261	0.0000	0.0000
RF-BigDataCS – 10 parts	0.8032	0.9822	0.9793	0.9894	0.9381	0.9691
RF-BigData - 20 parts	0.0267	0.0000	0.8841	0.8767	0.0000	0.0000
RF-BigDataCS – 20 parts	0.9186	0.8853	0.9719	0.9657	0.9373	0.9593

Extremely Imbalanced Big Data

Datasets	RLCP	
	GM_{tr}	GM_{tst}
Sequential versions		
RF	NC	NC
RF-CS	NC	NC
Big data versions		
RF-BigData - 10 parts	0.2171	0.2052
RF-BigDataCS – 10 parts	0.9747	0.9201
RF-BigData - 20 parts	0.0572	0.0505
RF-BigDataCS – 20 parts	0.9661	0.9191

Extremely Imbalanced Big Data

Time elapsed (seconds) for sequential versions:

Datasets	RF			RF-CS		
	10%	50%	full	10%	50%	full
DOS_versus_normal	6344.42	49134.78	NC	5144.69	42328.64	NC
DOS_versus_PRB	4825.48	28819.03	NC	4173.99	30871.02	NC
DOS_versus_R2L	4454.58	28073.79	NC	3811.26	33226.90	NC
DOS_versus_U2R	3848.97	24774.03	NC	4240.18	35352.57	NC
normal_versus_PRB	468.75	6011.70	NC	558.38	5630.37	NC
normal_versus_R2L	364.66	4773.09	14703.55	357.69	5494.44	15324.86
normal_versus_U2R	295.64	4785.66	14635.36	360.05	5160.40	13919.74

Time elapsed (seconds) for Big data versions with 20 partitions:

Datasets	RF-BigData			RF-BigDataCS		
	10%	50%	full	10%	50%	full
DOS_versus_normal	98	221	236	177	641	701
DOS_versus_PRB	100	186	190	168	504	555
DOS_versus_R2L	97	157	136	164	469	288
DOS_versus_U2R	93	134	122	159	445	214
normal_versus_PRB	94	58	72	143	70	99
normal_versus_R2L	92	39	69	140	70	96
normal_versus_U2R	93	52	64	139	218	214

