Imbalanced Data Issues

Zonghong Liu

Rutgers University

April 23, 2019

Overview

Introduction

- What is imbalanced data?
  In a data set, the number of observations belonging to one class is significantly higher(or lower) than those belonging to the other classes.
- What can imbalanced data cause?
  It can compromise the performance of most standard machine learning algorithms.
- Why?
  Most standard algorithms assume equal misclassification costs.

Example in Our Data

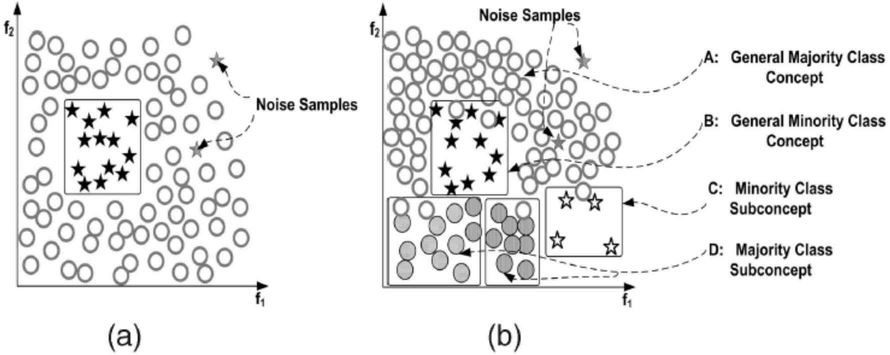
Summary of the data

Class	CLEAVED	MIDDLE	UNCLEAAVED
Number	1931	1382	5410

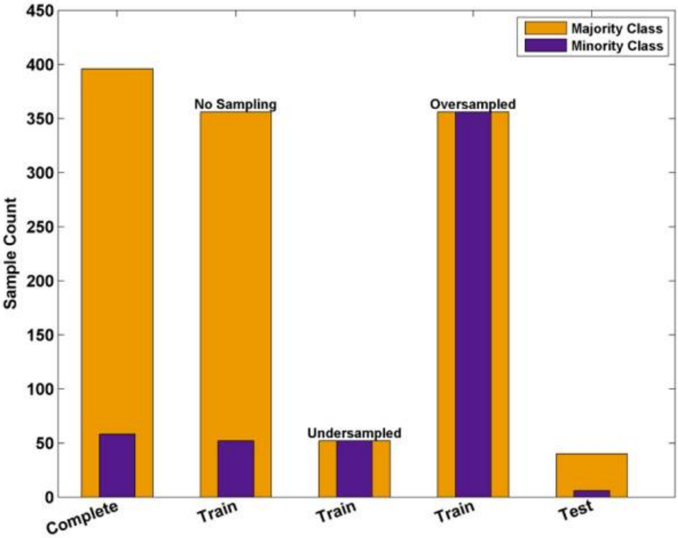
Confusion Matrix given by SVM

	CLEAVED	MIDDLE	UNCLEAAVED
CLEAVED	514	31	105
MIDDLE	27	156	36
UNCLEAVED	77	275	1687

Between-class and Within-class Imbalances



Random Oversampling and Undersampling



Random Oversampling and Undersampling

- Disadvantage of undersampling
  - Removing the examples from the majority class may cause the classifier to miss important concepts pertaining to the majority class.
- Disadvantages of oversampling
  - Time consuming
  - Overfitting

    When classifier produces multiple clauses in a rule for multiple copies of the same example, it may cause the rule to become too specific.

Informed undersampling – EasyEnsemble

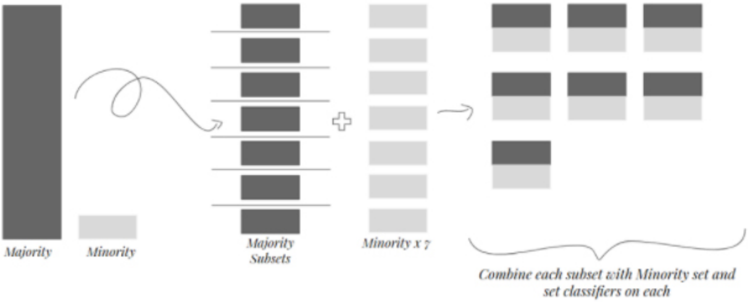


Figure-1: Undersampling for Ensemble Learners Illustration

Informed undersampling – BalanceCascade

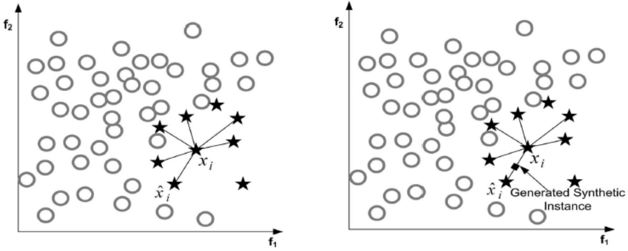
To some extent, BalanceCascade can be seen as a sequential ensemble learning method, compared to EasyEnsemble, which is a parallel ensemble learning method.

