Data Mining and Soft Computing

Francisco Herrera

Research Group on Soft Computing and Information Intelligent Systems (SCI²S)
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
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
8. Some Advanced Topics I: Classification with Imbalanced Data Sets
9. Some Advanced Topics II: Subgroup Discovery
10. Some advanced Topics III: Data Complexity
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 Advanced Topics I: Classification with Imbalanced Data Sets

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
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 classifying 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 introduces 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.
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.

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.
¿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.

Learning in non-balance 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, rare cases are unknown since high dimensional data cannot be visualized to reveal areas of low coverage.

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?

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

Classical evaluation:

Error Rate: \( \frac{\text{FP} + \text{FN}}{N} \)
Accuracy Rate: \( \frac{\text{TP} + \text{TN}}{N} \)

It doesn't take into account the False Negative Rate, which is very important in imbalanced problems.
Learning in non-balanced domains

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: \( \frac{2 \times \text{precision} \times \text{recall}}{\text{recall} + \text{precision}} \)

Learning in non-balanced domains

ROC Curves

The confusion matrix is normalized by columns

<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>

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.

AUC: Área under ROC curve. Scalar quantity widely 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 resampling techniques which allow the adjustment of this parameter.

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 to deal with imbalanced data sets

**Over-Sampling**
- Random
- Focused

**Under-Sampling**
- Random
- Focused

**Cost Modifying**

**Motivation**
- Retain influent examples
- Balance the training set
- Remove noisy instances in the decision boundaries
- Reduce the training set
Data Balancing through re-sampling

- under-sampling
- over-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 \textit{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 $d(E_l, E_i) < d(E_i, E_j)$ 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 $E'$ contain 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: |
| \[ f_{1,1} = 6 \quad f_{2,1} = 4 \quad f_{2,1} - f_{1,1} = -2 \] |
| \[ f_{1,2} = 4 \quad f_{2,2} = 3 \quad f_{2,2} - f_{1,2} = -1 \] |
| The new samples will be generated as |
| \[ (f_{1}',f_{2}') = (6,4) + \text{rand}(0-1) \times (-2,-1) \] |
| \text{rand}(0-1) generates a random number between 0 and 1. |
State-of-the-art algorithm: SMOTE.


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

- Minority sample
- Synthetic sample
- Majority sample
State-of-the-art algorithm: SMOTE.
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.

<table>
<thead>
<tr>
<th>Data set</th>
<th>1°</th>
<th>2°</th>
<th>3°</th>
<th>4°</th>
<th>5°</th>
<th>6°</th>
<th>7°</th>
<th>8°</th>
<th>9°</th>
<th>10°</th>
<th>11°</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>Tmk</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>NCL</td>
<td>RdUdr</td>
<td>Tmk</td>
<td>OSS*</td>
<td>CNN*</td>
<td>Original*</td>
<td>CNN+Tmk*</td>
<td></td>
</tr>
<tr>
<td>Splice-ei</td>
<td>RdOvr</td>
<td>Original</td>
<td>Tmk</td>
<td>Smt</td>
<td>CNN</td>
<td>NCL</td>
<td>Smt+Tmk</td>
<td>Smt+ENN*</td>
<td>CNN+Tmk*</td>
<td>RdUdr*</td>
<td>OSS*</td>
</tr>
<tr>
<td>Splice-eii</td>
<td>Smt+ENN</td>
<td>Smt+Tmk</td>
<td>CNN+Tmk</td>
<td>OSS</td>
<td>Tmk</td>
<td>CNN</td>
<td>Original</td>
<td>RdUdr</td>
<td>CNN</td>
<td>NCL</td>
<td>Original</td>
</tr>
<tr>
<td>Vehicle</td>
<td>RdOvr</td>
<td>Smt+Tmk</td>
<td>OSS</td>
<td>CNN</td>
<td>Tmk</td>
<td>CNN</td>
<td>NCL*</td>
<td>Smt+ENN*</td>
<td>RdUdr*</td>
<td></td>
<td></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>Tmk*</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>RdUdr</td>
<td>CNN*</td>
<td>OSS*</td>
<td>Tmk*</td>
<td>Original*</td>
<td>NCL*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glass</td>
<td>Smt+ENN</td>
<td>RdOvr</td>
<td>NCL</td>
<td>Smt+Tmk</td>
<td>Tmk</td>
<td>CNN+Tmk*</td>
<td>OSS*</td>
<td>CNN*</td>
<td>CNN</td>
<td>RdUdr*</td>
<td></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>Tmk*</td>
<td>Smt*</td>
<td>CNN+Tmk*</td>
<td>Smt+ENN*</td>
<td>RdUdr*</td>
</tr>
</tbody>
</table>

State-of-the-art algorithm: SMOTE.

