Dealing with Severely Imbalanced Data

William Klement[1], Szymon Wilk[2,3], Wojtek Michalowski[2], Stan Matwin[1,4]

Mobile Emergency Triage (MET) Research Program

[1] School of Information Technology and Engineering
University of Ottawa, Canada

[2] Telfer School of Management, University of Ottawa, Canada

[3] Laboratory of Intelligent Decision Support Systems
Poznan University of Technology, Poland

[4] Institute of Computer Science, Polish Academy of Sciences, Poland

Ontario Centres of Excellence

1. Motivation
2. ML Solutions
   - Modify the data
   - Modify the learning
3. Methodology
   - Classification models
4. Experiments
   - The Data
   - Results
5. Conclusions

Motivation: Medical classification challenges

- Does a child with head trauma require a CT scan for neurological assessment in a hospital?
- Serious head trauma is rare.
- In practice, about 95% of children are fine and do not require CT scans;
- Too many CT scans are not good for children;
- Other domains: fraud detection, environmental disasters, etc.

Machine learning struggles

- Imbalance: minority vs. majority class.
- Insufficient data: small minority class.
- The lack of instances (very few people have a given disease or very few disasters)
- Severely imbalanced problems contain all the above.
- Machine learning methods appear to predict the majority class because of two assumptions:
  1. the goal is to maximize accuracy, and
  2. the classifier will operate on data drawn from the same distribution as the training data.

Machine learning solutions

To improve classifier's performance on the minority class, we can modify:

- the class distribution in the training data, or
- the classification algorithm to handle the imbalance.

Modifying the class distribution

We can:

- increase the frequency of the minority class by over-sampling, or
- decrease the frequency of the majority class by under-sampling.
- But! Under-sampling is better than over-sampling (Drummond & Holte 2005).
- Then, what is the correct class distribution?, and
- What is the appropriate sampling strategy?

Modifying the learning algorithm

- We can adjust the classification threshold. Then, we need:
  - probability estimates or ranking scores,
  - higher sensitivity, and
  - acceptable specificity
- Cost-sensitive learning
- Learn each class separately (one-class learning)

Our methodology

- Using Bayesian learning, we focus on three methods:
  - under-sampling the minority class,
  - adjusting the classification threshold to maximize the F-measure (maximizes precision and recall), and
  - combining an ensemble of 10 classifiers.
- Our objective is to determine which combination of the above techniques produces better classification performance on imbalanced data.

The Naive Bayes classification models

Model	Under-sampling Only	Threshold Selection	Voting an Ensemble (with under-sampling)
1	✖	✖	✖
2	✖	✔	✖
3	✔	✖	✖
4	✖	✖	✔
5	✔	✔	✖
6	✖	✔	✔

Experiment design

- Select data sets from the UCI repository
- The minority class is no more than 20% of the data
- When applicable, randomly under-sample the training data (without replacement) using a sampling ratio (s) equal to twice the size of the minority class.
- The test set retains the original class distribution.
- Use 10-fold cross validation.
- Repeat 10 $\times$ 10-fold cross validation.
- Record accuracy, sensitivity, specificity, and AUC.
- Compute the average and standard deviation.

The UCI data sets

Code	Data Set	n	n^+	n^-	$\frac{n^+}{n}$	s
A	dis	3772	58	3714	1.54	3.0
B	ozone-onehr	2536	73	2463	2.88	5.7
C	hyperthyroid	3163	151	3012	4.77	9.5
D	sick	3772	231	3541	6.12	12.2
E	ozone-eithhr	2534	160	2374	6.31	12.6
F	sick-euthyroid	3163	293	2870	9.26	18.5
G	spect	267	55	212	20.60	41.0
H	hepatitis	155	32	123	20.65	41.2

For n instances, consisting of n^+ positives and n^- negatives, $\frac{n+}{n}$ is the size of the minority class as a percentage of n, and s is the percentage of n being sampled (when applicable).

Almost equal AUC standard deviations

Code	Model	Accuracy	Sensitivity	Specificity	AUC
A	1	95.8 (0.9)	44.8 (17.1)	96.6 (0.9)	83.5 (**12.3**)
	2	95.2 (1.2)	60.7 (**22.4**)	95.8 (1.2)	83.5 (**12.3**)
	3	59.8 (14.8)	83.8 (15.7)	59.4 (15.0)	84.7 (**10.5**)
	4	62.6 (8.5)	81.9 (17.7)	62.3 (8.7)	85.5 (**11.2**)
	5	81.9 (12.5)	79.3 (18.3)	81.9 (12.7)	84.7 (**10.5**)
	6	82.6 (6.0)	**80.9** (18.6)	**82.6** (6.2)	85.7 (**11.2**)
F	1	84.1 (2.2)	89.9 (5.7)	83.5 (2.4)	92.0 (3.2)
	2	93.6 (1.3)	65.8 (**9.0**)	96.5 (1.5)	92.0 (3.2)
	3	71.5 (3.6)	92.4 (4.9)	69.3 (4.1)	92.1 (3.1)
	4	71.1 (2.5)	92.5 (4.9)	68.9 (2.8)	92.2 (3.2)
	5	85.5 (3.0)	87.9 (6.8)	85.2 (3.4)	92.1 (3.1)
	6	83.3 (2.1)	**90.6** (5.8)	**82.6** (2.2)	92.2 (3.2)
G	1	78.7 (7.4)	75.8 (18.1)	79.4 (8.2)	85.1 (**8.9**)
	2	82.4 (5.5)	66.2 (**21.5**)	86.6 (6.5)	85.1 (**8.9**)
	3	71.8 (8.3)	79.9 (17.2)	69.6 (10.2)	85.0 (**8.7**)
	4	71.8 (8.3)	79.7 (17.7)	69.7 (10.5)	85.0 (**8.8**)
	5	71.9 (11.9)	77.1 (19.6)	70.5 (16.5)	85.0 (**8.7**)
	6	73.1 (7.7)	**77.1** (18.1)	**72.0** (10.1)	85.0 (**8.8**)
H	1	83.8 (9.7)	70.0 (25.6)	87.4 (10.5)	85.7 (**12.8**)
	2	83.8 (9.7)	70.8 (**25.8**)	87.2 (10.6)	85.7 (**12.8**)
	3	79.9 (10.8)	80.3 (21.3)	79.7 (12.3)	86.3 (**12.2**)
	4	80.6 (11.9)	80.5 (22.6)	80.7 (14.3)	87.8 (**12.2**)
	5	78.1 (11.3)	80.7 (20.7)	77.4 (12.8)	86.3 (**11.7**)
	6	79.7 (11.9)	**80.8** (20.8)	**79.4** (13.7)	87.8 (**11.9**)

