UNIVERSITÀ DI PISA

Dottorato di Ricerca in Ingegneria dell'Informazione

# Data Mining and Soft Computing


Francisco Herrera

Research Group on Soft Computing and Information Intelligent Systems (SCI$^2$S)

Dept. of Computer Science and A.I.

University of Granada, Spain

Email: herrera@decsai.ugr.es

http://sci2s.ugr.es

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


DECSAI

Universidad de Granada

# Summary

1. **Introduction to Data Mining and Knowledge Discovery**
2. **Data Preparation**
3. **Introduction to Prediction, Classification, Clustering and Association**
4. **Data Mining - From the Top 10 Algorithms to the New Challenges**
5. **Introduction to Soft Computing. Focusing our attention in Fuzzy Logic and Evolutionary Computation**
6. **Soft Computing Techniques in Data Mining: Fuzzy Data Mining and Knowledge Extraction based on Evolutionary Learning**
7. **Genetic Fuzzy Systems: State of the Art and New Trends**
8. **Some Advanced Topics I: Classification with Imbalanced Data Sets**
9. **Some Advanced Topics II: Subgroup Discovery**
10. **Some advanced Topics III: Data Complexity**
11. **Final talk: How must I Do my Experimental Study? Design of Experiments in Data Mining/Computational Intelligence. Using Non-parametric Tests. Some Cases of Study.**

# Outline

- ✓ Introduction to Imbalanced Data Sets
- ✓ Some results on the use of evolutionary prototype selection for imbalanced data sets
- ✓ Class imbalance related topics:
  Cost-Sensitive Learning and anomaly detection
- ✓ Concluding Remarks

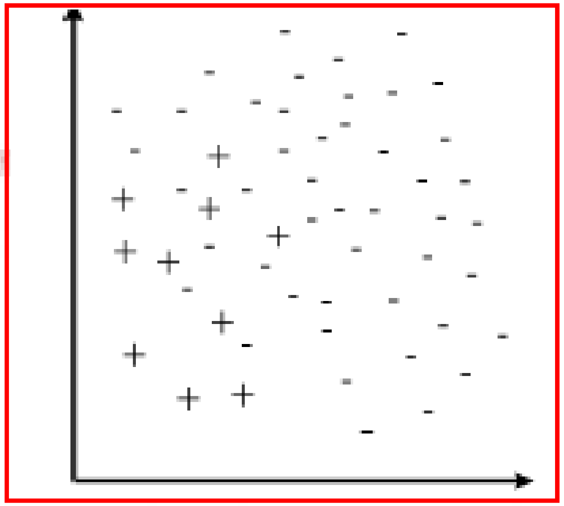
# Outline

- ✓ Introduction to Imbalanced Data Sets
- ✓ Some results on the use of evolutionary prototype selection for imbalanced data sets
- ✓ Class imbalance related topics:
  Cost-Sensitive Learning and anomaly detection
- ✓ Concluding Remarks

# 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. Such a situation poses challenges for typical classifiers such as decision tree induction systems or multi-layer perceptrons 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 cassifying the large ones accurately.

Such a problem occurs in a large number of practical domains and is often dealt with by using re-sampling or cost-based methods.

This talk introduce the "classification with imbalanced data sets" analyzing in depth the solutions based on re-sampling.

# Introduction to Imbalanced Datasets

**Learning in non-Balanced domains.**

**Data balancing through resampling.**

**State-of-the-art algorithm: *SMOTE.***

# Introduction to Imbalanced Datasets

**Learning in non-Balanced domains.**

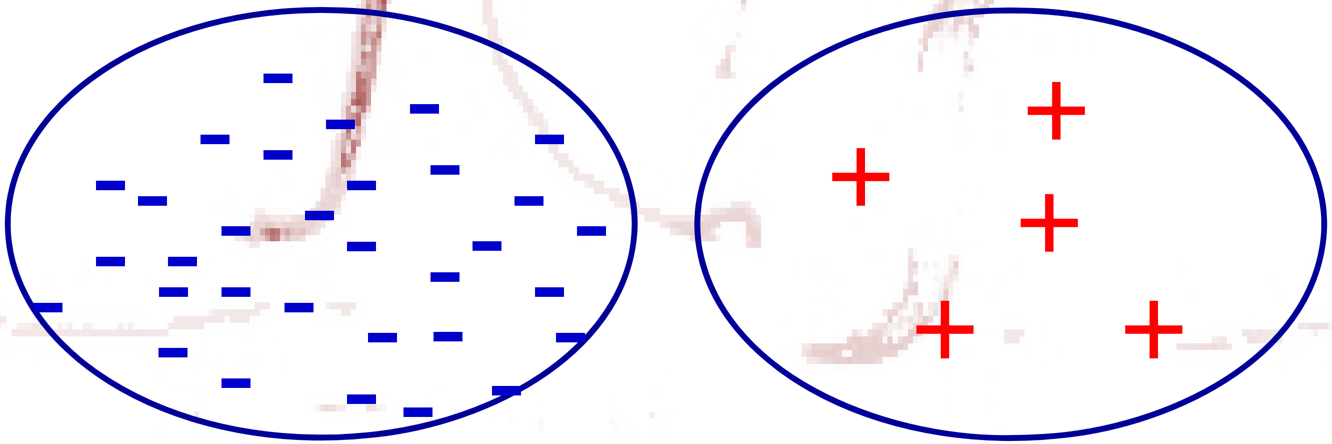
**Data balancing through resampling.**

**State-of-the-art algorithm: *SMOTE.***

# Learning in non-balanced domains

**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.**



G. Cohen, M. Hilario, H. Sax, S. Hugonnet, A. Geisbuhler. Learning from Imbalanced Data in Surveillance of Nosocomial Infection. Artificial Intelligence in Medicine 37 (2006) 7-18

# Learning in non-balanced domains

The classes of small size are usually labeled by rare cases (rarities).

**The most important knowledge usually resides in the rare cases.**

**These cases are common in classification problems:**

**Ej.: Detection of uncommon diseases.**

**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**

# Learning in non-balanced domains

## 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:

**As a result**

- **examples from the overwhelming class are well-classified**

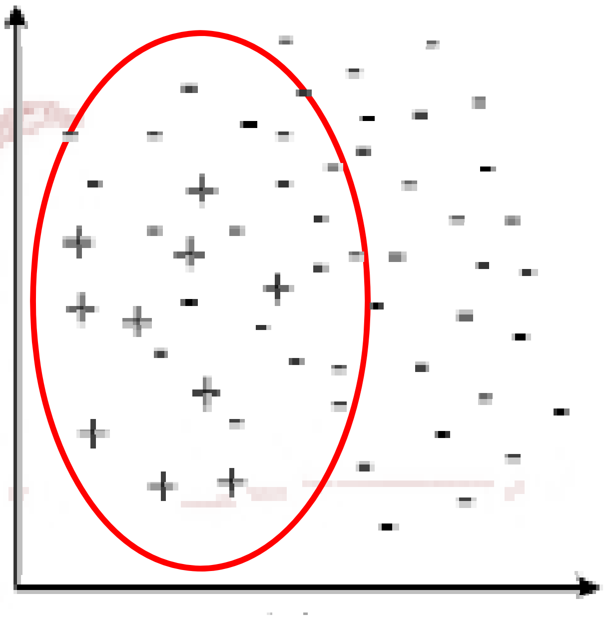
- **whereas examples from the minority class tend to be misclassified.**

# Learning in non-balanced domains

## ¿Why is difficult to learn in imbalanced domains?

**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.**



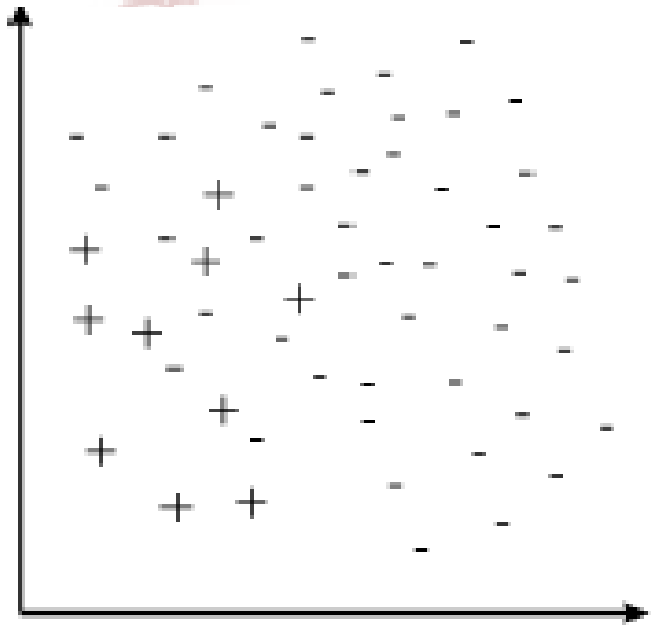
N.V. Chawla, N. Japkowicz, A. Kolcz. Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations 6:1 (2004) 1-6

# Learning in non-balanced domains

**Why Learning from Imbalanced Data Sets might be difficult?**

## Four Groups of Negative Examples

- **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**

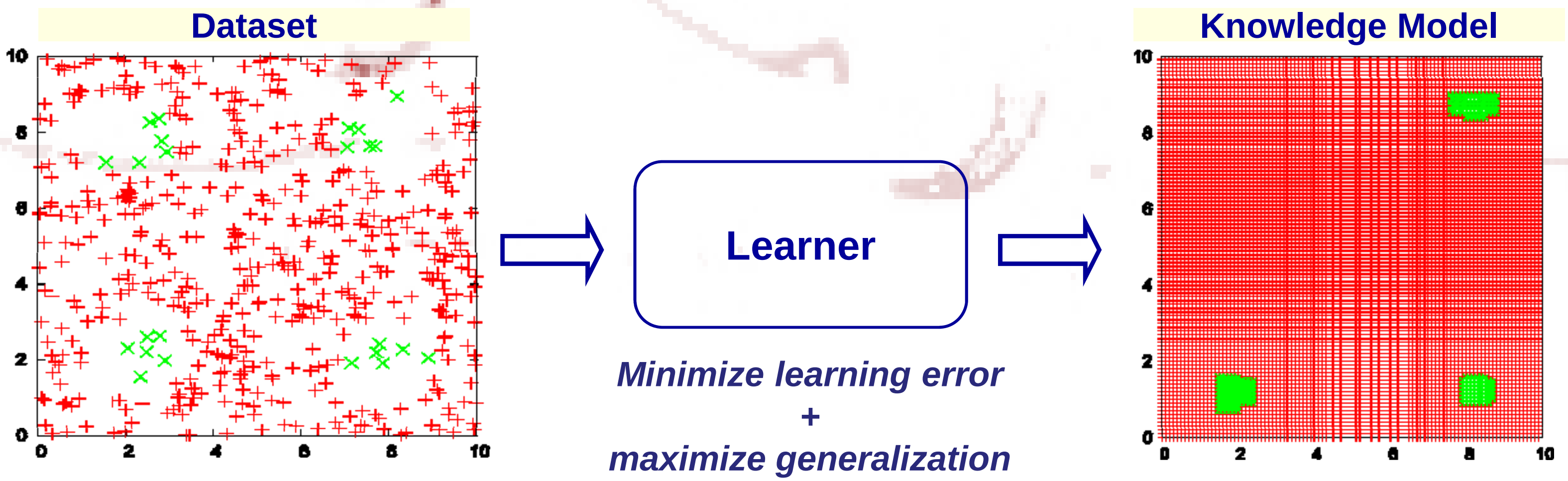


# Learning in non-balanced domains

## Why Learning from Imbalanced Data Sets might be difficult?

**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, reare cases are unknown since high dimensional data cannot be visualizad to reveal areas of low coverage.**