Adaptive Synthetic Minority Oversampling Method (ASMO)

- Clustering
  - 2-class sample generation

: Minority sample

: Majority sample

: Synthetic sample
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*).

Some Advanced Topics I: Classification with Imbalanced Data Sets

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

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:

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:


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

Selection

Representation

Initialization

Population

Fitness function

Replacement

POPULATION

PARENTS

Crossover

Mutation

Replacement

DESCENDANTS

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

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
  - Evolutionary Balancing Under-Sampling
    - Global Selection
    - Majority Selection
      - EBUS-3S-GM
      - EBUS-GS-GM
      - EBUS-MS-GM
      - EBUS-MS-AUC
  - Evolutionary Under-Sampling guided for Classification Measures
    - Global Selection
    - Majority Selection
      - EUROSM-GS-GM
      - EUROSM-GS-AUC
      - EUROSM-MS-GM
      - EUROSM-MS-AUC
Evolutionary Under-Sampling

Fitness function in EBUS model:

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

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

\(P\): 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:

\[
\text{Fitness}(S) = g,
\]

\[
\text{Fitness}(S) = \text{AUC},
\]
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-Sampling  TomekLinks (TL)
CNN  OSS  CNN+TL  NCL
CPM  SBC
## Experimental Framework and Results

### Data sets:

<table>
<thead>
<tr>
<th>Data set</th>
<th>#Examples</th>
<th>#Attributes</th>
<th>Class (min., maj.)</th>
<th>%Class (min., maj.)</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlassBWNFP</td>
<td>214</td>
<td>9</td>
<td>(build-window-non_float-proc, remainder)</td>
<td>(35.51, 64.49)</td>
<td>1.82</td>
</tr>
<tr>
<td>EcoliCP-IM</td>
<td>220</td>
<td>7</td>
<td>(im, cp)</td>
<td>(35.00, 65.00)</td>
<td>1.86</td>
</tr>
<tr>
<td>Pima</td>
<td>768</td>
<td>8</td>
<td>(1, 0)</td>
<td>(34.77, 66.23)</td>
<td>1.9</td>
</tr>
<tr>
<td>GlassBWFP</td>
<td>214</td>
<td>9</td>
<td>(build-window-floating-proc, remainder)</td>
<td>(32.71, 67.29)</td>
<td>2.06</td>
</tr>
<tr>
<td>German</td>
<td>1000</td>
<td>20</td>
<td>(1, 0)</td>
<td>(30.00, 70.00)</td>
<td>2.33</td>
</tr>
<tr>
<td>Haberman</td>
<td>306</td>
<td>3</td>
<td>(Die, Survive)</td>
<td>(26.47, 73.53)</td>
<td>2.68</td>
</tr>
<tr>
<td>Splice-ie</td>
<td>3176</td>
<td>60</td>
<td>(ie, remainder)</td>
<td>(24.09, 75.91)</td>
<td>3.15</td>
</tr>
<tr>
<td>Splice-ei</td>
<td>3176</td>
<td>60</td>
<td>(ei, remainder)</td>
<td>(23.98, 76.02)</td>
<td>3.17</td>
</tr>
<tr>
<td>GlassNW</td>
<td>214</td>
<td>9</td>
<td>(non-windows glass, remainder)</td>
<td>(23.93, 76.17)</td>
<td>3.19</td>
</tr>
<tr>
<td>VehicleVAN</td>
<td>846</td>
<td>18</td>
<td>(van, remainder)</td>
<td>(23.52, 76.48)</td>
<td>3.25</td>
</tr>
<tr>
<td>EcoliM</td>
<td>336</td>
<td>7</td>
<td>(im, remainder)</td>
<td>(22.92, 77.08)</td>
<td>3.36</td>
</tr>
<tr>
<td>New-thyroid</td>
<td>215</td>
<td>5</td>
<td>(hypo, remainder)</td>
<td>(16.28, 83.72)</td>
<td>4.92</td>
</tr>
