INTELGECIA DE NEGOCIO
2020 - 2021

- Tema 1. Introducción a la Inteligencia de Negocio
- Tema 2. Minería de Datos. Ciencia de Datos
- Tema 3. Modelos de Predicción: Clasificación, regresión y series temporales
- Tema 4. Preparación de Datos
- Tema 5. Modelos de Agrupamiento o Segmentación
- Tema 6. Modelos de Asociación
- **Tema 7. Modelos Avanzados de Minería de Datos**
- Tema 8. Big Data
Modelos avanzados de Minería de Datos

Objetivos:

- Analizar diferentes problemas y técnicas avanzadas de ciencia de datos, tanto extensiones del problema clasificación clásico con nuevos problemas: anomalías, análisis de sentimientos, clasificación no balanceada, deep learning, ....
Inteligencia de Negocio

TEMA 7. Modelos Avanzados de Minería de Datos

1. Clases no balanceadas/equilibradas
2. Características intrínsecas de los datos en clasificación
3. Detección de anomalías
4. Problemas no estándar de clasificación
5. Análisis de Sentimientos
6. Deep Learning
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 of Melanomas, Intrusion detection, Insurance risk modeling, Hardware fault detection.
Such a situation introduce challenges for typical classifiers (such as decision tree) “systems that are designed to optimize overall accuracy without taking into account the relative distribution of each class”.

As a result, these classifiers tend to ignore small classes while concentrating on classifying the large ones accurately.
Why learning from imbalanced data-sets might be difficult?

1. Search process guided by global error rates.
2. Classification rules over the positive class are highly specialized.
3. Classifiers tend to ignore small classes concentrating on classifying large ones accurately.
Why learning from imbalanced data-sets might be difficult?

- Skewed class distribution:
  - Measured by the fraction between majority and minority samples
  - Imbalance ratio (IR)

- **Intrinsic Data Characteristics**
  - Not only imbalance hinders classification performance
  - IR ≈ 9

Why learning from imbalanced data-sets might be difficult?

Contents

I. Introduction to imbalanced data sets

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

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

IV. Class imbalance: Trends and final comments
I. Introduction to imbalanced data sets

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

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

IV. Class imbalance: Trends and final comments
Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

Resampling the original training set

Cost Modifying: Cost-sensitive learning

Ensembles to address class imbalance
Introduction to Imbalanced Data Sets

Some recent applications

- Significance of the topic in recent applications

- Tan, Shing Chiang; Watada, Junzo; Ibrahim, Zuwairie; et ál.; Evolutionary Fuzzy ARTMAP Neural Networks for Classification of Semiconductor Defects. IEEE Transactions on Neural Networks and Learning Systems 26 (5): 933-950 (MAY 2015)

- Danenas, Paulius; Garsva, Gintautas; Selection of Support Vector Machines based classifiers for credit risk domain Experty Systems with Applications 42 (6) : 3194-3204  (APR 2015)

- Liu, Nan; Koh, Zhi Xiong; Chua, Eric Chern-Pin; et ál.; Risk Scoring for Prediction of Acute Cardiac Complications from Imbalanced Clinical Data. IEEE Journal of Biomedical and Health Informatics 18 (6) : 1894-1902  (NOV 2014)
Introduction to Imbalanced Data Sets

Some recent applications

- Significance of the topic in recent applications

- Radtke, Paulo V. W.; Granger, Eric; Sabourin, Robert; et ál.; Skew-sensitive boolean combination for adaptive ensembles - An application to face recognition in video surveillance Information Fusion 20: 31-48 (NOV 2014)

- Yu, Hualong; Ni, Jun; An Improved Ensemble Learning Method for Classifying High-Dimensional and Imbalanced Biomedicine Data IEEE-ACM Transactions on Computational Biology and Bioinformatics 11(4) : 657-666 (AUG 2014)


Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets
Resampling the original training set
Cost Modifying: Cost-sensitive learning
Ensembles to address class imbalance
Introduction to Imbalanced Data Sets

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

We need to change the way to evaluate a model performance!
How can we evaluate an algorithm in imbalanced domains?