Algorithm

1. Randomly sample a set E from majority class, s.t. |E|=|S_{min}|.
2. Train the model using $N = E \cup S_{min}$, denoted as M_1.
3. Remove those observations which are correctly classified by M_1 from S_{maj}.
4. Sample an observation set E from S_{maj}, s.t. |E|=|S_{min}|, back to step 2.
5. Repeat 2-4.

Synthetic Minority Oversampling Technique (SMOTE)

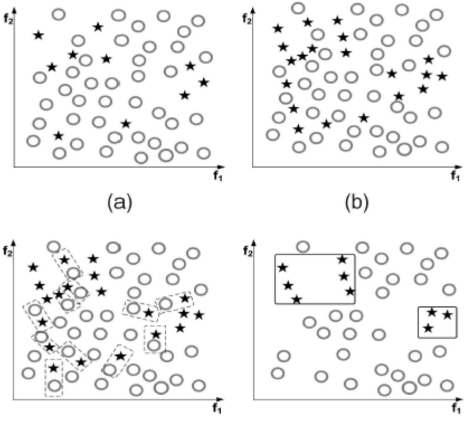
- For each $x_i \in S_{min}$, find the k-nearest neighbors for some k, denoted as K_i.
- Randomly choose one of the k elements in K_i, denoted as $\hat{x}_i$, generate new data by:
  $x_{new} = c\hat{x}_i + (1-c)x_i, \ \textit{for some } c \in (0,1).$



(a) (b)

Sampling with Data Cleaning Techniques

- Tomek links
  For $x_i \in S_{min}$ and $x_j \in S_{maj}$, the pair (x_i, x_j) is called Tomek link if there is no x_k s.t. $\mathrm{d}(x_i, x_k) < \mathrm{d}(x_i, x_j)$ or $\mathrm{d}(x_k, x_j) < \mathrm{d}(x_i, x_j)$



(a) (b)

(c) (d)

Framework of Cost Sensitive Learning

- Cost matrix
  - Can be viewed as numerical representation of the penalty of classifying examples from one class to another.
  - Define C(i,j) as the cost of misclassifying a jth class observation as an ith class observation.
  - Typically, there is no cost for correct classification.
  - Generally, C(M_{maj}, M_{min}) should be larger than C(M_{min}, M_{maj}).
- Implementing cost-sensitive learning
  - One can incorporate cost function directly into the learning algorithm to fit the cost-sensitive model.
  - There is no unifying framework for cost-sensitive learning.
  - We will take AdaBoost as an example

Cost-Sensitive Adaptive Boosting

- In original AdaBoost algorithm, the weight is updated by:

$$w_i^{m+1} = w_i^m e^{-\alpha_m y_i G_m(x_i)} \tag{1}$$

- In cost-sensitive AdaBoost algorithm, the weight is updated in the following way:

$$w_i^{m+1} = w_i^m e^{-\alpha_m C_i y_i G_m(x_i)} \tag{2}$$

  - Note here AdaBoost is a two-class classifier, C_i is the cost that x_i is being misclassfied, and should take higher value if x_i belongs to the minority class.

Similarity Between Sampling and Cost-Sensitive Methods

- Cost-Sensitive Learning

$$min\ L(y, f(x)) = min\ \sum_i C_i l(y_i, f(x_i)) \tag{3}$$

- Sampling Methods

$$min\ L(y, f(x)) = min\ \sum_i \sum_{j=1}^{n_i} l(y_i, f(x_i)) = min\ \sum_i n_i l(y_i, f(x_i)) \tag{4}$$

Evaluation Measure

- F-score
- G-mean function
- Balanced error rate
  - Similarity to the sampling methods.

References

He H. et al. (2008)
Learning from imbalanced data.
IEEE Trans. Knowl. Data Eng. 1263 – 1684.

Bilal M. et al. (2019)
Machine learning and integrative analysis of biomedical big data.
Genes 2019, 10, 87.

Thank You!