- A very small number of positives results in higher standard deviation of AUC for all models (A,G,H vs. F).
- The naive Bayes classifier alone struggles (confirmed).
- Using classification threshold selection alone (model 2) produces a high standard deviation on sensitivity.
- Combining all three methods (model 6) produces the best balance of sensitivity and specificity.
- Combining threshold selection with under-sampling (model 5), at the least, produces acceptable balance of sensitivity and specificity.
- The accuracy decreases as the performance on the minority class improves (expected).

Varying standard deviations of AUC

Code	Model	Accuracy	Sensitivity	Specificity	AUC
B	1	70.8 (3.2)	81.4 (14.6)	70.5 (3.4)	84.2 (6.4)
	2	90.1 (2.2)	51.4 (**18.6**)	91.2 (2.1)	84.2 (6.4)
	3	68.3 (3.5)	83.2 (13.7)	67.8 (3.7)	83.9 (6.5)
	4	68.2 (3.3)	82.8 (13.6)	67.7 (3.5)	84.5 (6.5)
	5	58.7 (9.8)	88.6 (14.9)	57.9 (10.1)	83.1 (10.1)
	6	65.7 (7.1)	**84.2** (14.6)	**65.2** (7.2)	83.7 (10.1)
C	1	97.9 (0.7)	77.3 (9.3)	98.9 (0.6)	97.8 (2.8)
	2	97.8 (0.7)	78.2 (10.4)	98.8 (0.7)	97.8 (2.8)
	3	96.3 (1.1)	93.5 (7.5)	96.4 (1.2)	98.2 (2.3)
	4	96.4 (0.9)	93.9 (6.6)	96.5 (1.0)	98.6 (1.9)
	5	93.9 (9.5)	96.3 (10.2)	93.7 (9.5)	97.2 (9.9)
	6	95.6 (1.2)	**96.9** (4.4)	**95.5** (1.3)	98.6 (1.9)
D	1	92.7 (1.4)	77.9 (8.2)	93.7 (1.3)	92.6 (3.8)
	2	95.6 (1.1)	58.8 (**10.4**)	98.0 (1.1)	92.6 (3.8)
	3	83.0 (3.5)	89.6 (7.0)	82.6 (3.7)	92.6 (3.5)
	4	82.6 (8.3)	88.7 (10.6)	82.2 (8.2)	92.8 (9.5)
	5	86.4 (3.0)	87.4 (8.1)	86.4 (3.4)	92.6 (3.5)
	6	86.4 (1.7)	**88.3** (7.3)	**86.2** (1.8)	**93.8** (3.0)
E	1	67.5 (2.8)	85.1 (8.3)	66.4 (3.0)	82.8 (4.2)
	2	87.2 (2.2)	48.5 (**12.1**)	89.9 (2.4)	82.8 (4.2)
	3	65.3 (3.0)	**85.7** (8.3)	**63.9** (3.3)	82.5 (4.1)
	4	64.8 (6.3)	**84.8** (11.1)	**63.5** (6.3)	82.1 (8.5)
	5	63.4 (9.7)	**84.9** (11.8)	**61.9** (10.2)	81.7 (8.8)
	6	65.6 (3.1)	**85.4** (8.5)	**64.3** (3.4)	82.8 (4.3)

- Under-sampling can increase the standard deviation of the AUC.
- Classification threshold selection alone (model 2) results in a high standard deviation of sensitivity.
- Combining at least two of the three methods (models 5 and 6) result in the best balanced performance on both classes.
- Combining all three methods is always the same or better in balancing the performance.

Conclusions & Future work

When dealing with severely imbalanced data, we should:

- aim to balance the performance on both classes.
- combine at least two of threshold selection, sampling and ensembles. A combination of all three is a wise choice.
- ignore accuracy, it is misleading.
- desire a low standard deviation of the AUC.

What's next?

- Cost sensitive learning, can it improve on these results? or can it be combined with it?
- When the costs are unknown, why not try all costs to see how these three methods compare.
- How does the one-class learning perform against this combination of three techniques?