Example-dependent cost-sensitive regression

Rahul Verma
AI23MTECH11008
Indian Institute of Technology, Hyderabad
Email: ai23mtech11008@iith.ac.in

Pratik Yawalkar
AI23MTECH11006
Indian Institute of Technology, Hyderabad
Email: ai23mtech11006@iith.ac.in

Sarang Kukade
EM23MTECH11008
Indian Institute of Technology, Hyderabad
Email: em23mtech11008@iith.ac.in

Mahesh Deshmukhe
CC23MTECH11003
Indian Institute of Technology, Hyderabad
Email: cc23mtech11003@iith.ac.in

Aniruddha Paradkar
AI23MTECH13001
Indian Institute of Technology, Hyderabad
Email: ai23mtech13001@iith.ac.in

Code can be found here here.

Abstract

The incorporation of a machine learning factor "example-dependent cost-sensitive regression" makes it possible to reflect unequal costs for different examples in regression problems. Such practical instances are where the faults are haphazardly and not uniformly distributed are the areas where the traditional regression techniques are not compliant to assumption that uniform costs are incurred in every case. First of all, researchers should come up with the idea of model training framework which is robust and though at the same time, incorporates the cost specific scenarios during training of the model. The next part of the paper often lists the goal, approach, and main outcome of a study in general. In this paper, we have utilized two crucial techniques which are: Bahnsen and Nikou-Gunnemann's approach.

1. Problem Statement

Just as financial transactions are important for our everyday lives, huge number of transactions are processed online on a daily basis. In addition to this, the exploration of such interactions in order to detect the fraud or deviant behavior is a rather complex and extremely time-consuming process since the volume of payment data is constantly growing. As the problem of fraud is becoming more complicated, there are more and more requirements for the higher level of fraud detection, which were not able to deal with the problems of the traditional methods.

Due to the development of machine learning algorithms over the years, there has been a constant growth in the accuracy rate of fraud detection, where patterns and abnormalities in finances are recognized. However, the most effective method is producing some costs induced for every transaction in the algorithm in order to improve the loss function and make better predictions. This will enable us to detect fraud by pinpointing something out of the ordinary that might be as a result of fraud.

The fact that payment data is usually complex and dimensional of high order poses some special conversion challenges. This is why it is critical that inexpensive methods that reduce the complexity of data are used. These techniques by applying some alterations have better outcomes which is a great step towards fraud detection.

2. Description of the dataset

The dataset contains one file, 'cost-sensitive regression.csv', with 1,47,636 sample data points and 13 features. Each sample represents a specific transaction. Details of each feature are listed below:

– 'not count' (Integer): Number of transactions that were counted as 'No'

– ‘yes count‘ (Integer): Number of transactions that were counted as ’Yes’

– ‘ATPM‘ (Float): Average duration per transaction, which is calculated in seconds.

– ‘PFD‘ (Float): Fraudulent data in the transactions percentage.

– ‘PFG‘ (Float): Percentage of transactions that get identified as fraudulent by the system.

– ‘SFD‘ (Float): Fraudulent data for all transactions can be modeled.

– ‘SFG‘ (Float): Flagging of transaction as a fraud.

– ‘WP‘ (Float): The weighted share of fraudulent transactions among all the transactions.

– ‘WS‘ (Float): Summed weighing of flagged fraudulent transactions.

– ‘AH‘ (Float): Averaged amount per transaction

– ‘AN‘ (Float): Mean number of transactions per account (MTA).

– ‘STATUS‘ (Integer): Status code describing the outcome or the state of each transaction execution.

– ‘FNC‘ (Float): Ratio of fraudulent-to-non-fraudulent transactions (cost).