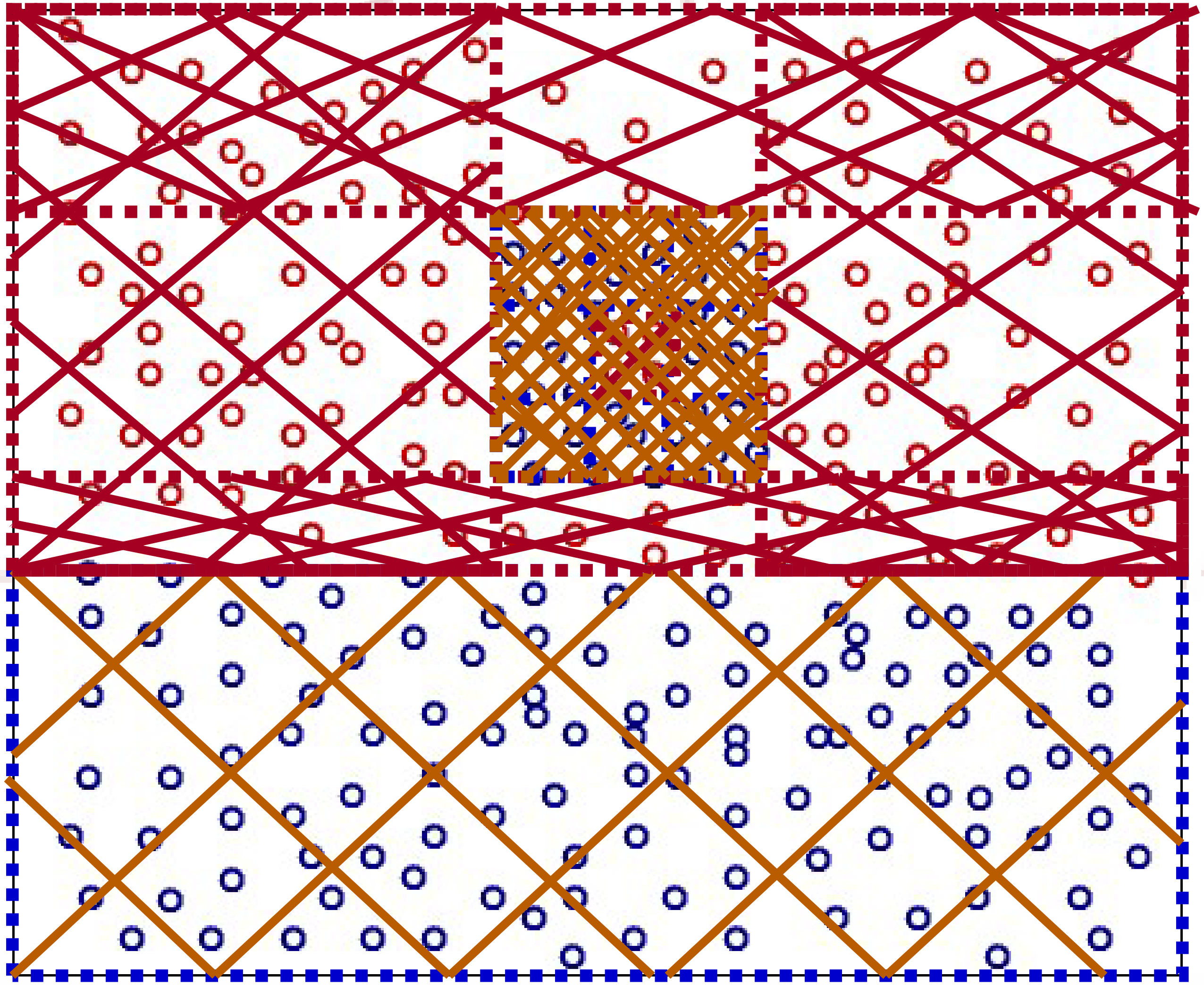



T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49

# Learning in non-balanced domains

**Why Learning from Imbalanced Data Sets might be difficult?**

**Small disjunct: Focusing the problem**



*Small Disjunct* or *Starved niche*

Again *more small disjuncts*

*Overgeneral Classifier*

# Learning in non-balanced domains

## ¿How can we evaluate an algorithm in imbalanced domains?

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

**Confusion matrix for a two-class problem**

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

**Classical evaluation:**

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

# Learning in non-balanced domains

**Imbalanced evaluation based on the geometric mean:**

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

Evaluation function: **True ratio**

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

Precision = TP/(TP+FP)
Recall = TP/(TP+FN)

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

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

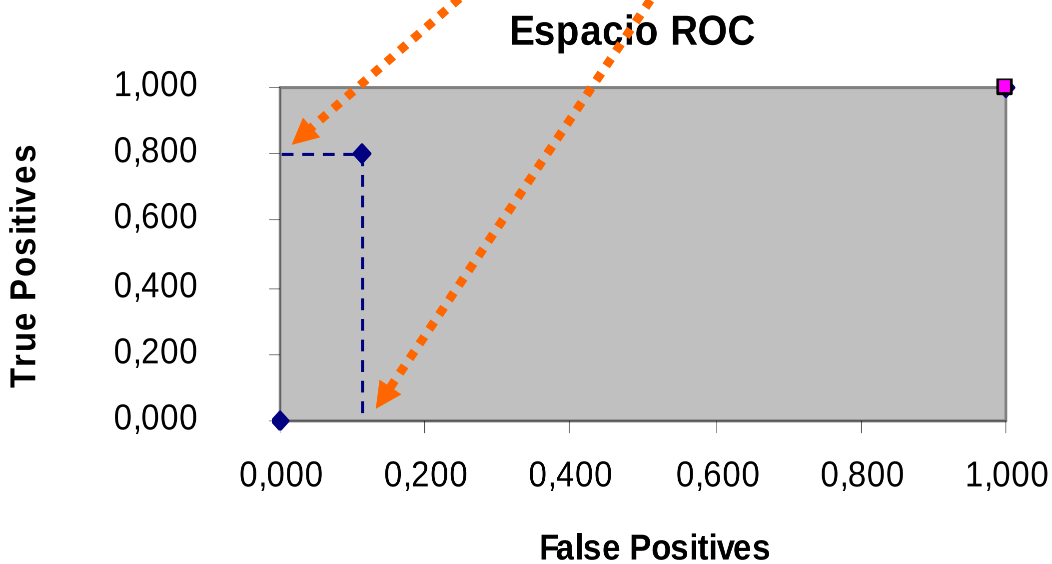
# Learning in non-balanced domains

## ROC Curves

**The confusion matrix is normalized by columns**

Real

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



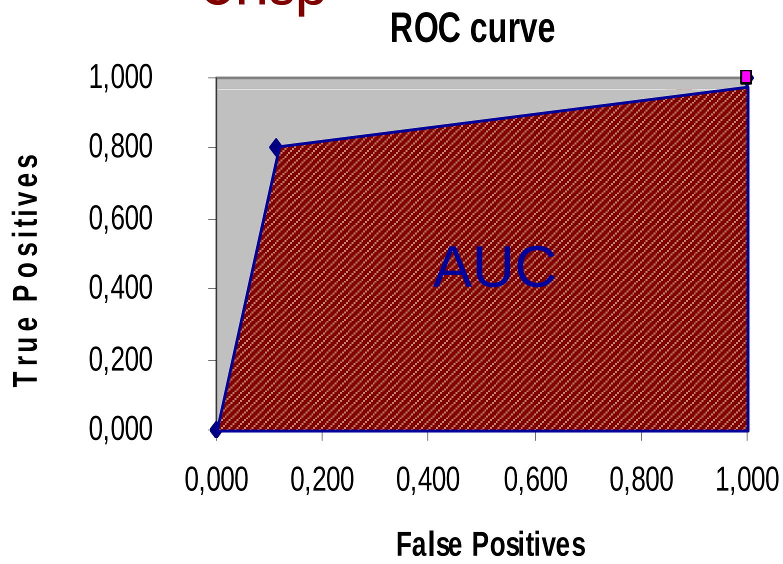

A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition 30(7) (1997) 1145-1159.

# Learning in non-balanced domains

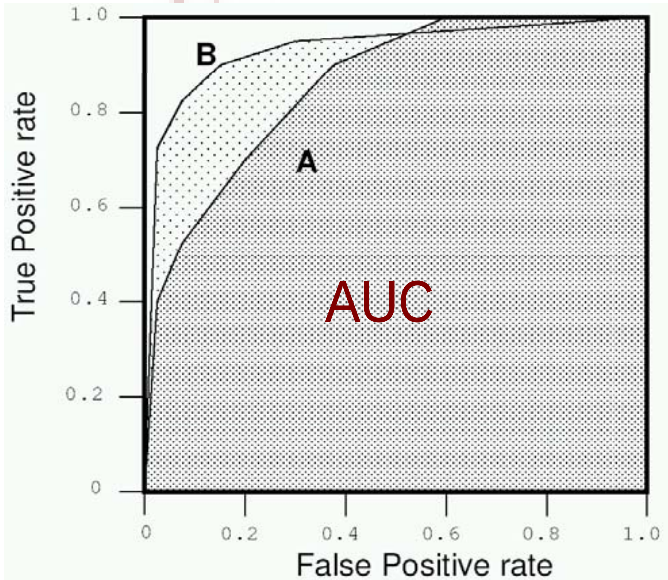
**"crisp" and "soft" classifiers:**

- **A "crisp" classifier (discrete) predicts a class among the candidates.**
- **A "soft" classifier (probabilistic) predicts a class, but this prediction is accompanied by a reliability value.**

Crisp




Soft




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

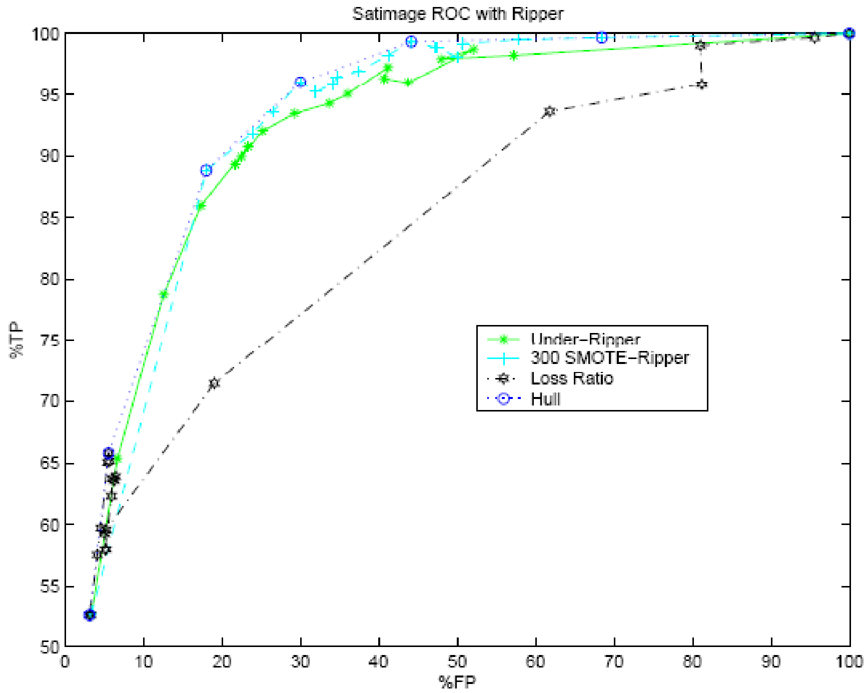
# Learning in non-balanced domains

## ROC analysis oriented to data resampling in imbalanced domains

The resampling algorithm must allow to adjust the rate of under/over sampling.

Performance of the classifier is measured with *over/under Sampling* at 25%, 50%, 100%, 200%, 300%, etc.