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

<table>
<thead>
<tr>
<th></th>
<th>Positive Prediction</th>
<th>Negative Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Positive Class</strong></td>
<td>True Positive (TP)</td>
<td>False Negative (FN)</td>
</tr>
<tr>
<td><strong>Negative Class</strong></td>
<td>False Positive (FP)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

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

**Classical evaluation:**

Error Rate: \((FP + FN) / N\)

Accuracy Rate: \((TP + TN) / N\)
Introduction to Imbalanced Data Sets

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

- **Positive true ratio:** $a^+ = \frac{TP}{TP+FN}$
- **Negative true ratio:** $a^- = \frac{TN}{FP+TN}$

**Evaluation function:** True ratio

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

- **Precision:** $\frac{TP}{TP+FP}$
- **Recall:** $\frac{TP}{TP+FN}$
- **F-measure:** $(2 \times \text{precision} \times \text{recall}) / (\text{recall} + \text{precision})$

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

AUC = \frac{1 + \text{TP rate}}{2} - \frac{\text{FP rate}}{2}

Espacio ROC

<table>
<thead>
<tr>
<th></th>
<th>PP</th>
<th>NP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>0,8</td>
<td>0,121</td>
</tr>
<tr>
<td>NC</td>
<td>0,2</td>
<td>0,879</td>
</tr>
</tbody>
</table>

Introduction to Imbalanced Data Sets
Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

Resampling the original training set

Cost Modifying: Cost-sensitive learning

Ensembles to address class imbalance
Introduction to Imbalanced Data Sets

**Data level vs Algorithm Level**

**Strategies to deal with imbalanced data sets**

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

**Over-Sampling**
- Random
- Focused

**Under-Sampling**
- Random
- Focused

**Cost Modifying (cost-sensitive)**

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

**Boosting approaches:** ensemble learning, AdaBoost, …
Some recent applications

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

**Resampling the original training set**

**Cost Modifying: Cost-sensitive learning**

**Ensembles to address class imbalance**
Resampling the original data sets

**Undersampling vs oversampling**

- **Under-sampling**
  - # examples –
  - # examples +

- **Over-sampling**
  - # examples –
  - # examples +
Resampling the original data sets

Oversampling: Replicating examples

SMOTE: Instead of replicating, let us invent some new instances.
Resampling the original data sets

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


... But what if there is a majority sample Nearby?

: Minority sample
: Majority sample
: Synthetic sample
Oversampling method: SMOTE

Example of a run

Resampling the original data sets
Resampling the original data sets

SMOTE hybridization: SMOTE + Tomek links

Figure: SMOTE+TomekLink

Figure 17: (a) Original data-set distribution. (b) Post-SMOTE dataset. (c) The identified Tomek Links. (d) The data-set after removing Tomek links
ENN removes any example whose class label differs from the class of at least two of their neighbors.

ENN removes more examples than the Tomek links do.

ENN removes examples from both classes.