Big Data. Concluding Remarks

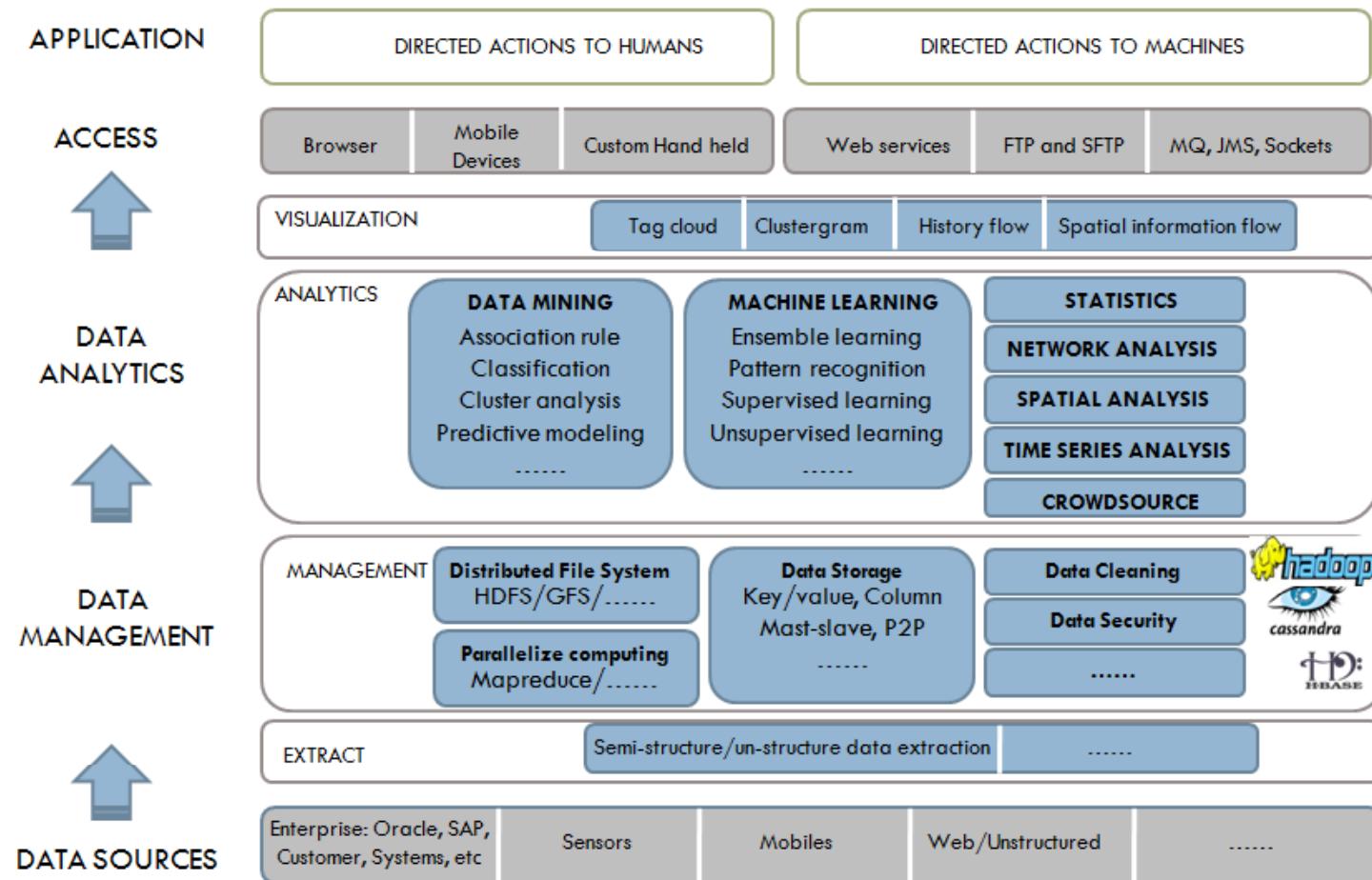


- The results obtained demonstrates that it is necessary to address jointly both the big data and the imbalanced issues as the different techniques isolated are not able to completely solve the problem.
- This is an initial approach, and there are a lot of open issues for imbalanced bigdata: **how to preprocess the data? How to deal with intrinsic data characteristics? How to deal with noise/missing values ...?**

Big Data. Concluding Remarks



The Framework of Big Data



Contents

- I. Introduction to imbalanced data sets
- II. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance
- VII. Multiple class imbalanced data-sets: A pairwise learning approach
- VIII. Imbalanced Big Data
- IX. **Class imbalance: Data sets, implementations, ...**
- X. **Class imbalance: Trends and final comments**

SESSION 1

SESSION 2

SESSION 3

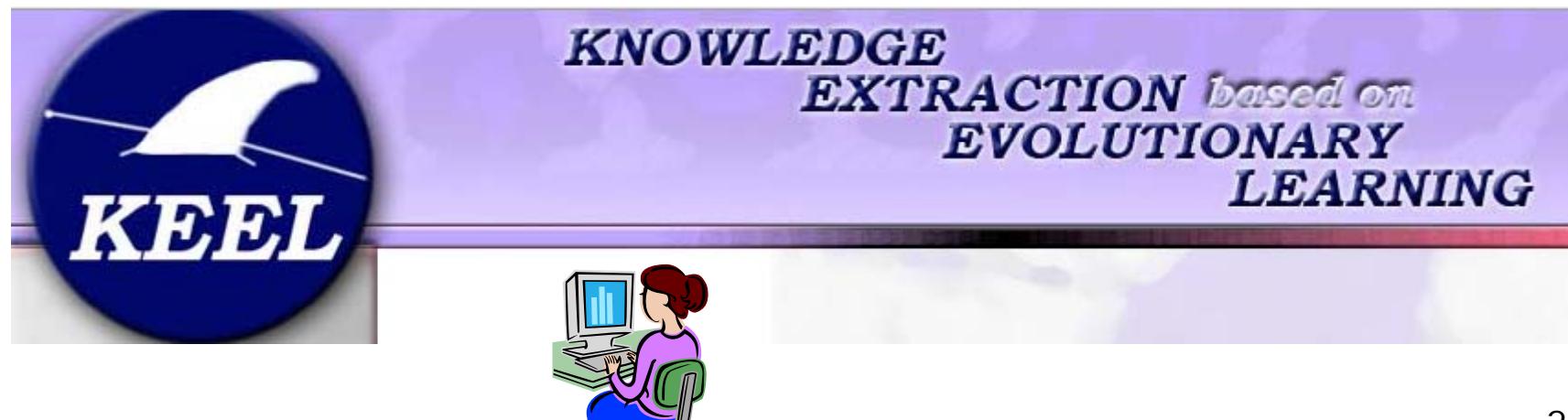
Class Imbalance: Data sets, implementations,

...

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



<http://www.keel.es>



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.



<i>Imbalanced Classification (42)</i>	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 borderline.