*It can be only used in resmapling techniques which allow the adjustment of this parameter.*




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

# Introduction to Imbalanced Datasets

**Learning in non-Balanced domains.**

**Data balancing through resampling.**

**State-of-the-art algorithm: *SMOTE.***

# Data Balancing through *re-sampling*

## Strategies tp deal with imbalanced data sets

**Over-Sampling**

- **Random**
- **Focused**

**Under-Sampling**

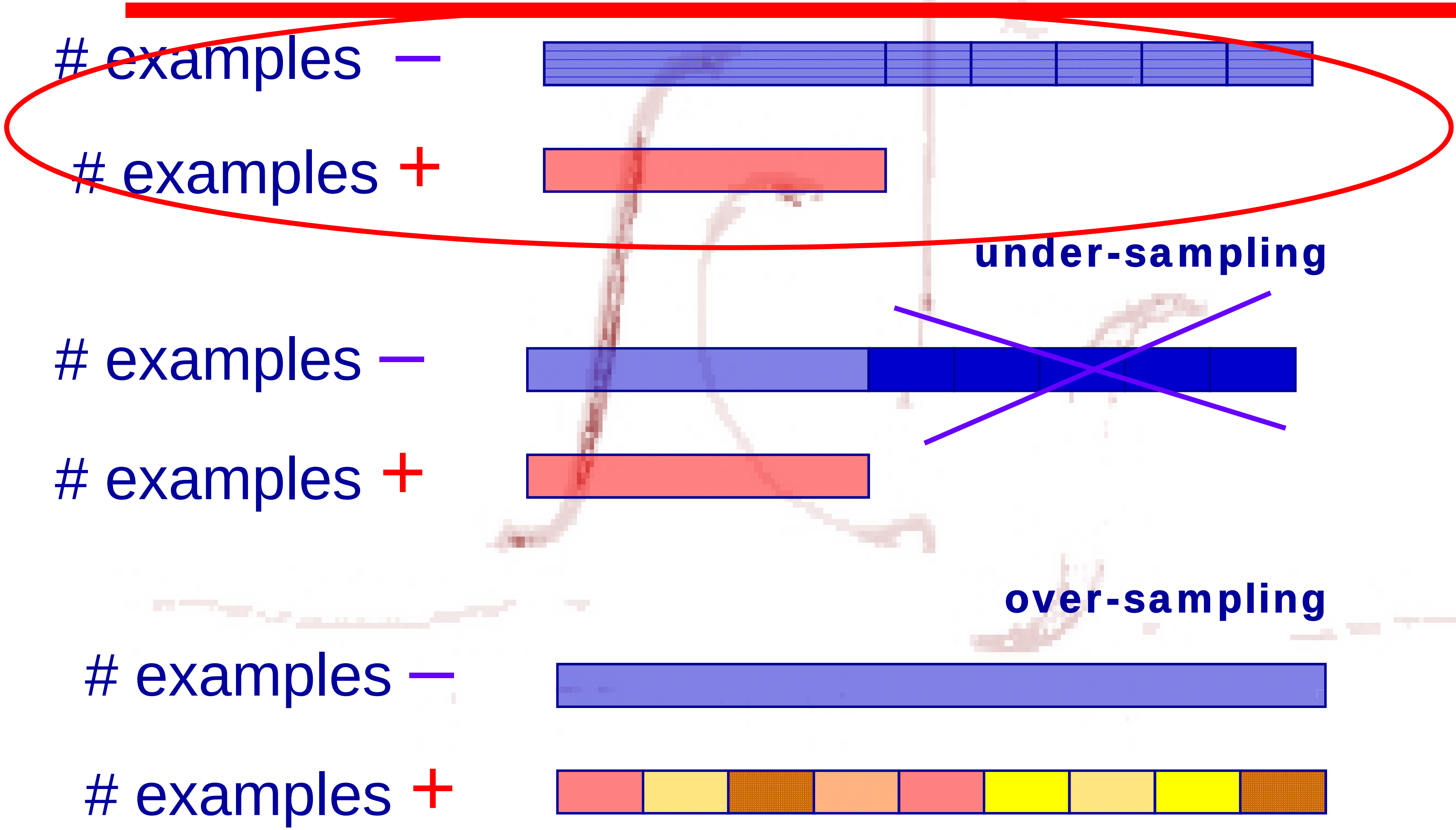
- **Random**
- **Focused**

Cost Modifying

**Motivation**

- **Retain influyent examples**
- **Balance the training set**
- **Remove noisy instances in the decision boundaries**
- **Reduce the training set**