<tr>
<td>Segment1</td>
<td>2310</td>
<td>19</td>
<td>(1, remainder)</td>
<td>(14.29, 85.71)</td>
<td>6.00</td>
</tr>
<tr>
<td>EcoliMU</td>
<td>336</td>
<td>7</td>
<td>(IMU, remainder)</td>
<td>(10.42, 89.58)</td>
<td>8.19</td>
</tr>
<tr>
<td>Optdigits0</td>
<td>5564</td>
<td>64</td>
<td>(0, remainder)</td>
<td>(9.90, 90.10)</td>
<td>9.10</td>
</tr>
<tr>
<td>Satimage4</td>
<td>6435</td>
<td>36</td>
<td>(4, remainder)</td>
<td>(9.73, 90.27)</td>
<td>9.28</td>
</tr>
<tr>
<td>Vowel0</td>
<td>990</td>
<td>13</td>
<td>(0, remainder)</td>
<td>(9.01, 90.99)</td>
<td>10.1</td>
</tr>
<tr>
<td>GlassVWFP</td>
<td>214</td>
<td>9</td>
<td>(V-es-windows float-proc, remainder)</td>
<td>(7.94, 92.06)</td>
<td>10.39</td>
</tr>
<tr>
<td>EcoliOM</td>
<td>336</td>
<td>7</td>
<td>(om, remainder)</td>
<td>(6.74, 93.26)</td>
<td>13.84</td>
</tr>
<tr>
<td>GlassContainers</td>
<td>214</td>
<td>9</td>
<td>(containers, remainder)</td>
<td>(6.07, 93.93)</td>
<td>15.47</td>
</tr>
<tr>
<td>Abalone14</td>
<td>281</td>
<td>9</td>
<td>(16, ?)</td>
<td>(5.75, 94.25)</td>
<td>16.08</td>
</tr>
<tr>
<td>GlassTableware</td>
<td>214</td>
<td>9</td>
<td>(tableware, remainder)</td>
<td>(4.2, 95.8)</td>
<td>22.81</td>
</tr>
<tr>
<td>YeastCYT-POX</td>
<td>483</td>
<td>8</td>
<td>(POX, CYT)</td>
<td>(4.14, 95.86)</td>
<td>23.15</td>
</tr>
<tr>
<td>YeastME2</td>
<td>1484</td>
<td>8</td>
<td>(ME2, remainder)</td>
<td>(3.43, 96.57)</td>
<td>28.41</td>
</tr>
<tr>
<td>YeastME1</td>
<td>1484</td>
<td>8</td>
<td>(ME1, remainder)</td>
<td>(2.96, 97.04)</td>
<td>32.78</td>
</tr>
<tr>
<td>YeastEXC</td>
<td>1484</td>
<td>8</td>
<td>(EXC, remainder)</td>
<td>(2.49, 97.51)</td>
<td>39.16</td>
</tr>
<tr>
<td>Car</td>
<td>1728</td>
<td>6</td>
<td>(good, remainder)</td>
<td>(3.99, 96.01)</td>
<td>71.94</td>
</tr>
<tr>
<td>Abalone19</td>
<td>4177</td>
<td>9</td>
<td>(19, remainder)</td>
<td>(0.77, 99.23)</td>
<td>128.87</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th></th>
<th>True = 0</th>
<th>True = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predict = 0</td>
<td>$C(0,0)$</td>
<td>$C(0,1)$</td>
</tr>
<tr>
<td>Predict = 1</td>
<td>$C(1,0)$</td>
<td>$C(1,1)$</td>
</tr>
</tbody>
</table>

- cost that depends on particular example $x$

<table>
<thead>
<tr>
<th></th>
<th>True = 0</th>
<th>True = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predict = 0</td>
<td>$C(0,0,x)$</td>
<td>$C(0,1,x)$</td>
</tr>
<tr>
<td>Predict = 1</td>
<td>$C(1,0,x)$</td>
<td>$C(1,1,x)$</td>
</tr>
</tbody>
</table>
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 disjuncts

Facet-wise analysis of the problems

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


CONCLUSIONS: Methods that deal with class imbalances and 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 SCI²S.

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


- 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.


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 contribution


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.

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
8. Some Advanced Topics I: Classification with Imbalanced Data Sets
9. Some Advanced Topics II: Subgroup Discovery
10. Some advanced Topics III: Data Complexity