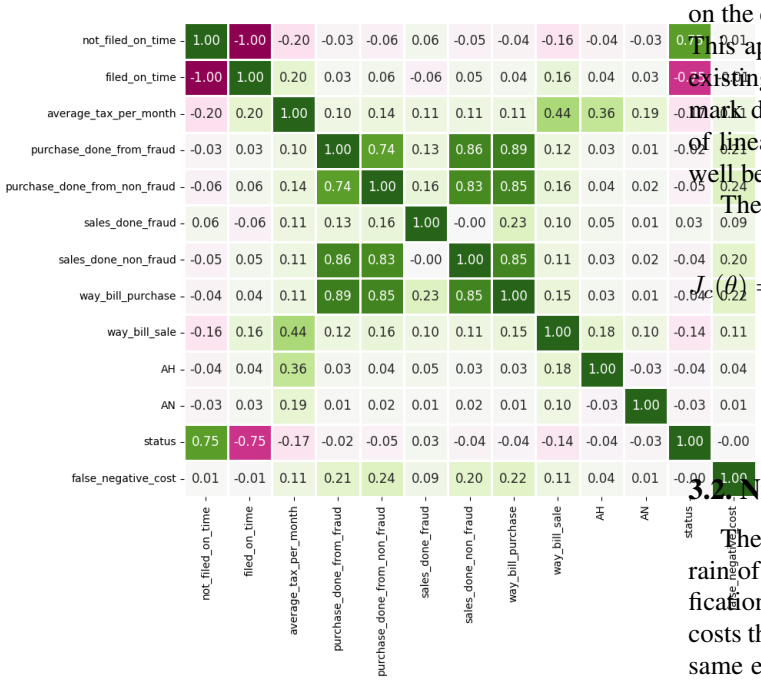

Figure 1. Plot of Correlation between columns

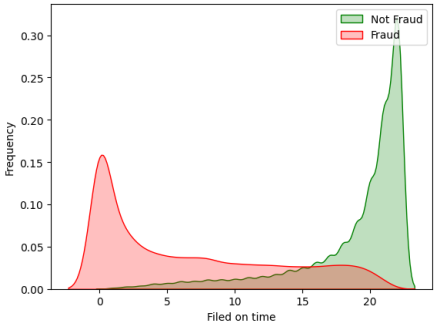

Figure 2. KDE Plot of Fraud Vs Non-Fraud Dataset

3. Cost-sensitive regression methods

In this section, we will explore two different methods to perform cost-sensitive logistic regression namely, **Bahnsen approach** and **Nikou Gunnemann's approach** .

3.1. Bahnsen approach

The Bahnsen approach deals with cost sensitive logistic modeling problem in which the mis-classification costs are different for different data points. Traditionally used methods may not take into account the differences between individuals, thus leading to the ineffectiveness of the treatment. The last part of their suggestion is the development of an example-dependent gradient matrix for credit scoring which is unique method. Besides that, they have developed an algorithm which puts these cost terms, that are based on the examples, directly into the logistic regression model. This approach is able to effectively save cost compared to existing methods, when performing tests with the benchmark datasets. But the authors point out that an actual use of linear loss function may not let the system differentiate well between correct and incorrect classifications.

The objective function used in the method:

$$J_{ac}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\Big[y_i\Big(h_\theta(x_i)\cdot CTP_i + (1-h_\theta(x_i))\cdot CFN_i\Big) + (1-y_i)\Big(h_\theta(x_i)\cdot CFP_i + (1-h_\theta(x_i))\cdot CTN_i\Big)\Big] \quad (1)$$

3.2. Nikou Gunnemann's approach

The idea of Nikou Gunnemann about remedying the terrain of cost-sensitive logistics involved amid binary classification problems is presumably to reduce the inequality in costs that diverse categories have. Classification errors with same error-rate are assumed to cost the same in each class in the traditional form of logistic regression. Nevertheless, in the real situations, disparity of the false positive cost and

false negative cost be very huge that it doesn't hold the assumption the cost mis-classification for the real and false diagnosis to be equally. By developing such a new technique, Gunnemann enriched the logistic regression model by considering class-proportional costs to measure every class prediction not only using prediction accuracy but also class wise cost from true/false positive. Hence, the abstract the main motives, methods, and results must also be revealed since the proposed method was used in contributing to unwrapping tags, building up classification accuracy when the scenario distribution is considered.

The objective function used in the method:

$$\sum_{i=1}^{n}[a_i \cdot y_i \cdot (-\log f(g(x_i, \beta)) \cdot b_i)] + [a_i \cdot (1 - y_i) \cdot (-\log(1 - f(g(x_i, \beta))) \cdot b_i)] \quad (2)$$

where $a_i = c_i and b_i = 1$

4. Optimization algorithm

Optimization algorithms –heuristics and algorithms which are used to find the best solution among the available alternatives. They are thrust all over the place in the different fields to fulfill their purpose of reducing or increasing objective functions while complying with constraints. These algorithms comprise varying methods, for example, gradient descent and 21st century genetic algorithms, as well as particle swarm optimization and others. Therefore, stochastic optimization is applicable in various fields like mathematics, computer science, engineering, and finance tackling problems such as parameter tuning in machine learning to the problem of resource allocation in logistics.