# Data Balancing through *re-sampling*



# Data Balancing through *re-sampling*

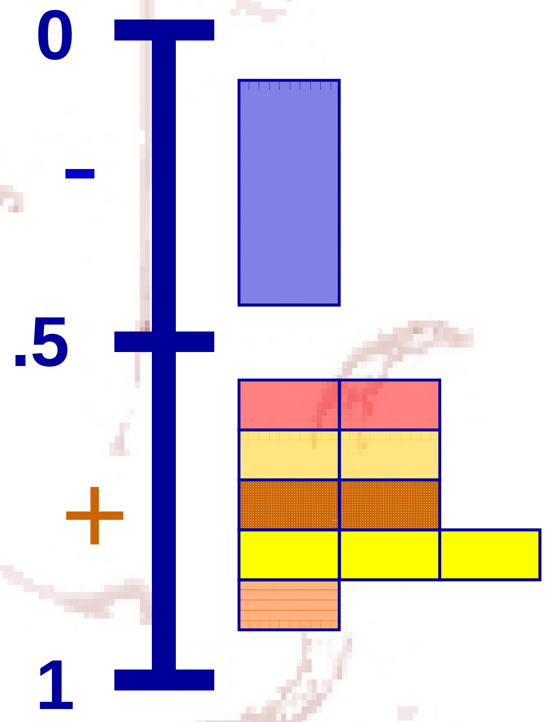
**Over Sampling**

- **Random**
- **Focused**

**Under Sampling**

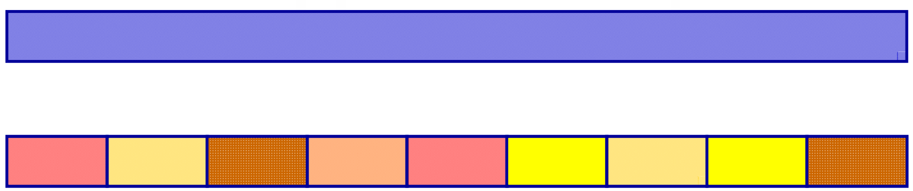
- **Random**
- **Focused**

**Cost Modifying**




# examples of -

# examples of +



# Data Balancing through *re-sampling*

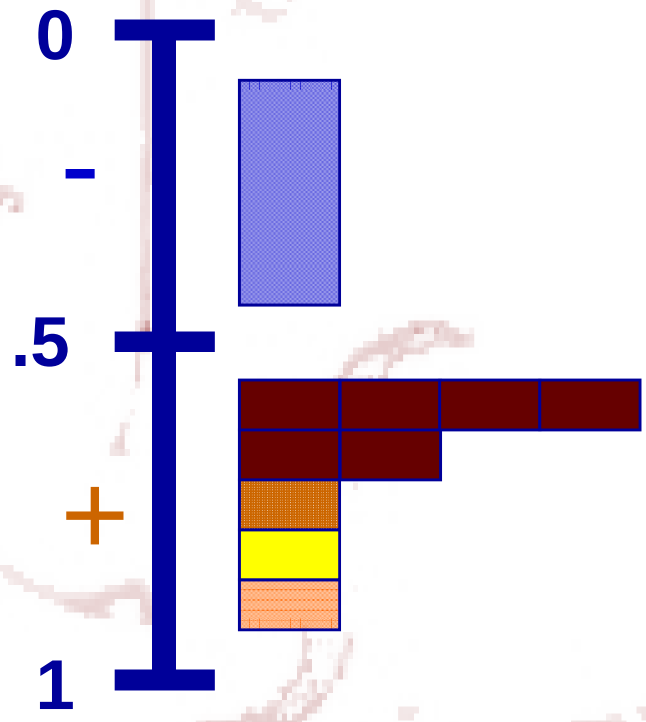
**Over Sampling**

- Random
- Focused

**Under Sampling**

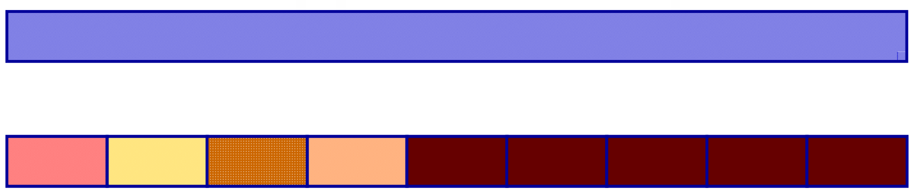
- Random
- Focused

**Cost Modifying**




# examples of -

# examples of +



# Data Balancing through *re-sampling*

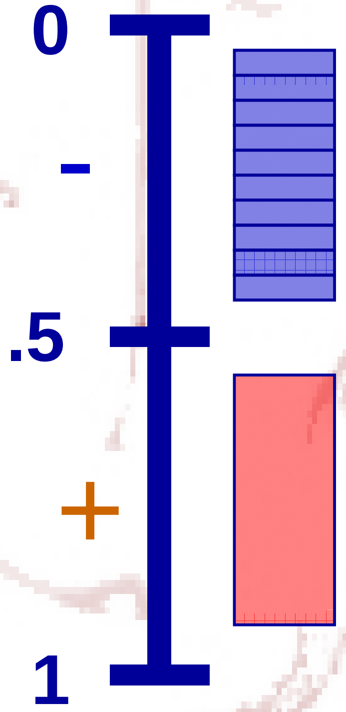
Over Sampling

- Random
- Focused

**Under Sampling**

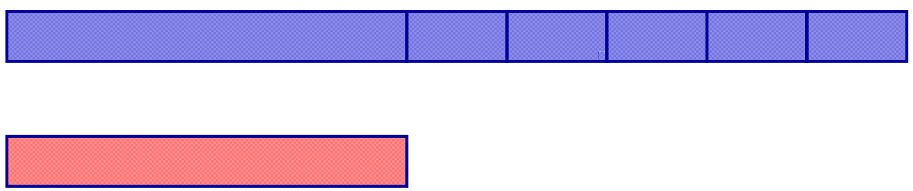
- **Random**
- Focused

Cost Modifying




\# examples of -

\# examples of +



# Data Balancing through *re-sampling*

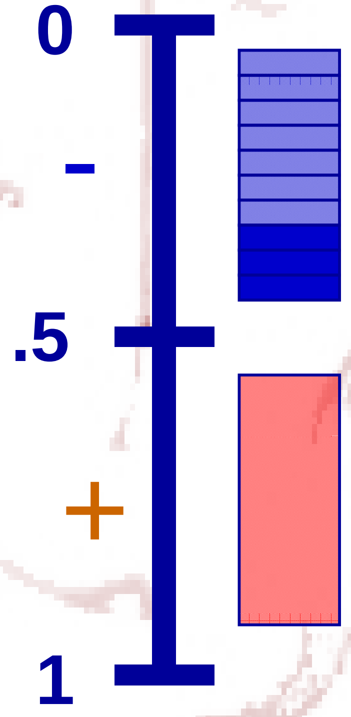
**Over Sampling**

- **Random**
- **Focused**

**Under Sampling**

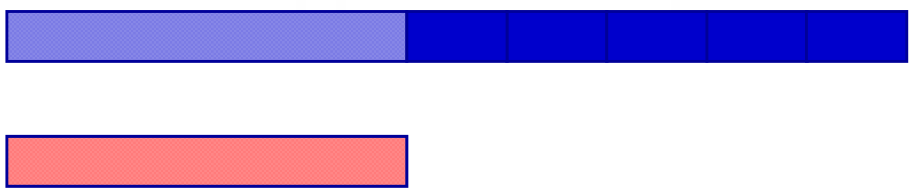
- **Random**
- **Focused**

**Cost Modifying**




# examples of -

# examples of +



# Data Balancing through *re-sampling*

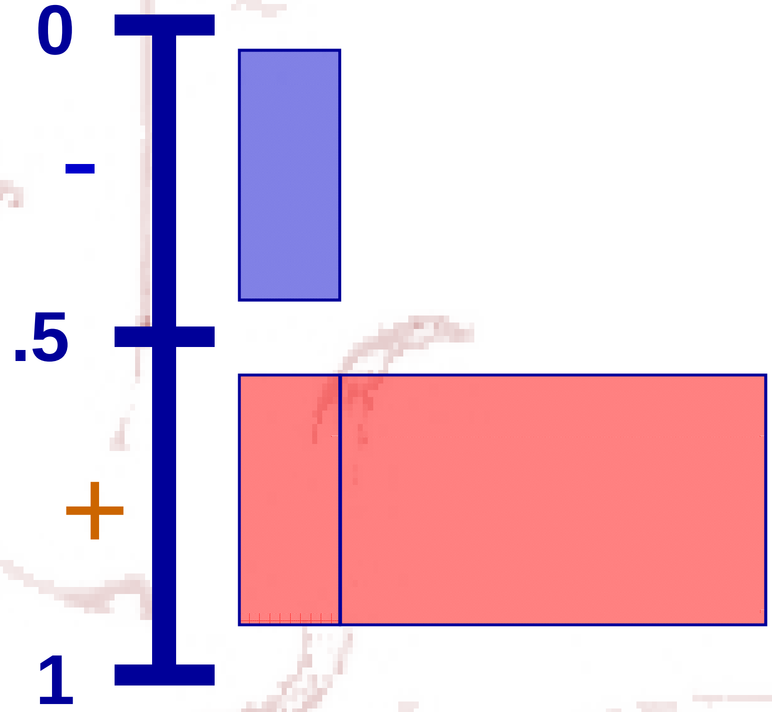
**Over Sampling**

- **Random**
- **Focused**

**Under Sampling**

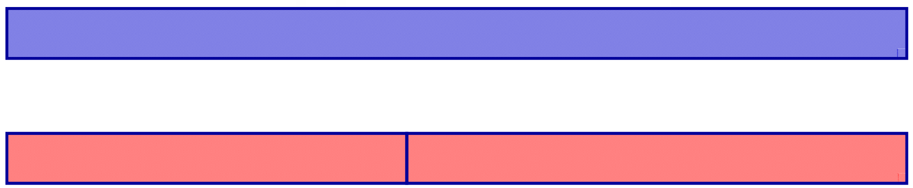
- **Random**
- **Focused**

**Cost Modifying**




# examples of -

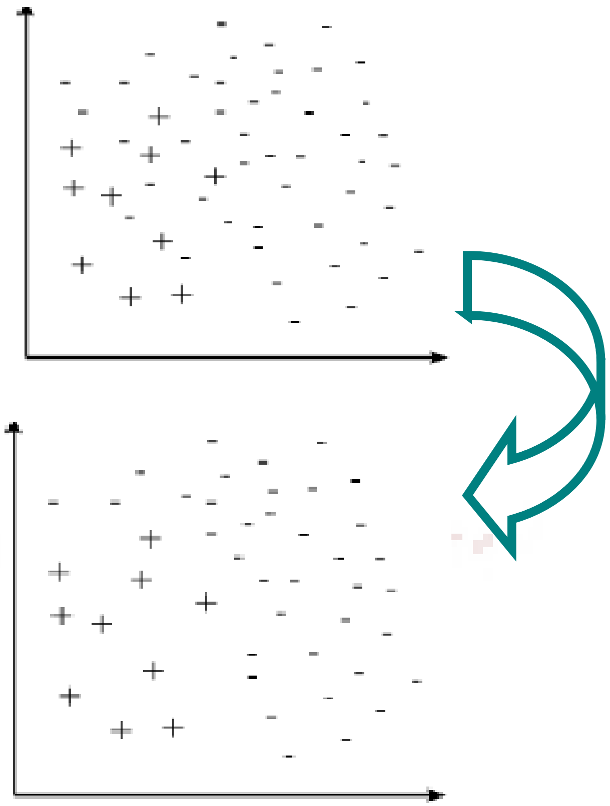
# examples of +



# Data Balancing through *re-sampling*

## **Under-sampling: Tomek Links**

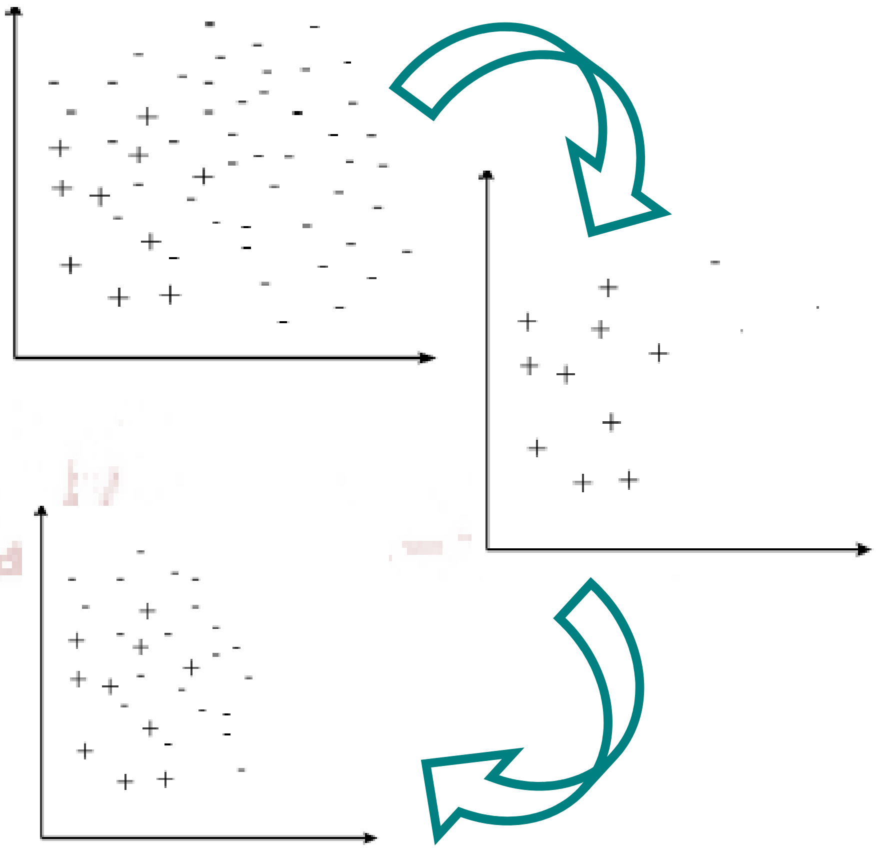
- To remove both noise and borderline examples of the majority class
- Tomek link
  - $E_i$, $E_j$ belong to different classes, d ($E_i$, $E_j$) is the distance between them.
  - A ($E_i$, $E_j$) pair is called a Tomek link if there is no example $E_l$, such that d($E_i$, $E_l$) < d($E_i$, $E_j$) or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



# Data Balancing through *re-sampling*

## **Under-sampling: US-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'



# Data Balancing through *re-sampling*

## 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

# Introduction to Imbalanced Datasets

**Learning in non-Balanced domains.**

**Data balancing through resampling.**

**State-of-the-art algorithm: *SMOTE.***

# State-of-the-art algorithm: SMOTE.

## Over-sampling method:

- 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.

- 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.

# State-of-the-art algorithm: SMOTE.

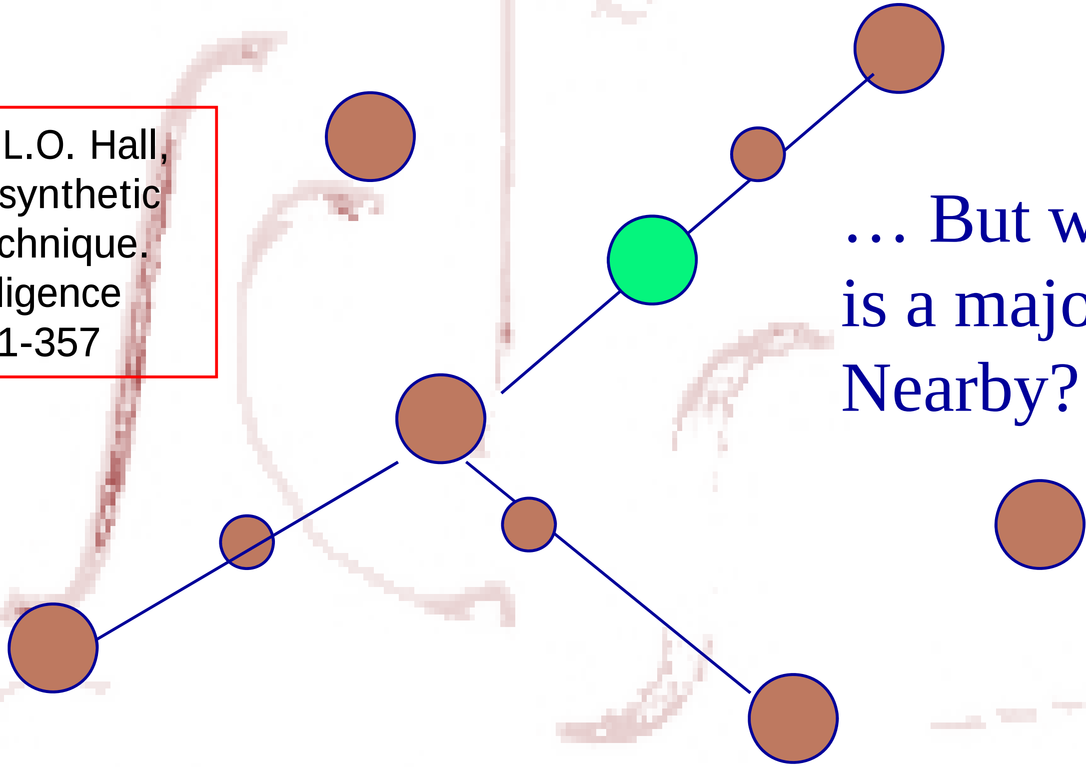
Smote: Synthetic Minority Over-sampling Technique

- **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  f2_1 = 4  f2_1 - f1_1 = -2** |
| **f1_2 = 4  f2_2 = 3  f2_2 - f1_2 = -1** |
| **The new samples will be generated as** |
| **(f1',f2') = (6,4) + rand(0-1) * (-2,-1)** |
| **rand(0-1) generates a random number between 0 and 1.** |

