Feature Corruption as Oversampling Technique for Imbalanced Data

Yevgeni Berkovitch

March, 2023

Abstract

In this project we propose the method of feature corruption to oversample the minority class instances in imbalanced datasets. The investigated feature corruptions include Zero-Out Noise, Mean Value Imputation, Gaussian Noise and Feature Permutation. The method's performance is compared to other oversampling techniques on a standard benchmark of 13 imbalanced datasets. The results show the feasibility of feature corruption as re-balancing technique - Gaussian Noise corruption achieving the best ranking.

1 Problem Description

Imbalanced data is ubiquitous in classification tasks. It poses a serious problem, since many algorithms are sensitive to the target distribution in data, resulting in poor performance. Many methods exist to tackle the imbalance problem, such as majority downsampling, minority oversampling and hybrid methods.

Here we propose a simple oversampling method, based on feature corruption, which can be easily incorporated into the machine learning pipeline pipeline and serve as a good starting point for classification tasks on imbalanced data.

2 Solution

Inspiration for this project is the work on Marginalized Corrupted Features [1] which was used to increase the number of training points irrespective of their association with majority or minority classes.
In our work we build on this idea to oversample the minority class instances and balance the dataset.

This is done in 2 steps.
(1) The instances of the minority class are duplicated to equalize their number with the majority class instances.
(2) All features of the minority class are corrupted (modified) with probability ***p*** or left unchanged with probability ***1-p***.

For each corruption method we perform the experiment with 3 sets of probabilities: **0.1**, **0.25** and **0.5**.

The explored corruption methods are as follows:

- Setting the feature value to 0 ("*Zero-Out Noise*")
- Setting the feature value to mean value of the feature in the population ("*Mean Value Imputation*")
- Assigning a random value from the Gaussian distribution parameterized by the mean and standard deviation of the corresponding feature ("*Gaussian Noise*")
- Sampling the random value from the underlying distribution of the feature. This is done by performing permutation of the feature values ("*Feature Permutation*")

Feature Corruption is applied only to the train set, while the test set is left in its original unbalanced state.

3 Experimental Evaluation

3.1 Datasets

In our experiments we used the datasets which serve as a standard benchmark [2] for working with imbalanced data, and have been incorporated in python's *imblearn* package API. The mentioned collection comprises 27 datasets with imbalance ratio varying from 8.6:1 to 130:1.

Out of 27 datasets we used 13, which produced the lowest *G-Means scores* when the imbalance was left untreated. The rationale behind selecting these datasets was that from machine learning practiotioner's view - they are in bigger need of re-balancing or any other kind of intervention to improve the classification outcome.

3.2 Evaluation

In order to evaluate our solution we compared the performance of the proposed feature corruption methods on 13 datasets with 4 other methods: (1) no balancing (2) random majority downsampling (3) random minority oversampling (4) SMOTE

The experiments were run in 5-fold cross-validation manner 10 times (each time with a different random seed to ensure a different split of data between folds), and the results were averaged.

The classification outcomes were evaluated with 2 metric scores that are often used for imbalanced data: **G-Means** and **F1-Score**.

Since these metrics are sensitive to the choice of decision threshold - in the end of each training episode an iterative search for the optimal threshold was performed and the score corresponding to that threshold was taken into account.

For each method the training and prediction were done with untuned Random Forest classifier (from sklearn package).

In order to compare the relative performance of the used methods across all 13 datasets - instead of averaging the metrics, we averaged the relative rankings of each method, as proposed by Kovacs [3] (the best method for a given dataset was awarded the rank 1 and the worst - 8).

3.3 Results

Below are the tables summarizing the **G-Means** and [**?**] as well as the rankings for these metrics for all methods across 13 datasets. More detailed results for each method can be found in the project's Git repository.

Dataset	Base	MajDwn	MinOver	SMOTE	Zero	MeanF	Gauss	Perm
ecoli	0.8795	0.8825	0.8818	0.8918	0.8637	0.8887	0.8898	0.8813
satimage	0.8795	0.8825	0.8818	0.8918	0.8637	0.8887	0.8898	0.8813
abalone	0.7876	0.7901	0.7816	0.7833	0.7812	0.7820	0.7806	0.7851
us_crime	0.8581	0.8582	0.8533	0.8515	0.8573	0.8565	0.8616	0.8605
yeast_ml8	0.5574	0.5738	0.5764	0.5668	0.5866	0.5826	0.5740	0.5759
scene	0.6914	0.7268	0.7350	0.7290	0.6679	0.7350	0.7353	0.7325
coil_2000	0.6530	0.6659	0.6591	0.6420	0.6245	0.6317	0.6579	0.6443
solar_flare_m0	0.7257	0.7193	0.7118	0.7343	0.7192	0.7180	0.7139	0.7112
oil	0.8448	0.8287	0.8728	0.8858	0.8737	0.8660	0.8744	0.8750
wine_quality	0.8137	0.7841	0.8108	0.8048	0.8099	0.8042	0.8211	0.8186
yeast_me2	0.8473	0.8403	0.8531	0.8570	0.8537	0.8653	0.8650	0.8573
ozone_level	0.8183	0.8177	0.8153	0.8180	0.8189	0.8120	0.8294	0.8236
abalone_19	0.6721	0.7330	0.6666	0.7314	0.6807	0.6960	0.7156	0.7196

Table 1: G-Means Scores (averaged per10 runs)

Dataset	Base	MajDwn	MinOver	SMOTE	Zero	MeanF	Gauss	Perm
ecoli	7	4	5	1	8	3	2	6
satimage	6	8	2	1	7	3	5	4
abalone	2	1	6	4	7	5	8	3
us_crime	4	3	7	8	5	6	1	2
yeast_ml8	8	6	3	7	1	2	5	4
scene	7	6	3	5	8	2	1	4
coil_2000	4	1	2	6	8	7	3	5
solar_flare_m0	2	3	7	1	4	5	6	8
oil	7	8	5	1	4	6	3	2
wine_quality	3	8	4	6	5	7	1	2
yeast_me2	7	8	6	4	5	1	2	3
ozone_level	4	6	7	5	3	8	1	2
abalone_19	7	1	8	2	6	5	4	3
AVERAGE	5.23	4.85	5.00	3.92	5.46	4.62	3.23	3.69

Table 2: G-Means Rankings

Among 4 corruption methods in our experiment, the best performance was achieved by adding Gaussian Noise to oversampled minority instances with the probability of 10%.