SMOTE hybridization: SMOTE + ENN
## Resampling the original data sets

### SMOTE and hybridization: Analysis

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

<table>
<thead>
<tr>
<th>Data set</th>
<th>1 st</th>
<th>2 nd</th>
<th>3 rd</th>
<th>4 th</th>
<th>5 th</th>
<th>6 th</th>
<th>7 th</th>
<th>8 th</th>
<th>9 th</th>
<th>10 th</th>
<th>11 th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pima</td>
<td>Smt</td>
<td>RdOvr</td>
<td>Smt+Tmk</td>
<td>Smt+ENN</td>
<td>Tmk</td>
<td>NCL</td>
<td>Original</td>
<td>RdUdr</td>
<td>CNN+Tmk</td>
<td>CNN*</td>
<td>OSS*</td>
</tr>
<tr>
<td>German</td>
<td>RdOvr</td>
<td>Smt+Tmk</td>
<td>Smt+ENN</td>
<td>Smt</td>
<td>RdUdr</td>
<td>CNN</td>
<td>CNN+Tmk*</td>
<td>OSS*</td>
<td>Original*</td>
<td>Tmk*</td>
<td>NCL*</td>
</tr>
<tr>
<td>Post-operative</td>
<td>RdOvr</td>
<td>Smt+ENN</td>
<td>Smt</td>
<td>Original</td>
<td>CNN</td>
<td>RdUdr</td>
<td>CNN+Tmk</td>
<td>OSS*</td>
<td>Tmk*</td>
<td>NCL*</td>
<td>Smt+Tmk*</td>
</tr>
<tr>
<td>Haberman</td>
<td>Smt+ENN</td>
<td>Smt+Tmk</td>
<td>RdOvr</td>
<td>Smt</td>
<td>CNN</td>
<td>NCL</td>
<td>Smt+Tmk</td>
<td>CNN*</td>
<td>Original*</td>
<td>CNN+Tmk*</td>
<td></td>
</tr>
<tr>
<td>Splice-ie</td>
<td>RdOvr</td>
<td>Original</td>
<td>Tmk</td>
<td>Smt</td>
<td>CNN</td>
<td>NCL</td>
<td>Smt+ENN</td>
<td>CNN+Tmk*</td>
<td>RdUdr*</td>
<td>OSS*</td>
<td></td>
</tr>
<tr>
<td>Splice-ei</td>
<td>Smt</td>
<td>Smt+Tmk</td>
<td>Smt+ENN</td>
<td>CNN+Tmk</td>
<td>OSS</td>
<td>RdOvr</td>
<td>Tmk</td>
<td>CNN</td>
<td>NCL</td>
<td>Original</td>
<td>RdUdr</td>
</tr>
<tr>
<td>Vehicle</td>
<td>RdOvr</td>
<td>Smt</td>
<td>Smt+Tmk</td>
<td>OSS</td>
<td>CNN</td>
<td>Original</td>
<td>CNN+Tmk</td>
<td>Tmk</td>
<td>NCL*</td>
<td>Smt+ENN*</td>
<td>RdUdr*</td>
</tr>
<tr>
<td>Letter-vowel</td>
<td>Smt+ENN</td>
<td>Smt+Tmk</td>
<td>RdOvr</td>
<td>Tmk</td>
<td>NCL*</td>
<td>Original*</td>
<td>CNN*</td>
<td>CNN+Tmk*</td>
<td>RdUdr*</td>
<td>OSS*</td>
<td></td>
</tr>
<tr>
<td>New-thyroid</td>
<td>Smt+ENN</td>
<td>Smt+Tmk</td>
<td>RdOvr</td>
<td>RdUdr</td>
<td>CNN</td>
<td>Original</td>
<td>Tmk</td>
<td>CNN+Tmk</td>
<td>NCL</td>
<td>OSS</td>
<td></td>
</tr>
<tr>
<td>E.Coli</td>
<td>Smt+Tmk</td>
<td>Smt+ENN</td>
<td>RdOvr</td>
<td>NCL</td>
<td>Tmk</td>
<td>RdUdr</td>
<td>Original</td>
<td>OSS</td>
<td>CNN+Tmk*</td>
<td>CNN*</td>
<td></td>
</tr>
<tr>
<td>Satimage</td>
<td>Smt+ENN</td>
<td>Smt+Tmk</td>
<td>RdOvr</td>
<td>NCL</td>
<td>Tmk</td>
<td>Original*</td>
<td>OSS*</td>
<td>CNN+Tmk*</td>
<td>RdUdr*</td>
<td>CNN*</td>
<td></td>
</tr>
<tr>
<td>Flag</td>
<td>RdOvr</td>
<td>Smt+ENN</td>
<td>Smt+Tmk</td>
<td>CNN+TmkSmt</td>
<td>RdUdr</td>
<td>CNN*</td>
<td>OSS*</td>
<td>Tmk*</td>
<td>Original*</td>
<td>NCL*</td>
<td></td>
</tr>
<tr>
<td>Glass</td>
<td>Smt+ENN</td>
<td>RdOvr</td>
<td>NCL</td>
<td>Smt</td>
<td>Smt+Tmk</td>
<td>Original</td>
<td>Tmk</td>
<td>RdUdr</td>
<td>CNN+Tmk*</td>
<td>OSS*</td>
<td>CNN*</td>
</tr>
<tr>
<td>Letter-a</td>
<td>Smt+Tmk</td>
<td>Smt+ENN</td>
<td>RdOvr</td>
<td>OSS</td>
<td>Original</td>
<td>Tmk</td>
<td>CNN+Tmk</td>
<td>NCL</td>
<td>CNN</td>
<td>RdUdr*</td>
<td></td>
</tr>
<tr>
<td>Nursery</td>
<td>RdOvr</td>
<td>Tmk</td>
<td>Original</td>
<td>NCL</td>
<td>CNN*</td>
<td>OSS*</td>
<td>Smt+Tmk*</td>
<td>Smt*</td>
<td>CNN+Tmk*</td>
<td>Smt+ENN*</td>
<td>RdUdr*</td>
</tr>
</tbody>
</table>

Resampling the original data sets

Other SMOTE hybridizations


Resampling the original data sets

Final comments

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

How can we evaluate an algorithm in imbalanced domains?