# State-of-the-art algorithm: SMOTE.

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



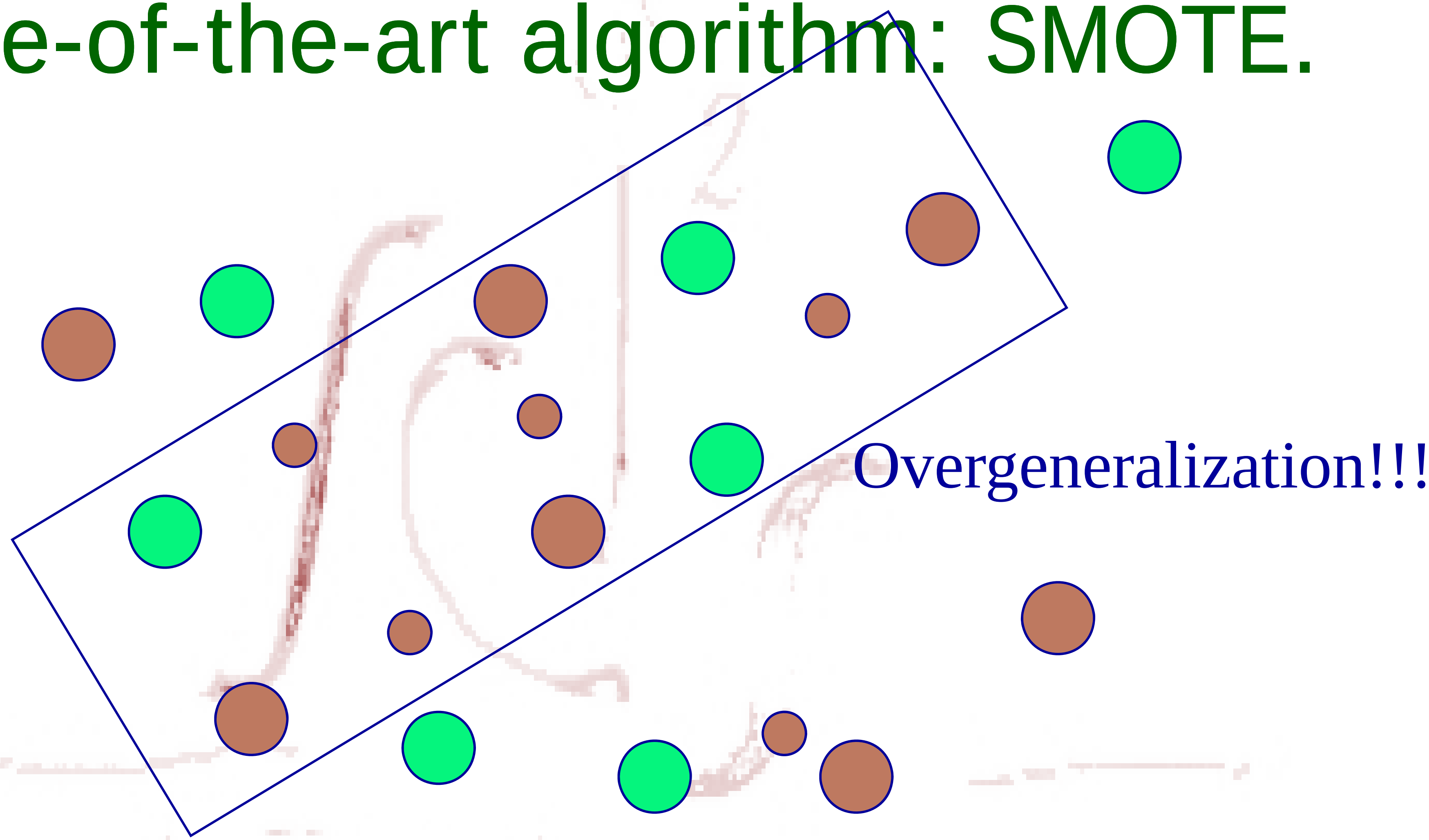
… But what if there is a majority sample Nearby?

: Minority sample

: Synthetic sample

: Majority sample

# State-of-the-art algorithm: SMOTE.




: Minority sample

: Majority sample

: Synthetic sample

# Smote + Tomek links

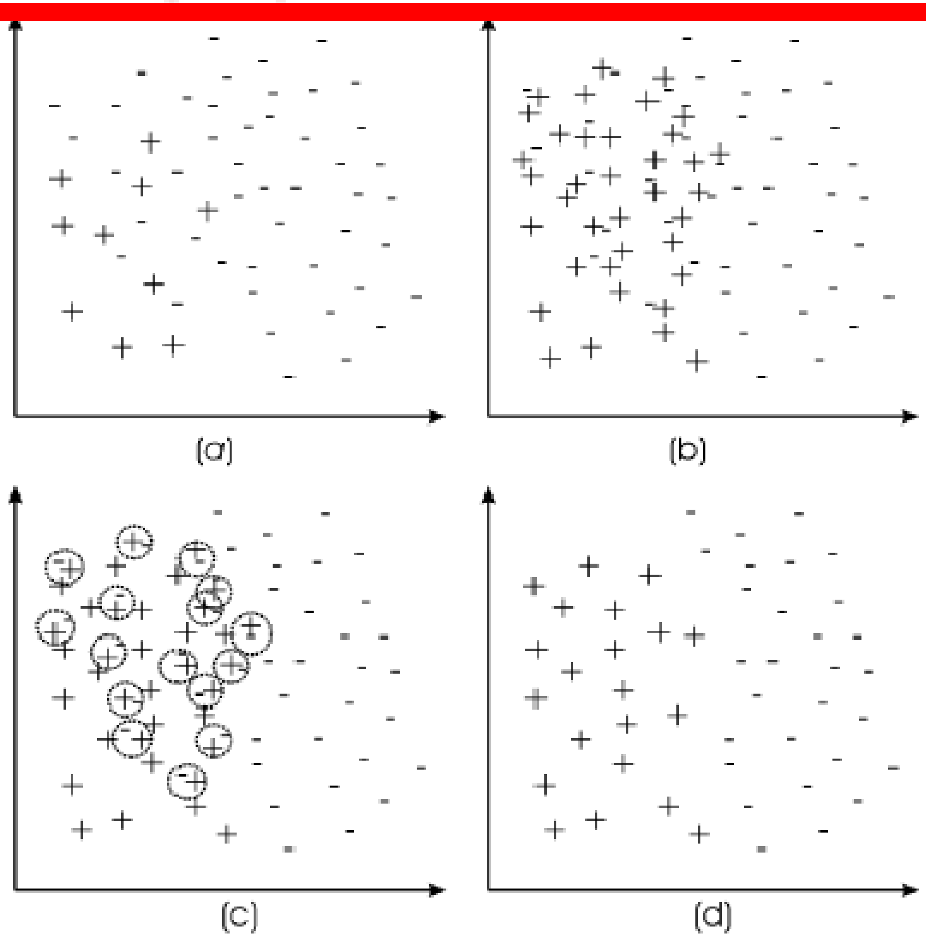
- **Problem with Smote: might introduce the artificial minority class examples too deeply in the majority class space.**
- **Tomek links: data cleaning**
- **Instead of removing only the majority class examples that form Tomek links, examples from both classes are removed**

# State-of-the-art algorithm: SMOTE.

SMOTE

+

TomekLinks



# State-of-the-art algorithm: SMOTE.

**SMOTE + ENN:**

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

# State-of-the-art algorithm: SMOTE.

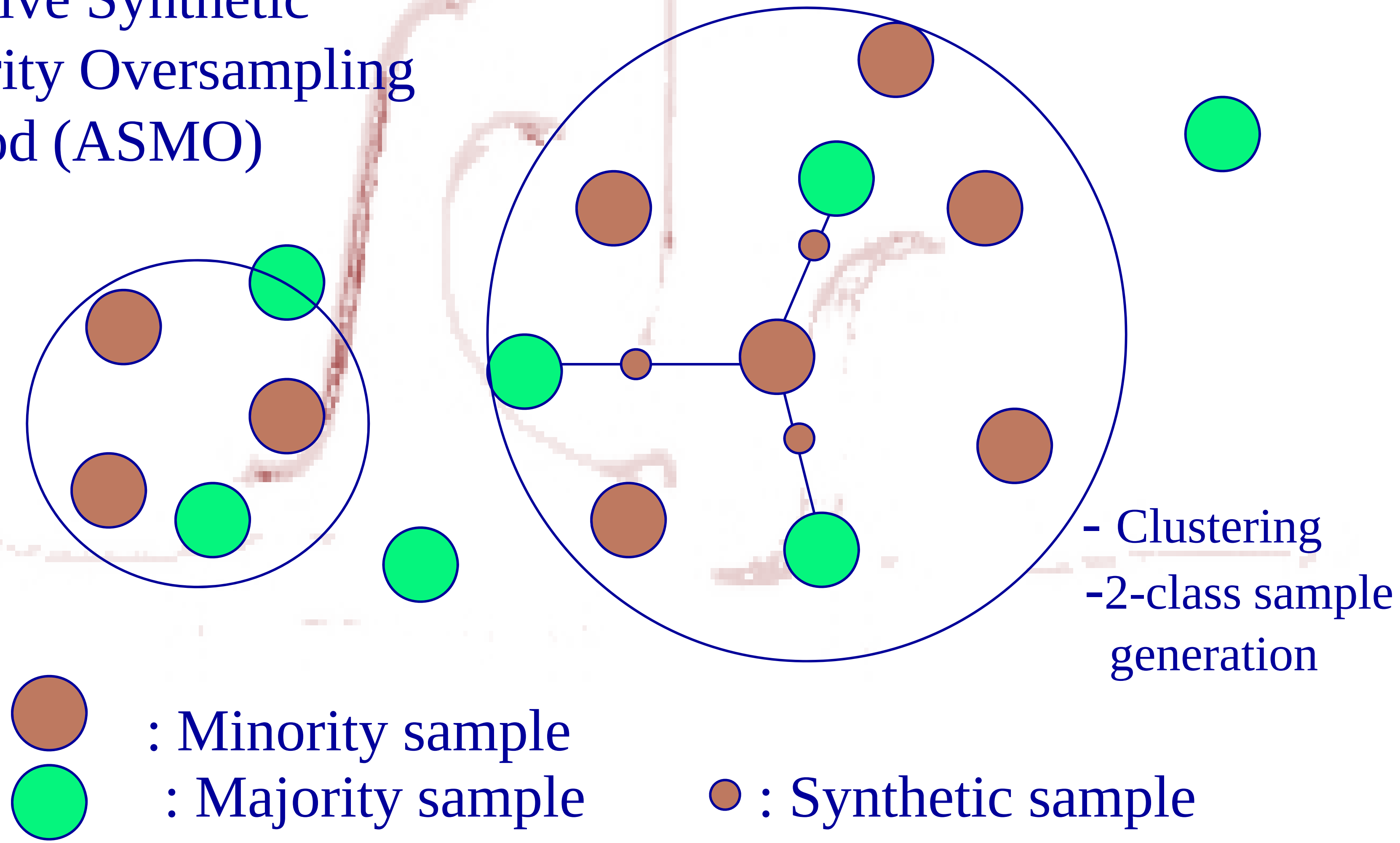
Table 6: Performance ranking for original and balanced data sets for pruned decision trees.

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

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

# State-of-the-art algorithm: SMOTE.

Adaptive Synthetic
Minority Oversampling
Method (ASMO)



# State-of-the-art algorithm: SMOTE.

Borderline-SMOTE: Genera ejemplos sintéticos entre ejemplos minoritarios y cercanos a los bordes.

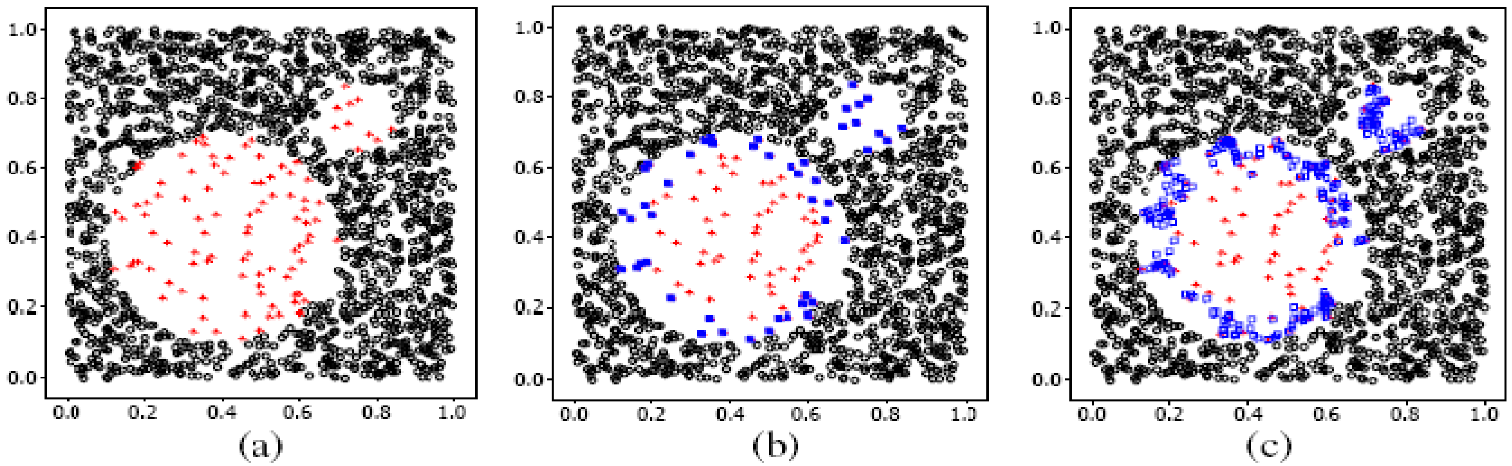



**Fig. 1.** (a) The original distribution of Circle data set. (b) The borderline minority examples (*solid squares*). (c) The borderline synthetic minority examples (*hollow squares*).