Imbalanced data sets

KEEL-dataset
Data set repository



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. Resampling the original training set
- III. Resampling: Some results on the use of evolutionary prototype selection for imbalanced data sets
- IV. Cost Modifying: Cost-sensitive learning
- V. Why is difficult to learn in imbalanced domains? Intrinsic data characteristics
- VI. Ensembles to address class imbalance
- VII. Multiple class imbalanced data-sets: A pairwise learning approach
- VIII. Imbalanced Big Data
- IX. Class imbalance: Data sets, implementations, ...
- X. **Class imbalance: Trends and final comments**

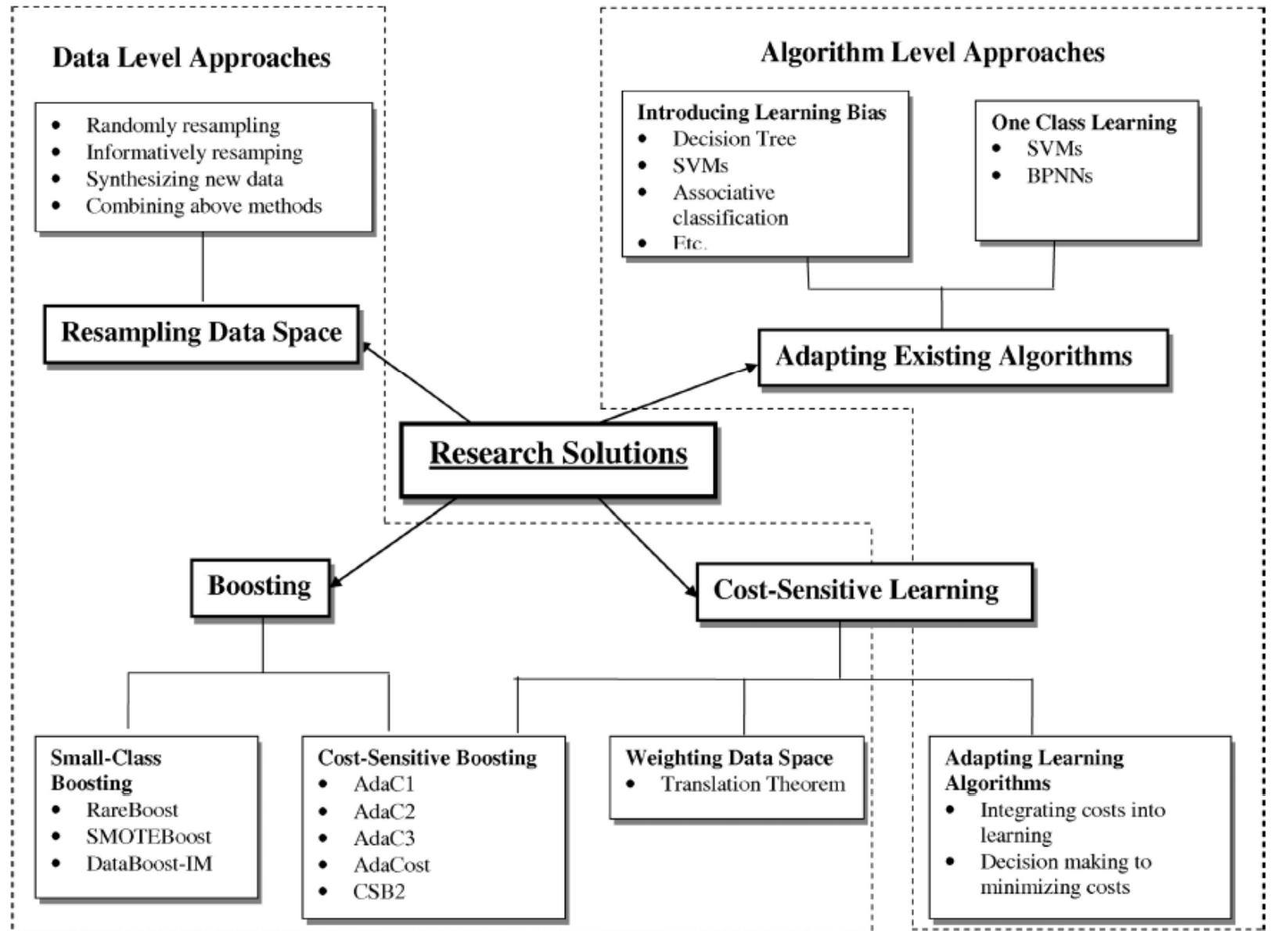
SESSION 1

SESSION 2

SESSION 3

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 approaches 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



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 (**fundamental**),
- 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.

→ Approaches for extremely imbalanced big data.



Class Imbalance: Trends and final comments

Final comments

- We have presented 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).**



Classification with Imbalanced Data Sets



Any Questions?



Classification with Imbalanced Data Sets



Thanks!!!