4.1. Genetic Algorithm

Genetic algorithms are often used for optimization problems where traditional gradient-based methods are not suitable, such as problems with non-linear and non-convex objective functions, discrete decision variables, or complex search spaces. They offer a robust and flexible approach to finding near-optimal solutions in a wide range of applications.

1. Initialization:
    - Building an initial population of the suggested solutions (they are often referred to as individuals, or genes) is a good point to start. Each of the individuals symbolises the potential for an answer to the optimization problem.
    - The size of the population is primarily defined before the start of the simulation and may vary depending on the level of problem complexity.

Algorithm 1 Genetic Algorithm

```
Initialize population P with random individuals
while termination criterion not met do
    Evaluate fitness of each individual in P
    Select parents from P based on fitness
    Apply crossover to generate offspring
    Apply mutation to offspring
    Replace least fit individuals in P with offspring
end while
return Best individual(s) found in P
```

2. Evaluation:
    - Evaluate each individual in the population using a predefined fitness function. The fitness function quantifies how well each individual solves the optimization problem.
    - The fitness function can be problem-specific and is designed to capture the objectives and constraints of the optimization problem.
3. Selection:
    - Nominate selected members of the existing population to be parents in the next generation.
    - Probabilistic selection is implemented which means that individuals with higher fitness have a greater chance to be selected.
4. Crossover:
    - To produce new individuals (offspring) combine genetic information of the selected parents.
    - The crossover procedure generally includes swapping sub-solutions between parents.
5. Mutation:
    - Apply random alterations of genetic information for specific members to acquire the desired diversity.
    - The mutation in the population prevents premature convergence and let the population explore more sections of the search space.
6. Replacement:
    - Substitute those of the lowest rank in the contemporary group with the newly formed offspring.
    - The replacement scheme may be different, like selecting individuals with the lowest fitness or having the elite individuals sustain to the next generation.

7. Termination:

   - Do selection, recombination, mutation, and replacement steps in repeating rounds until termination conditions are satisfied.
   - If we want to end our run algorithm for example with the number of generations reached, reaching a satisfactory level of fitness, or using up computer resources, we should set a criterion.

8. Result: Once the termination criterion is met, the genetic algorithm returns the best individual(s) found during the optimization process as the solution(s) to the optimization problem.

5. Results

5.1. Results from Bahnsen approach

The Accuracy score was 75.75 % and cost score was 4566953.39

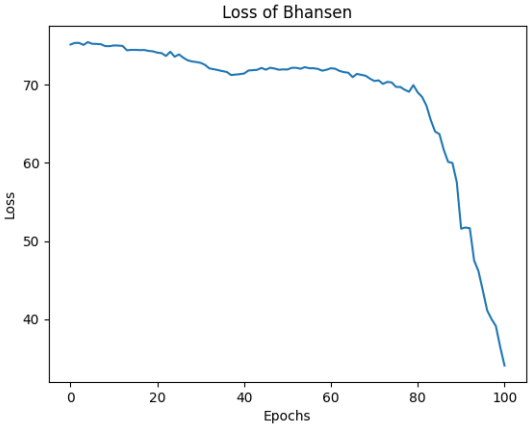

Figure 3. Loss of Bhansen

5.2. Results from Nikou Gunnemann's approach

The Accuracy score was 61.14 % and cost score was 14016355.62

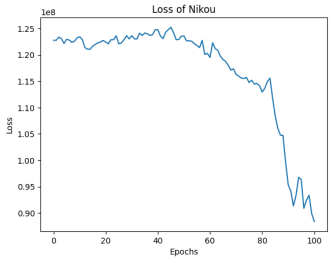

Figure 4. Loss of Nikou Gunnemann'

6. Conclusion

In this investigation, Bhansen and Nikou Gunnemann's strategies were put to the ultimate test under strict scrutiny. The research findings, which are very detailed, confirm that such teaching approaches are the best method of learning when it comes to the discovery of the outcomes of cost-sensitive regressions.

References

[1] Alejandro Correa Bahnsen "Cost Sensitive Credit Card Fraud Detection using Bayes Minimum Risk " , 12th International Conference on Machine Learning and Applications (2013).

[2] Nikou G unnemann(B) and J urgen Pfeffer "Cost Matters: A New Example-Dependent Cost-Sensitive Logistic Regression Model " , April 2017.