Strategies to deal with imbalanced data sets

Resampling the original training set

Cost Modifying: Cost-sensitive learning

Ensembles to address class imbalance
Cost-sensitive learning

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

Over Sampling
- Random
- Focused

Under Sampling
- Random
- Focused

Cost Modifying

# examples of -

# examples of +

Similar sum of costs
Results and Statistical Analysis

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>( \text{AUC}_{\text{tr}} )</th>
<th>( \text{AUC}_{\text{tst}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>C45</td>
<td>0.8774 ± 0.0392</td>
<td>0.7902 ± 0.0804</td>
</tr>
<tr>
<td>C45 SMOTE</td>
<td>0.9606 ± 0.0142</td>
<td>0.8324 ± 0.0728</td>
</tr>
<tr>
<td>C45 SENN</td>
<td>0.9471 ± 0.0154</td>
<td><strong>0.8390 ± 0.0772</strong></td>
</tr>
<tr>
<td>C45CS</td>
<td>0.9679 ± 0.0103</td>
<td>0.8294 ± 0.0758</td>
</tr>
<tr>
<td>C45 Wr_SMOTE</td>
<td>0.9679 ± 0.0103</td>
<td>0.8296 ± 0.0763</td>
</tr>
<tr>
<td>C45 Wr_US</td>
<td>0.9635 ± 0.0139</td>
<td>0.8245 ± 0.0760</td>
</tr>
<tr>
<td>C45 Wr_SENN</td>
<td>0.9083 ± 0.0377</td>
<td>0.8145 ± 0.0712</td>
</tr>
</tbody>
</table>

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 \((\alpha = 0.05)\)

<table>
<thead>
<tr>
<th>C4.5</th>
<th>none</th>
<th>SMOTE</th>
<th>SENN</th>
<th>CS</th>
<th>Wr_SMOTE</th>
<th>Wr_US</th>
<th>Wr_SENN</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>(\times)</td>
<td>((6.404E-05))</td>
<td>((1.058E-06))</td>
<td>((6.404E-06))</td>
<td>((7.904E-06))</td>
<td>(0.00341)</td>
<td>(0.37846)</td>
</tr>
<tr>
<td>SMOTE</td>
<td>(+6.404E-05)</td>
<td>(\times)</td>
<td>((1.0))</td>
<td>((1.0))</td>
<td>((1.0))</td>
<td>((1.0))</td>
<td>(+0.04903)</td>
</tr>
<tr>
<td>SENN</td>
<td>(+4.058E-06)</td>
<td>((1.0))</td>
<td>(\times)</td>
<td>((1.0))</td>
<td>((1.0))</td>
<td>((0.22569))</td>
<td>(+0.00152)</td>
</tr>
<tr>
<td>CS</td>
<td>(+6.404E-05)</td>
<td>((1.0))</td>
<td>((1.0))</td>
<td>(\times)</td>
<td>((1.0))</td>
<td>((1.0))</td>
<td>(+0.04903)</td>
</tr>
<tr>
<td>Wr_SMOTE</td>
<td>(+7.904E-06)</td>
<td>((1.0))</td>
<td>((1.0))</td>
<td>(\times)</td>
<td>((1.0))</td>
<td>((1.0))</td>
<td>(+0.04903)</td>
</tr>
<tr>
<td>Wr_US</td>
<td>(+0.00341)</td>
<td>((1.0))</td>
<td>((0.22569))</td>
<td>((1.0))</td>
<td>((1.0))</td>
<td>(\times)</td>
<td>((1.0))</td>
</tr>
<tr>
<td>Wr_SENN</td>
<td>(+0.37846)</td>
<td>(-0.04903)</td>
<td>(-0.00152)</td>
<td>(-0.04903)</td>
<td>(-0.04903)</td>
<td>((1.0))</td>
<td>(\times)</td>
</tr>
</tbody>
</table>
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 undersampling although all approaches are helpful, and directed oversampling is close to cost-adjusting”. Our study shows similar results.

Some recent applications
How can we evaluate an algorithm in imbalanced domains?
Strategies to deal with imbalanced data sets
Resampling the original training set
Cost Modifying: Cost-sensitive learning
Ensembles to address class imbalance
Ensemble-based classifiers try to improve the performance of single classifiers by inducing several classifiers and combining them to obtain a new classifier that outperforms every one of them. Hence, the basic idea is to construct several classifiers from the original data and then aggregate their predictions when unknown instances are presented. This idea follows human natural behavior which tend to seek several opinions before making any important decision.
Ensembles to address class imbalance

Fig. 3. Proposed taxonomy for ensembles to address the class imbalance problem.
TABLE XV
REPRESENTATIVE METHODS SELECTED FOR EACH FAMILY