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.

# Outline

- ✓ Introduction to Imbalanced Data Sets
- ✓ Some results on the use of evolutionary prototype selection for imbalanced data sets
- ✓ Class imbalance related topics:
  Cost-Sensitive Learning and anomaly detection
- ✓ Concluding Remarks

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

**Evolutionary Under-Sampling**

**Experimental Framework and Results**

**Conclusions and Future Work**

**Source:** García S, Herrera F (2008) Evolutionary Under-Sampling for Classification with Imbalanced Data Sets: Proposals and Taxonomy. Evolutionary Computation. In press.

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

**Evolutionary Under-Sampling**

**Experimental Framework and Results**

**Conclusions and Future Work**

# Evolutionary Under-Sampling

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

**Representation:**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

Selected pattern for classifying With 1-NN

Eliminated pattern

**Evolutionary algorithms are good global search methods**

# Evolutionary Under-Sampling

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

## Previous results:


a. 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*

b. 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*

c. 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*

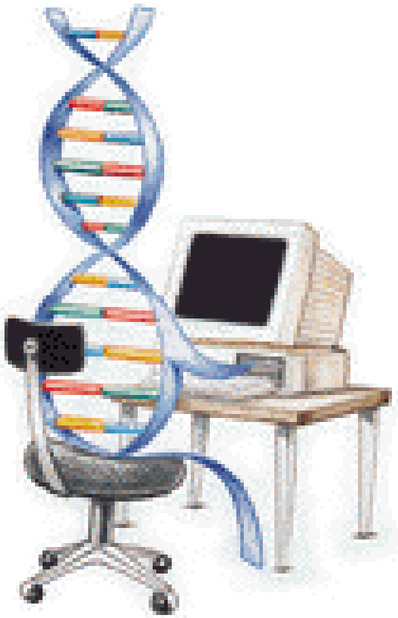
d. J.R. Cano, F. Herrera, M. Lozano, Evolutionary Stratified Training Set Selection for Extracting Classification Rules with Trade-off Precision-Interpretability. *Data and Knowledge Engineering 60 (2007) 90-108,* *doi:10.1016/j.datak.2006.01.008*

e. 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*

f. J.R. Cano, F. Herrera, M. Lozano, S. García, Making CN2-SD Subgroup Discovery Algorithm scalable to Large Size Data Sets using Instance Selection. *Expert Systems with Applications,* *doi:10.1016/j.eswa.2007.08.083*, *in press (2008)*

# What is a genetic algorithm?

## Genetic algorithms



**They are optimization algorithms,**

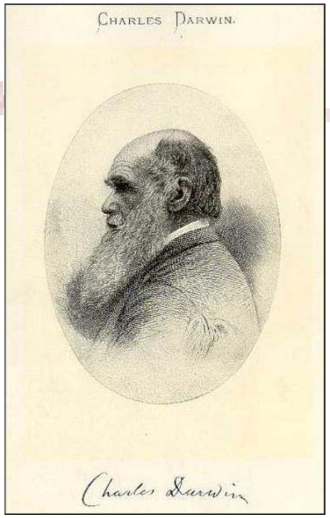
**search**

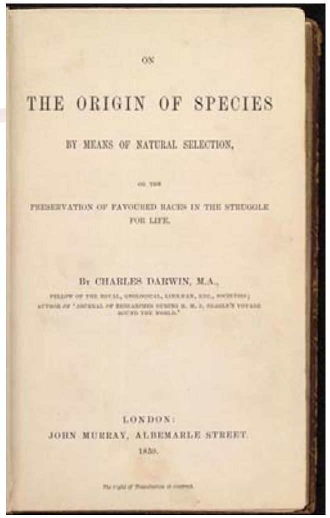
**and learning**

**inspired in the process of**

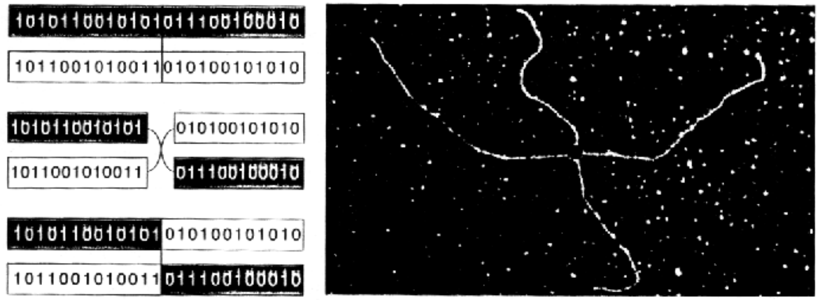
**Natural and**

**Genetic Evolution**



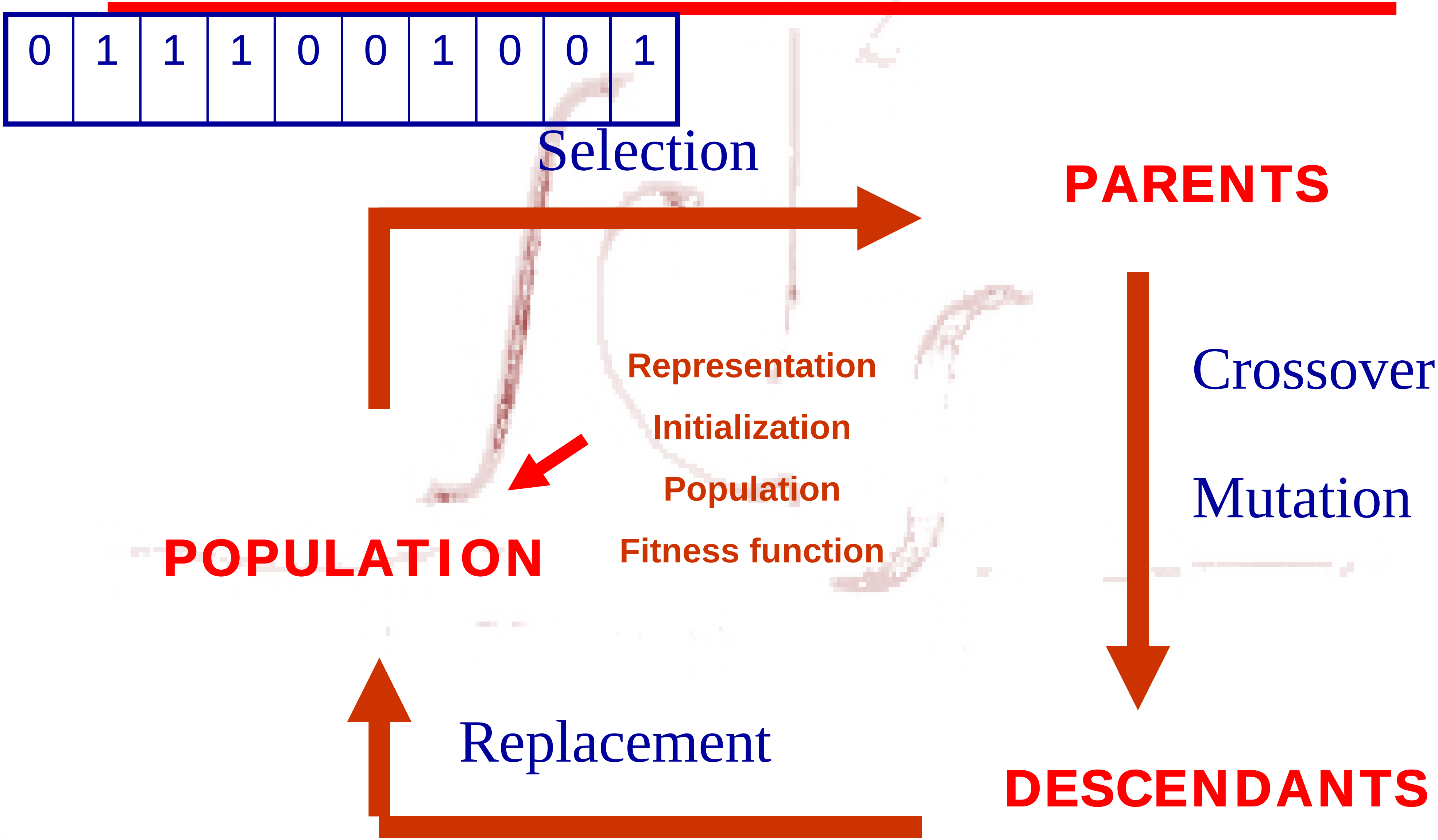



# Genetic Algorithms



CROSSOVER is the fundamental mechanism of genetic rearrangement for both real organisms and genetic algorithms.

Chromosomes line up and then swap the portions of their genetic code beyond the crossover point.



# Evolutionary Under-Sampling

**Evolutionary algorithm for re-sampling:**

**Representation:**

| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|

**Base Method: CHC**

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

**Models:**

- **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.

# Evolutionary Under-Sampling

## Type of Selection:

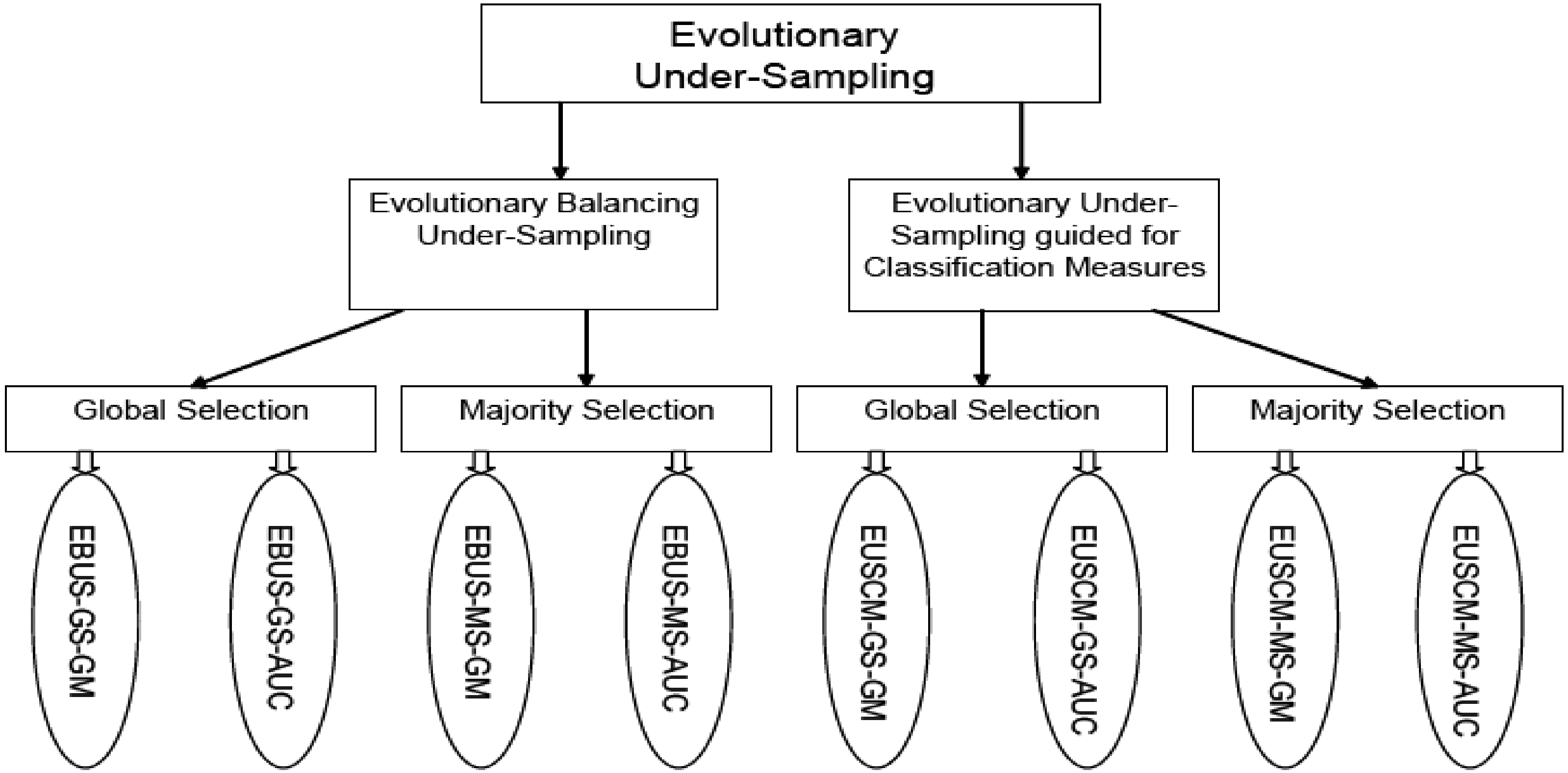
- GS: Global Selection, the selection scheme proceeds over any kind of instance.
- MS: Majority Selection, the selection scheme only proceeds over majority class instances.