Dataset	Base	MajDwn	MinOver	SMOTE	Zero	MeanF	Gauss	Perm
ecoli	0.6666	0.6128	0.6212	0.6531	0.6067	0.6178	0.6402	0.6191
satimage	0.6932	0.6478	0.6920	0.6914	0.6810	0.6930	0.6786	0.6803
abalone	0.3996	0.4037	0.4013	0.3915	0.3964	0.3925	0.3955	0.3943
us_crime	0.5490	0.5247	0.5249	0.5193	0.5460	0.5302	0.5392	0.5357
yeast_ml8	0.1608	0.1726	0.1705	0.1640	0.1743	0.1725	0.1683	0.1688
scene	0.3015	0.3040	0.3178	0.3142	0.2740	0.3192	0.3269	0.3296
coil_2000	0.2079	0.2196	0.2139	0.2012	0.1754	0.2004	0.2255	0.2053
solar_flare_m0	0.2542	0.2368	0.2111	0.2454	0.2363	0.2532	0.2462	0.2360
oil	0.5733	0.4647	0.6037	0.6232	0.5865	0.5954	0.6119	0.5978
wine_quality	0.4520	0.3321	0.4264	0.3926	0.4469	0.4081	0.4424	0.4346
yeast_me2	0.4322	0.3715	0.4592	0.4193	0.4120	0.4124	0.4389	0.4060
ozone_level	0.3703	0.3381	0.3696	0.3618	0.3788	0.3943	0.4135	0.4007
abalone_19	0.0767	0.0588	0.0894	0.0832	0.0705	0.0670	0.0608	0.0649

Table 3: F1 Scores (averaged per10 runs)

Dataset	Base	MajDwn	MinOver	SMOTE	Zero	MeanF	Gauss	Perm
ecoli	1	7	4	2	8	6	3	5
satimage	1	8	3	4	5	2	7	6
abalone	3	1	2	8	4	7	5	6
us_crime	1	7	6	8	2	5	3	4
yeast_ml8	8	2	4	7	1	3	6	5
scene	7	6	4	5	8	3	2	1
coil_2000	4	2	3	6	8	7	1	5
solar_flare_m0	1	5	8	4	6	2	3	7
oil	7	8	3	1	6	5	2	4
wine_quality	1	8	5	7	2	6	3	4
yeast_me2	3	8	1	4	6	5	2	7
ozone_level	5	8	6	7	4	3	1	2
abalone_19	3	8	1	2	4	5	7	6
AVERAGE	3.46	6.00	3.85	5.00	4.92	4.54	3.46	4.77

Table 4: F1-Score Rankings

It's worth noting that when measured by F1-score none of the rebalancing strategies is able to improve the classification result compared to the unbalanced version.

4 Related Work

The balancing strategies belong to 3 main classes [4]: (1) downsampling the majority class (2) oversampling the minority class and (3) hybrid sampling strategies.

Among minority oversampling techniques the most popular are SMOTE [5], AdaSyn [6] and CTGAN [7]. There are also multiple variations of these methods - only for SMOTE there are few dozens of known modifications.

It seems, however, that the need for new methods is not saturated since there is no single method that works universally well for all imbalanced datasets. Some authors even insist that imbalanced data should not be treated at all - instead the problem should be solved at the algorithm level or with the proper selection of thresholds.

The idea for this project comes from the research which used Marginalized Corrupted Features [1] to increase the amount of training data (regardless of its class).

5 Conclusions and Future Work

Minority oversampling via feature corruption proves to be a valid technique and can be recommended for treating the imbalanced data.

However, the choice of the appropriate resampling strategy should be done individually for each dataset and compared to other methods as well as the baseline (untreated data).

There are also some ways to modify the feature corruption strategy and potentially get better results.

For example, it is possible to create few corrupted versions of the test instances and average the probabilities of the predicted classes (similar to the test-time augmentation technique often employed in computer vision).

Also different corruption methods can be performed on the features with continuous vs. discrete values.

References

[1] Laurens van der Maaten et al., *Learning with Marginalized Corrupted Features*, (ICML'13: Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28 June 2013).

[2] Lemaitre, Guillaume; Nogueira, Fernando; Aridas, Christos K.; Oliveira, Dayvid V. R., *Imbalanced dataset for benchmarking*, 2016

[3] Gyorgy Kovacs, *An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets*, 2019

[4] Susan S, Kumar A, *The balancing trick: optimized sampling of imbalanced datasets—a brief survey of the recent State of the Art*, Eng Rep 3(4):1–24, 2020

[5] Chawla N.V. et al., *SMOTE: Synthetic Minority Over-sampling Technique*, Journal of Artificial Intelligence Research 16: 321–357, 2002

[6] He H. et al., *ADASYN: Adaptive synthetic sampling approach for imbalanced learning*, IEEE World Congress on Computational Intelligence), Hong Kong, pp. 1322-1328, 2008

[7] Xu, Lei et al., *Modeling Tabular data using Conditional GAN*, 2019