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

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

TABLE XVI
HOLM TABLE FOR BEST INTERFAMILY ANALYSIS

<table>
<thead>
<tr>
<th>i</th>
<th>Algorithm (Rank)</th>
<th>Z</th>
<th>p-value</th>
<th>Holm</th>
<th>Hypothesis ($\alpha = 0.05$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>M14 (4.01)</td>
<td>5.7835</td>
<td>0.00000</td>
<td>0.01</td>
<td>Rejected for SBAG4</td>
</tr>
<tr>
<td>4</td>
<td>SMT (4.01)</td>
<td>3.90315</td>
<td>0.00009</td>
<td>0.0125</td>
<td>Rejected for SBAG4</td>
</tr>
<tr>
<td>3</td>
<td>C24 (3.58)</td>
<td>2.82052</td>
<td>0.00479</td>
<td>0.01667</td>
<td>Rejected for SBAG4</td>
</tr>
<tr>
<td>2</td>
<td>EASY (3.51)</td>
<td>2.64958</td>
<td>0.00806</td>
<td>0.025</td>
<td>Rejected for SBAG4</td>
</tr>
<tr>
<td>1</td>
<td>RUS1 (2.68)</td>
<td>0.56980</td>
<td>0.56881</td>
<td>0.05</td>
<td>Not Rejected</td>
</tr>
</tbody>
</table>

Control method : SBAG4, Rank 2.45.

TABLE XVIII
SHAFFER TESTS FOR INTERFAMILY COMPARISON

<table>
<thead>
<tr>
<th></th>
<th>SMT</th>
<th>M14</th>
<th>C24</th>
<th>RUS1</th>
<th>SBAG4</th>
<th>EASY</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMT</td>
<td>×</td>
<td>=0.02024</td>
<td>×</td>
<td>×</td>
<td>(0.00858)</td>
<td>(0.00095)</td>
</tr>
<tr>
<td>M14</td>
<td>=0.024024</td>
<td>×</td>
<td>=0.03047</td>
<td>=0.03047</td>
<td>=0.17082</td>
<td>(0.03356)</td>
</tr>
<tr>
<td>C24</td>
<td>=1.0</td>
<td>=0.03047</td>
<td>=1.0</td>
<td>=0.17082</td>
<td>(1.0)</td>
<td>(0.01725)</td>
</tr>
<tr>
<td>RUS1</td>
<td>+0.00858</td>
<td>(0.0)</td>
<td>×</td>
<td>(0.17082)</td>
<td>(1.0)</td>
<td>(0.22527)</td>
</tr>
<tr>
<td>SBAG4</td>
<td>+0.00995</td>
<td>(0.0)</td>
<td>+0.03356</td>
<td>(0.1)</td>
<td>×</td>
<td>(0.05641)</td>
</tr>
<tr>
<td>EASY</td>
<td>+0.01725</td>
<td>=1.0</td>
<td>=1.0</td>
<td>=0.22527</td>
<td>=0.05641</td>
<td>×</td>
</tr>
</tbody>
</table>

TABLE XVII
WILCOXON TESTS TO SHOW DIFFERENCES BETWEEN SBAG4 AND RUS1

<table>
<thead>
<tr>
<th>Comparison</th>
<th>$R^+$</th>
<th>$R^-$</th>
<th>Hypothesis ($\alpha = 0.05$)</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBAG4 vs. RUS1</td>
<td>527.5</td>
<td>462.5</td>
<td>Not Rejected</td>
<td>0.71717</td>
</tr>
</tbody>
</table>

$R^+$ are ranks for SBAG4 and $R^-$ for RUS1.
Our proposal: We develop a new ensemble construction algorithm (EUSBoost) based on RUSBoost, one of the simplest and most accurate ensemble, combining random undersampling with Boosting algorithm.

Our methodology aims to improve the existing proposals enhancing the performance of the base classifiers by the usage of the evolutionary undersampling approach. Besides, we promote diversity favoring the usage of different subsets of majority class instances to train each base classifier.

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

Ensemble-based algorithms are worthwhile, improving the results obtained by using data preprocessing techniques and training a single classifier.

The use of more classifiers make them more complex, but this growth is justified by the better results that can be assessed.

We have to remark the good performance of approaches such as RUSBoost or SmoteBagging, which despite of being simple approaches, achieve higher performance than many other more complex algorithms.

We have shown the positive synergy between sampling techniques (e.g., undersampling or SMOTE) and Bagging ensemble learning algorithm.
Contents

I. Introduction to imbalanced data sets

II. Why is difficult to learn in imbalanced domains?
   Intrinsic data characteristics
   Challenges on class distribution

I. Class imbalance: Data sets, implementations, …

II. Class imbalance: Trends and final comments
1. Clases no balanceadas/equilibradas
2. Características intrínsecas de los datos en clasificación
3. Detección de anomalías
4. Problemas no estándar de clasificación: MIL, MLL, ...
5. Análisis de Sentimientos
6. Deep Learning
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

Introduction to Imbalanced Data Sets
Why is difficult to learn in imbalanced domains?

Imbalance – why is it difficult?