## Evaluation Measures:

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

# Evolutionary Under-Sampling

## Taxonomy:



# Evolutionary Under-Sampling

## Fitness function in EBUS model:

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

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

***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:

$$Fitness(S) = g,$$

$$Fitness(S) = AUC,$$

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

**Evolutionary Under-Sampling**

**Experimental Framework and Results**

**Conclusions and Future Work**

# Experimental Framework and Results

## Algorithms used in the comparison:

**Prototype Selection:**

**IB3 DROP3 EPS-CHC EPS-IGA**

**Undersampling:**

**Random Under-Samplig TomekLinks (TL)**

**CNN OSS CNN+TL NCL**

**CPM SBC**

Under-Sampling based on clustering

# Experimental Framework and Results

## Data sets:

IR:

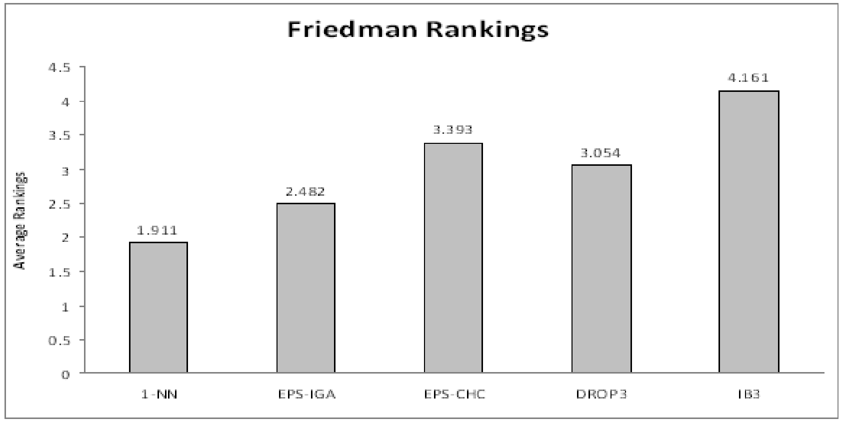
Imbalance ratio:

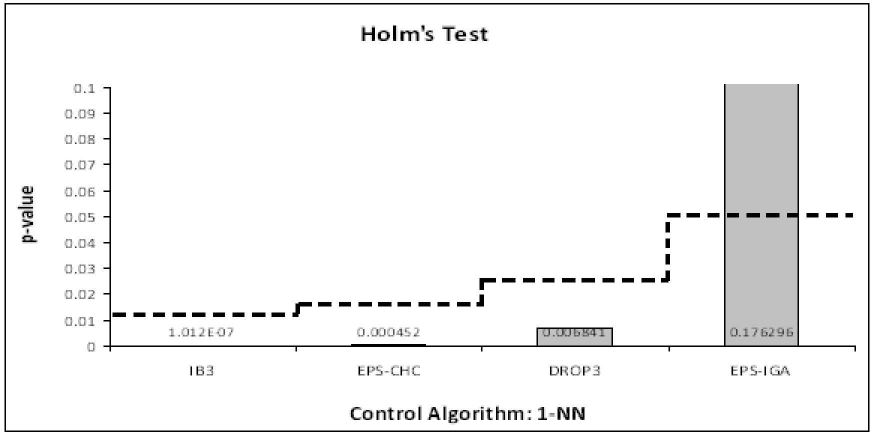
Number negative examples / Number positive examples

| 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 |

# Experimental Framework and Results

## **Part I:** Classical prototype selection as imbalanced undersampling

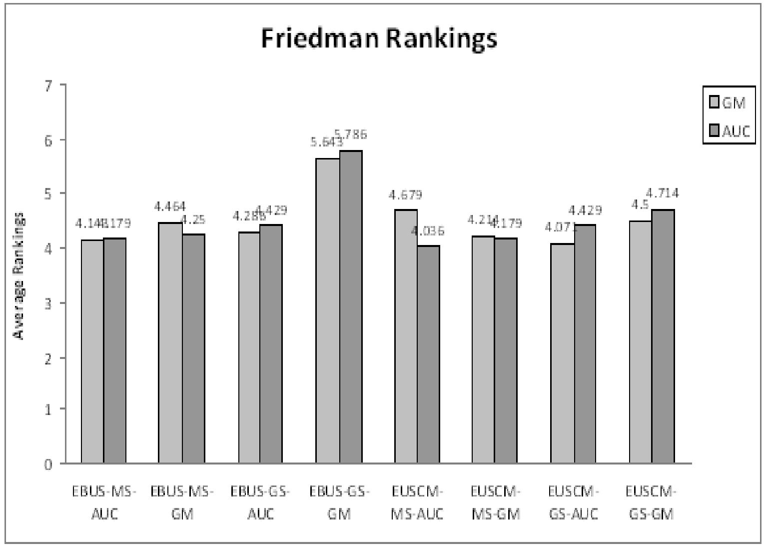


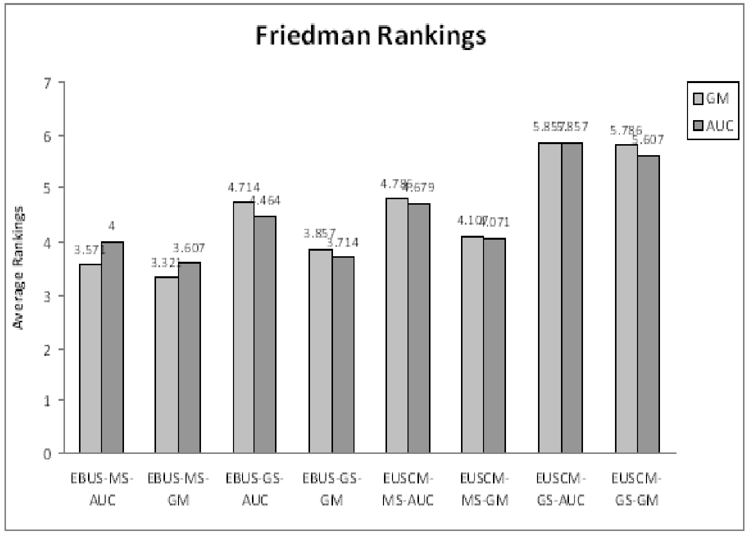

**Classical prototype selection is not recommendable for tackling imbalanced data sets. 1-NN without preprocessing behaves the best.**

# Experimental Framework and Results

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




IR < 9




IR > 9

# Experimental Framework and Results

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

**IR < 9:**

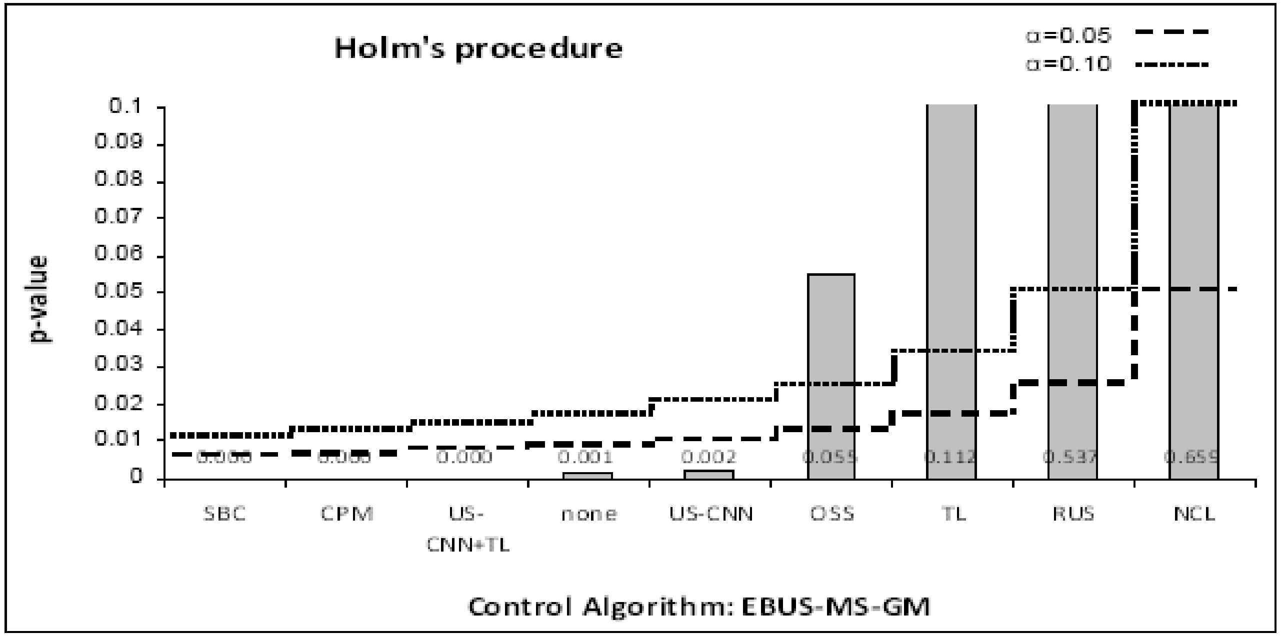
- **EUSCM behaves better than EBUS (P factor has little interest)**
- **Little differences between GM and AUC.**

**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**

# Experimental Framework and Results

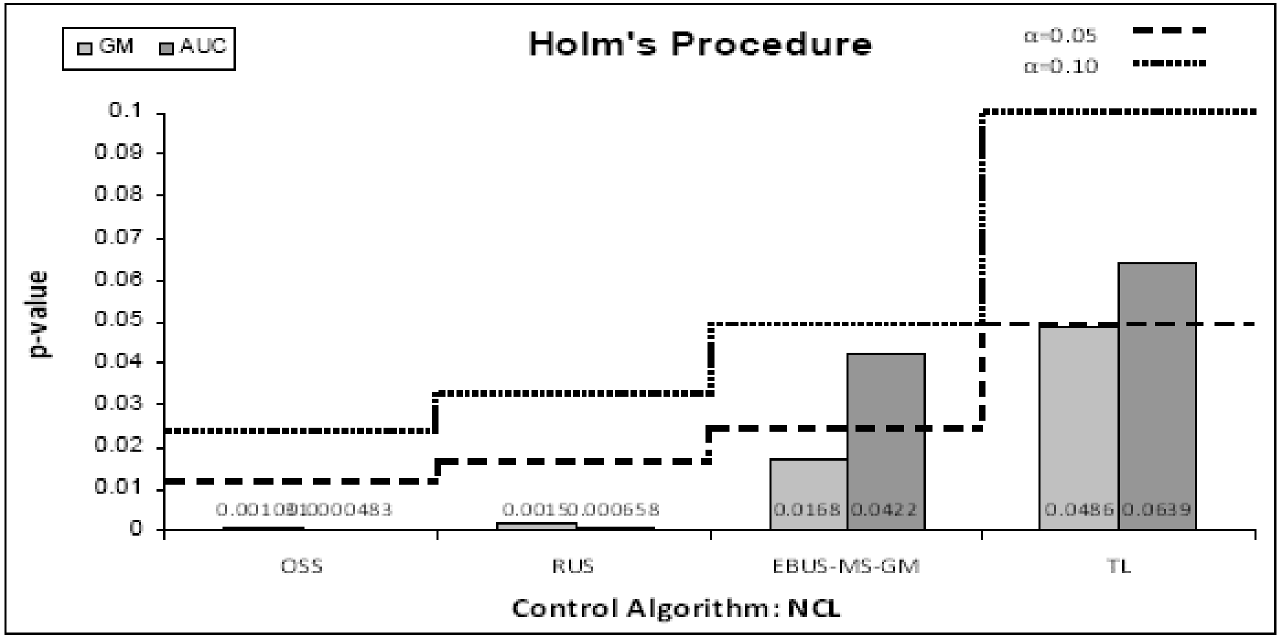
**Part III:** **Comparison with other under-sampling approaches**