An easier problem

More difficult one

Some of sources of difficulties:
• Overlapping,
• Small disjuncts,
• Lack of data,
• ...

Majority classes overlaps the minority class:
• Ambiguous boundary between classes
• Influence of noisy examples
• Difficult border, …
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

- Overlapping
- Small disjuncts/rare data sets
- Density: Lack of data
- Bordeline and Noise data
- Dataset shift

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.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

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

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

\[ F1: \text{maximum Fisher's discriminant ratio.} \]
\[ f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \]


Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

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

<table>
<thead>
<tr>
<th>Overlap Degree</th>
<th>$TP_{rate}$</th>
<th>$TN_{rate}$</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 %</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>20 %</td>
<td>.79.00</td>
<td>1.000</td>
<td>.8950</td>
</tr>
<tr>
<td>40 %</td>
<td>.4900</td>
<td>1.000</td>
<td>.7450</td>
</tr>
<tr>
<td>50 %</td>
<td>.4700</td>
<td>1.000</td>
<td>.7350</td>
</tr>
<tr>
<td>60 %</td>
<td>.4200</td>
<td>1.000</td>
<td>.7100</td>
</tr>
<tr>
<td>80 %</td>
<td>.2100</td>
<td>.9989</td>
<td>.6044</td>
</tr>
<tr>
<td>100 %</td>
<td>.0000</td>
<td>1.000</td>
<td>.5000</td>
</tr>
</tbody>
</table>
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**


**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.

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.
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
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.
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

---

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?**

**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

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.

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

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Small disjuncts/Rare or exceptional cases

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


CBO method: Cluster-based resampling identifies rare cases and resamples 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

Fig. 5 Example of small disjuncts on imbalanced data

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Original Data</th>
<th></th>
<th></th>
<th>Preprocessed Data with CBO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$TP_{rate}$</td>
<td>$TN_{rate}$</td>
<td>AUC</td>
<td>$TP_{rate}$</td>
</tr>
<tr>
<td>Artificial dataset</td>
<td>.0000</td>
<td>1.000</td>
<td>.5000</td>
<td>1.000</td>
</tr>
<tr>
<td>Subclus dataset</td>
<td>1.000</td>
<td>.9029</td>
<td>.9514</td>
<td>1.000</td>
</tr>
</tbody>
</table>
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 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

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

<table>
<thead>
<tr>
<th></th>
<th>Positive (‘1’)</th>
<th>Negative (‘0’)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:9</td>
<td>40</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>180</td>
</tr>
<tr>
<td>1:3</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>150</td>
</tr>
<tr>
<td>1:1</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>100</td>
</tr>
</tbody>
</table>

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

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

Bordeline and Noise data
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data

3 kind of artificial problems:

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

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

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

Intrinsic data characteristics

Bordeline and Noise data

Clover data

Paw data
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data

Subclus data
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data

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

Bordeline and Noise data

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


Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Bordeline and Noise data

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

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Original Data</th>
<th>20% of Gaussian Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$TP_{rate}$</td>
<td>$TN_{rate}$</td>
</tr>
<tr>
<td>None</td>
<td>1.000</td>
<td>.9029</td>
</tr>
<tr>
<td>RandomUnderSampling</td>
<td>1.000</td>
<td>.7800</td>
</tr>
<tr>
<td>SMOTE</td>
<td>.9614</td>
<td>.9529</td>
</tr>
<tr>
<td>SMOTE+ENN</td>
<td>.9676</td>
<td>.9623</td>
</tr>
<tr>
<td>SPIDER2</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>
Why is difficult to learn in imbalanced domains?

**Intrinsic data characteristics**

**Bordeline and Noise data**

<table>
<thead>
<tr>
<th>Data set</th>
<th>Base</th>
<th>RO</th>
<th>C4.5 CO</th>
<th>NCR</th>
<th>SP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>subclus-0</td>
<td>0.9540</td>
<td>0.9500</td>
<td>0.9500</td>
<td>0.9460</td>
<td>0.9640</td>
</tr>
<tr>
<td>subclus-30</td>
<td>0.4500</td>
<td>0.6840</td>
<td>0.6720</td>
<td>0.7160</td>
<td>0.7720</td>
</tr>
<tr>
<td>subclus-50</td>
<td>0.1740</td>
<td>0.6160</td>
<td>0.6000</td>
<td>0.7020</td>
<td>0.7700</td>
</tr>
<tr>
<td>subclus-70</td>
<td>0.0000</td>
<td>0.6380</td>
<td>0.7000</td>
<td>0.5700</td>
<td>0.8300</td>
</tr>
<tr>
<td>clover-0</td>
<td>0.4280</td>
<td>0.8340</td>
<td>0.8700</td>
<td>0.4380</td>
<td>0.4860</td>
</tr>
<tr>
<td>clover-30</td>
<td>0.1260</td>
<td>0.7180</td>
<td>0.7060</td>
<td>0.5820</td>
<td>0.7260</td>
</tr>
<tr>
<td>clover-50</td>
<td>0.0540</td>
<td>0.6560</td>
<td>0.6960</td>
<td>0.4460</td>
<td>0.7700</td>
</tr>
<tr>
<td>clover-70</td>
<td>0.0080</td>
<td>0.6340</td>
<td>0.6320</td>
<td>0.5460</td>
<td>0.8140</td>
</tr>
<tr>
<td>paw-0</td>
<td>0.5200</td>
<td>0.9140</td>
<td>0.9000</td>
<td>0.4900</td>
<td>0.5960</td>
</tr>
<tr>
<td>paw-30</td>
<td>0.2640</td>
<td>0.7920</td>
<td>0.7960</td>
<td>0.8540</td>
<td>0.8680</td>
</tr>
<tr>
<td>paw-50</td>
<td>0.1840</td>
<td>0.7480</td>
<td>0.7200</td>
<td>0.8040</td>
<td>0.8320</td>
</tr>
<tr>
<td>paw-70</td>
<td>0.0060</td>
<td>0.7120</td>
<td>0.6800</td>
<td>0.7460</td>
<td>0.8780</td>
</tr>
</tbody>
</table>

Table 14 Performance obtained by C4.5 in the Subclus dataset with and without noisy instances.
Why is difficult to learn in imbalanced domains?

**Intrinsic data characteristics**

**Bordeline and Noise data**

- **SPIDER 2**: allows to get good results in comparison with classical ones.

- It has interest to analyze the use of noise filtering algorithms for these problems: IPF filtering algorithm shows good results.


- 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
- Bordeline 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:

  Training set → Build a Classifier → Accurate Class Prediction

  Test set → Classifier → Accurate Class Prediction
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.

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.
Intrinsic data characteristics

Dataset shift

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

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

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

Sample selection bias and non-stationary environments.

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

Sample selection bias:

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

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

- Training and test following the same data distribution
Causes of Dataset Shift

- DATASET SHIT: Training and test following different data distribution
Causes of Dataset Shift
Sample bias selection: Influence of partitioning on classifiers' performance

<table>
<thead>
<tr>
<th></th>
<th>Iteration 216</th>
<th>Iteration 459</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C45</td>
<td>HDDT</td>
</tr>
<tr>
<td>breast-w</td>
<td>0.9784</td>
<td>0.9753</td>
</tr>
<tr>
<td>bupa</td>
<td>0.6936</td>
<td>0.6913</td>
</tr>
<tr>
<td>credit-a</td>
<td>0.8996</td>
<td>0.8967</td>
</tr>
<tr>
<td>crx</td>
<td>0.8993</td>
<td>0.8877</td>
</tr>
<tr>
<td>heart-c</td>
<td>0.8431</td>
<td>0.8181</td>
</tr>
<tr>
<td>heart-h</td>
<td>0.8756</td>
<td>0.8290</td>
</tr>
<tr>
<td>horse-colic</td>
<td>0.8646</td>
<td>0.8848</td>
</tr>
<tr>
<td>ion</td>
<td>0.9353</td>
<td>0.9301</td>
</tr>
<tr>
<td>krkp</td>
<td>0.9992</td>
<td><strong>0.9993</strong></td>
</tr>
<tr>
<td>pima</td>
<td>0.7781</td>
<td>0.7717</td>
</tr>
<tr>
<td>promoters</td>
<td>0.8654</td>
<td>0.8514</td>
</tr>
<tr>
<td>ringnorm</td>
<td>0.8699</td>
<td>0.8533</td>
</tr>
<tr>
<td>sonar</td>
<td>0.8053</td>
<td>0.7929</td>
</tr>
<tr>
<td>threenorm</td>
<td>0.7964</td>
<td>0.7575</td>
</tr>
<tr>
<td>tic-tac-toe</td>
<td>0.9354</td>
<td>0.9254</td>
</tr>
<tr>
<td>twonorm</td>
<td>0.8051</td>
<td>0.8023</td>
</tr>
<tr>
<td>vote</td>
<td>0.9843</td>
<td>0.9824</td>
</tr>
<tr>
<td>vote1</td>
<td>0.9451</td>
<td>0.9343</td>
</tr>
<tr>
<td>avg. rank</td>
<td>1.11</td>
<td>1.89</td>
</tr>
<tr>
<td>α = 0.10</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>α = 0.05</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

- 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.