**Considering all data sets**

# Experimental Framework and Results

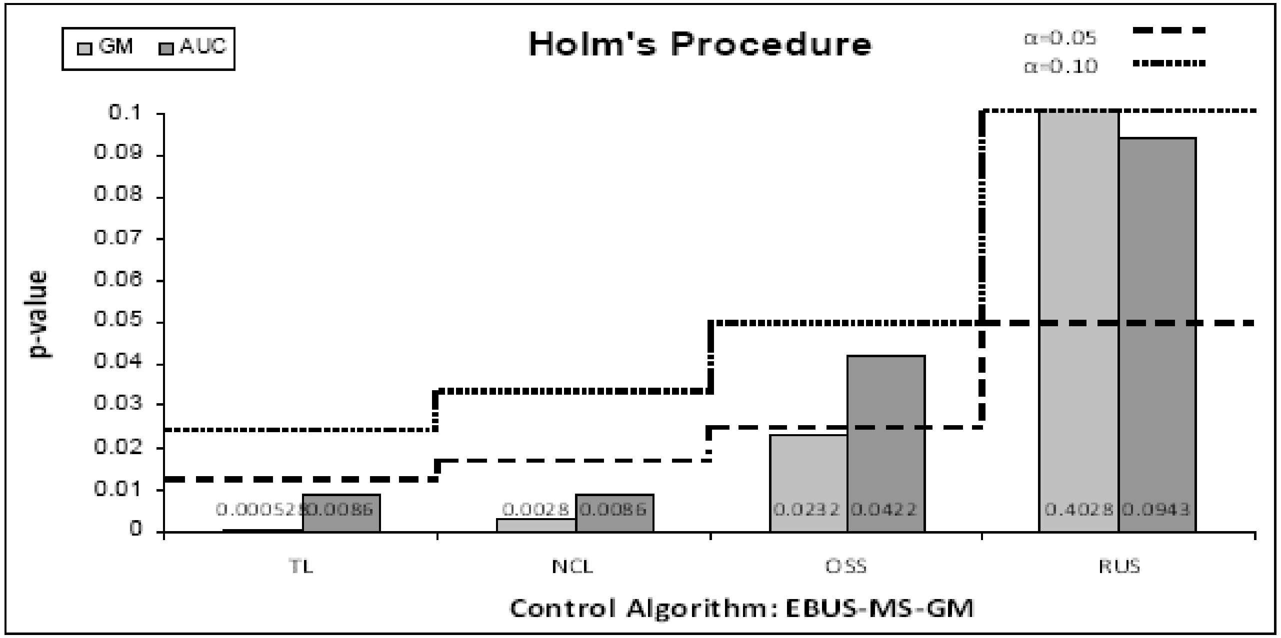
## **Part III:** Comparison with other under-sampling approaches




**Considering data sets with IR < 9**

# Experimental Framework and Results

## **Part III:** Comparison with other under-sampling approaches




**Considering data sets with IR > 9**

# 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

**Evolutionary Under-Sampling**

**Experimental Framework and Results**

**Conclusions and Future Work**

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

## Conclusions and Future Work

- **Prototype Selections 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

**FUTURE WORK**

- **Use of evolutionary under-sampling in training set selection, in order to optimize the performance of other classification algorithms.**
- **Study the scalability of these models in very large data sets.**
- **Hybridize evolutionary under-sampling with SMOTE or other over-sampling approaches.**

# Outline

- ✓ Introduction to Imbalanced Data Sets
- ✓ Some results on the use of evolutionary prototype selection for imbalanced data sets
- ✓ Class imbalance related topics:
  Cost-Sensitive Learning and anomaly detection
- ✓ Concluding Remarks

# Class Imbalance related topics

## Class Imbalance vs. Asymmetric Misclassification costs

- Class Imbalance: one class occurs much more often than the other
- Asymmetric misclassification costs: the cost of misclassifying an example from one class is much larger than the cost of misclassifying an example from the other class.

- **One way to correct for imbalance: train a cost sensitive classifier with the misclassification cost of the minority class greater than that of the majority class.**

- **One way to make an algorithm cost sensitive: intentionally imbalance the training set.**

# Class Imbalance related topics

## Cost-sensitive

- **Traditionally assumed a cost matrix of the form:**

| | True = 0 | True = 1 |
|---|---|---|
| Predict = 0 | $C(0,0)$ | $C(0,1)$ |
| Predict = 1 | $C(1,0)$ | $C(1,1)$ |

- **cost that depends on particular example $x$**

| | True = 0 | True = 1 |
|---|---|---|
| Predict = 0 | $C(0,0,x)$ | $C(0,1,x)$ |
| Predict = 1 | $C(1,0,x)$ | $C(1,1,x)$ |

# Class Imbalance related topics

## Making Classifiers Cost-sensitive

- **A solution would be to have a procedure that converted a broad variety of classifiers into cost-sensitive ones**
  - **Stratification: change the frequency of classes in the training data in proportion to their cost**
    - **distort the distribution of examples**
    - **If it is done by under-sampling, it reduces the data available for learning.**
    - **If it is done by over-sampling, it increase learning time**
  - **Cost modifying**

# Class Imbalance related topics

Weighting versus Sampling

**Two weighting**

- **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**

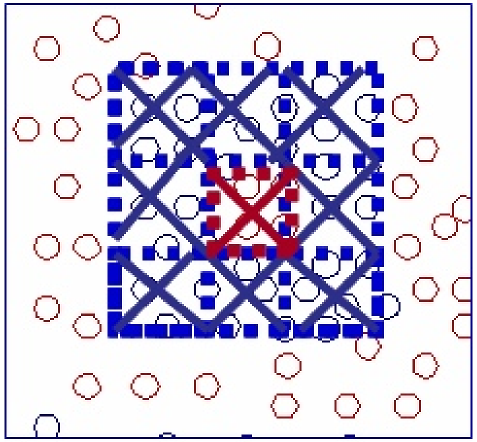
# Class Imbalance related topics

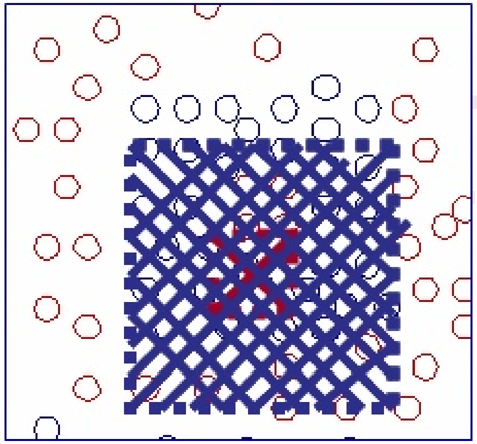
## Anomaly detection/outlier detection

- The problem of detecting anomalies (irregularities that cannot be explained by simple domain models and knowledge) in data.
- Much of the existing work focuses on detecting outliers solely for the purpose of removing them from the analysis to prevent them from unduly affecting the data mining process instead of treating them as interesting phenomena in their own right.
- Outlier detection and anomaly detection can be managed as classification of imbalanced data sets.

# Learning in non-balanced domains

## Anomaly detection/outlier detection/rare cases/small disjunts





**Facet-wise analysis of the problems**

- Conditions to obtain classifiers that represent starved niches
- Take-over time of starved niches

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

CONCLUSIONS: Methods that deal with class imbalancesand small disjuncts simultaneously, cluster-based oversampling, is shown to outperform all the class imbalance geared methods used in the study.

# Outline

- ✓ Introduction to Imbalanced Data Sets
- ✓ Some results on the use of evolutionary prototype selection for imbalanced data sets
- ✓ Class imbalance related topics:
  Cost-Sensitive Learning and anomaly detection
- ✓ Concluding Remarks

# Classification with Imbalanced data sets

## Final Comments

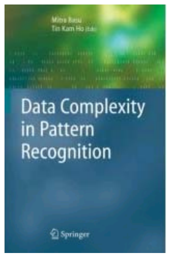
**Other studies with imbalanced data sets in the research group SCI$^2$S.**

- **Analysis of the use of fuzzy rule based classification systems (FRBCSs) for imbalanced data sets.**

**Fernandez, S. García, M.J. del Jesus, F. Herrera, A Study of the Behaviour of Linguistic Fuzzy Rule Based Classification Systems in the Framework of Imbalanced Data Sets. *Fuzzy Sets and Systems (2008)*. *doi: 10.1016/j.fss.2007.12.023***

- ***To develop new learning algorithms for FRBCSs for imbalanced data sets.***

- **To analyze the data in terms of data complexity in order to guide EUS to a better selection of instances and obtain generalized subsets.**



# Classification with Imbalanced data sets

## Final Comments

**Resampling is a good approach for managing imbalanced data sets and it is under evolution:**

**The following is an interesting paper analysing the balance for resampling.**

**Chawla NV, Cieslak DA, Hall LO, Joshi A (2008) Automatically countering imbalance and its empirical relationship to cost. Data Mining and Knowledge Discovery, in press.**

**Cost-proportionate weighted sampling allow us to solve cost-sensitive learning, and hence learning from imbalanced dataset. It is necessary to manage algorithms for learning with weights. See the recent contibution**

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

**Imbalanced data sets and related areas (cost-sensitive learning, anomaly detection, outlier detection) are important topics from the practical point of view in Data Mining, and they are important problems in Data Mining for the next years.**

# Classification with Imbalanced data sets

## Final Comments

**A list of bibliography in the topic can be found in the link:**

**http://sci2s.ugr.es/keel/specific.php?area=43**

**The following recent publications are two examples of the application in the field of medicine, an important area where we find imbalanced data sets.**


- **B. Lerner, J. Yeshaya, L. Koushnir. On the classification of a small imbalanced cytogenetic image database. IEEE/ACM Transactions on Computational Biology and Bioinformatics 4:2 (2007) 204-215**
- **Maciej A. Mazurowski, Piotr A. Habas, Jacek M. Zurada, Joseph Y. Lo, Jay A. Baker, Georgia D. Tourassi. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural Networks Volume 21 , Issue 2-3 (March, 2008) Pages 427-436**

# Summary

1. Introduction to Data Mining and Knowledge Discovery
2. Data Preparation
3. Introduction to Prediction, Classification, Clustering and Association
4. Data Mining - From the Top 10 Algorithms to the New Challenges
5. Introduction to Soft Computing. Focusing our attention in Fuzzy Logic and Evolutionary Computation
6. Soft Computing Techniques in Data Mining: Fuzzy Data Mining and Knowledge Extraction based on Evolutionary Learning
7. Genetic Fuzzy Systems: State of the Art and New Trends
8. Some Advanced Topics I: Classification with Imbalanced Data Sets
9. Some Advanced Topics II: Subgroup Discovery
10. Some advanced Topics III: Data Complexity
11. Final talk: How must I Do my Experimental Study? Design of Experiments in Data Mining/Computational Intelligence. Using Non-parametric Tests. Some Cases of Study.