Wilcoxon test: Clear differences for both algorithms
Causes of Dataset Shift
Challenges in correcting the dataset shift generated by the sample selection bias
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?

\[ p(x, y) \]

\[ p(x)p(y \mid x) \]

\[ p_{\text{tra}}(y \mid x) \neq p_{\text{tst}}(y \mid x) \]

\[ p_{\text{tra}}(x) \neq p_{\text{tst}}(x) \]

labeling difference

instance difference

labeling adaptation

instance adaptation
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.

Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Data-set shift

GP-RST: From $N$ dimensions to 2
A Genetic-Programming based Feature Selection and RST for Visualization of Fracture between Data

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

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

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

**Algorithm 1** Fitness evaluation procedure

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

Good behaviour. pageblocks 13v4, 1\textsuperscript{st} 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, 1\textsuperscript{st} partition.

Example of mild data fracture

(a) Training set (0.9663)  (b) Test set (0.8660)
Overlap and dataset shift. glass 016v2, 4\textsuperscript{th} 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, 2\textsuperscript{nd} partition

Example of overlap and fracture

(a) Training set (0.6794)  (b) Test set (0.0000)
There are two different potential approaches in the study of the effect and solution of data-set shift in imbalanced domains.

- The first one focuses on intrinsic data-set shift, that is, the data of interest includes some degree of shift that is producing a relevant drop in performance. In this case, we need to:
  - Develop techniques to discover and measure the presence of data-set shift adapting them to minority classes.
  - Design algorithms that are capable of working under data-set shift conditions. These could be either preprocessing techniques or algorithms that are designed to have the capability to adapt and deal with dataset shift without the need for a preprocessing step.
The second branch in terms of data-set shift in imbalanced classification is related to induced data-set shift. Most current state of the art research is validated through stratified cross-validation techniques, which are another potential source of shift in the machine learning process.

*A more suitable validation technique needs to be developed in order to avoid introducing data-set shift issues artificially.*
Why is difficult to learn in imbalanced domains?

Intrinsic data characteristics

Dataset shift

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

Intrinsic data characteristics

What domain characteristics aggravate the problem?

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

http://www.keel.es
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.
We include 111 data sets: 66 for 2 classes, 15 for multiple classes and 30 for noise and borderline.

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.
I. Introduction to imbalanced data sets

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

III. Class imbalance: Data sets, implementations, …

IV. Class imbalance: Trends and final comments
Class Imbalance: Trends and final comments Data level vs algorithm Level

**Data Level Approaches**
- Randomly resampling
- Informatively resampling
- Synthesizing new data
- Combining above methods

**Algorithm Level Approaches**
- Introducing Learning Bias
  - Decision Tree
  - SVMs
  - Associative classification
  - Ftc.
- One Class Learning
  - SVMs
  - BPNNs

**Research Solutions**
- Resampling Data Space
- Adapting Existing Algorithms
- Boosting
- Cost-Sensitive Learning

**Small-Class Boosting**
- RareBoost
- SMOTEBoost
- DataBoost-IM

**Cost-Sensitive Boosting**
- AdaC1
- AdaC2
- AdaC3
- AdaCost
- CSB2

**Weighting Data Space**
- Translation Theorem

**Adapting Learning Algorithms**
- Integrating costs into learning
- Decision making to minimizing costs

---

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

- Intrinsic data characteristics. To analyze the challenges on the class distribution.
In short, it is necessary to do work for:

Establishing some fundamental results regarding:

a) the nature of the problem,

b) the behaviour of different types of classifiers, and

c) the relative performance of various previously proposed schemes for dealing with the problem.

Designing new methods addressing the problem. Tackling data preprocessing and changing rule classification strategy.
Class imbalance is a challenging and critical problem in the knowledge discovery field, the classification with imbalanced data sets.

Due to the intriguing topics and tremendous potential applications, the classification of imbalanced data will continue to receive more and more attention along next years. Class of interest is often much smaller or rarer (minority class).
1. Clases no balanceadas/equilibradas
2. Características intrínsecas de los datos en clasificación
3. Detección de anomalías
4. Problemas no estándar de clasificación: MIL, MLL, ...
5. Análisis de Sentimientos
6. Deep Learning
Tema 1. Introducción a la Inteligencia de Negocio
Tema 2. Minería de Datos. Ciencia de Datos
Tema 3. Modelos de Predicción: Clasificación, regresión y series temporales
Tema 4. Preparación de Datos
Tema 5. Modelos de Agrupamiento o Segmentación
Tema 6. Modelos de Asociación
Tema 7. Modelos Avanzados de Minería de Datos
Tema 8. Big Data