Fetal acidosis detection

Aleksandr Barinov

***Abstract*— early detection of fetal acidosis during labor is a problem approached in this article. One possible solution is a supervised machine learning classification model that predicts whether fetus suffers from acidosis. Several models were described, tuned and tested for the mentioned purpose. As a side effect troubles with learning on an imbalanced dataset, that are common in medical diagnosis, were encountered and elaborated on the way to the main goal. As a result of experiments it was found that cost-sensitive learning conducted on a logistic regression model is best suited for the purpose of fetal acidosis detection. Codes that produced the results published in this article are a part of submission.**

I. Assignment

The main goal of the assignment is to find the best machine learning classification model for fetal acidosis detection. The best here means the model that will achieve higher g_mean score that is defined farther in the article. Model is to be trained and tested on a dataset that includes features computed from fetal heart rate recordings labeled as 1 in case of acidosis and 0 in a normal case.

II. Introduction

Early detection of fetal acidosis during labor is not crucial according to [1]. Quote from the mentioned article: "In the past much attention has been paid to acute acidosis during labour, but in previously normal fetuses this is rarely associated with subsequent damage". Nevertheless, even the subsequent damage is rare the probability of it is not zero. Therefore it make sense to develop better ways to correctly predict if fetal suffers from acidosis.

One way to detect fetal acidosis is by monitoring fetal heart rate during labour. The dataset of features computed from fetal heart rate recordings was provided for the purpose of supervised model training.

As in many medical diagnostic problems there are much more negative cases than positive. That leads to the problem of learning on an imbalanced dataset. This problem could be encountered in several ways such as data-level preprocessing, cost-sensitive learning, ensemble learning that are to be elaborated farther in the article.

The main contribution of this article lies in:

- describing existing methods of dealing with learning on an imbalanced dataset

- evaluation of supervised machine learning models for the purpose of fetal acidosis detection.

III. Methodology

Notation: data set is denoted as $T = \{(x^{(i)}, y^{(i)})\}$, where $y^{(i)} \in \{0, 1\}$ and $x^{(i)}$ is a vector of features.

Three types of classification model were trained and evaluated:

Logistic regression

Logistic regression is a linear model for classification.

Presented formulas correspond to [2].

Prediction: $f_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{x}\mathbf{w}^T) = 1/(1 + e^{-\mathbf{x}\mathbf{w}^T})$ where $\mathbf{w}$ are weights that we intent to learn and $g(z) = 1/(1 + e^{-z})$ is the logistic (sigmoid) function.

Cost function: $J(\mathbf{w}, T) = 1/|T| \sum_{i=1}^{|T|}(-y^{(i)} \cdot log(\hat{y}^{(i)}) - (1 - y^{(i)}) \cdot log(1 - \hat{y}^{(i)}))$ where $\hat{y}^{(i)} = f_{\mathbf{w}}(\mathbf{x}^{(i)})$

Model learning: $\mathbf{w}^* = argmin_{\mathbf{w}} J(\mathbf{w}, T)$

Random forests

Random forests algorithm could be simply explained as follows:

first step - create a bootstrapped dataset from available data by randomly choosing samples;

second step - build a decision tree on a bootstrapped dataset considering only a subset of features;

repeat step 1 and step 2 several times;

at the end of the described loop we get a variety of decision trees that are used for making predictions by using voting for classification tasks.

Bagging classifier

A bagging classifier is one of ensemble methods that combine the predictions of several base estimators. It fits each base classifier on a random subset of the original dataset and produce the final prediction by voting.

IV. Experiments

Two main libraries used for models implementation and testing are scikit-learn and imbalanced-learn. Most of the mention classes/functions farther in the text could be found in these two libraries if not specified otherwise. Imbalanced-learn library could be easily installed with the following command: "pip install -U imbalanced-learn".

Each implemented model described in the article was evaluated by 5-fold cross validation. Custom cross validation function was implemented in order to be able to preprocess train data subset in each iteration and also compute confusion matrix for test data subsets for all iterations. Confusion matrices presented in the article were computed as a sum of confusion matrices produced in each iteration of cross validation on test subsets. Final score of each model (so called g_mean score) was computed as average over all cross validation tests of values returned by custom g_mean_score$(y, \hat{y})$ function: $g = \sqrt{se \cdot sp}$ where $se = TP/(TP + FN)$ is sensitivity and $sp = TN/(TN + FP)$ is specificity.

Dataset

Provided for training dataset consists of features computed from a fetal heart rate recordings and corresponding labels.

The dataset is highly imbalanced i.e. there is less than 3% of positive values. That is a challenging factor common for medical diagnosis.

Input **x**:

Vector **x** of automated FIGO-like features, spectral features, scale-free dynamics features.

Ground truth:

The data are divided into two classes: acidotic with $pH \leq 7.05$ ($y = 1$) and normal with $pH > 7.05$ ($y = 0$).

Some approaches to deal with an imbalanced dataset:

Data-level preprocessing

The naive way is to simply copy the minority data several times to balance the dataset. Even with this naive approach at a first stage of model development it was possible to get much better predictions. But soon enough imbalanced-learn library was found by the author and over/under sampling methods such as SMOTE and RandomUnderSampler were tested. So the results for the naive approach are not included.

Several methods from imbalanced-learn library were used to balance the dataset. The model these methods were tested on is a RandomForestClassifier(n_estimators=8, max_depth=4).

Briefly explained the RandomOverSampler is randomly duplicating original samples of the minority class, SMOTE generate new samples by interpolation, RandomUnderSampler is randomly selecting a subset of data of the majority class, ClusterCentroids makes use of K-means to reduce the number of samples in the majority class.

According to the Table.1 random forests perform best on a dataset balanced with SMOTE and RandomUnderSampler methods. The difference between achieved g_mean score is negligible. To farther elaborate which of these two methods is best lets take a look at Fig.1 and Fig.2. It can be seen that despite similar g_mean score sensitivity achieved with RandomUnderSampler is 0.61 while for SMOTE it's only 0.48. Because higher sensitivity is preferred in diagnostic problems (i.e it is more important to find all sick people than to not bother healthy) RandomUnderSampler could be considered a better technique for balancing the dataset for random forests model.

Cost-sensitive learning

The basis of this method is to apply weights on the minority and majority classes costs in a cost function. In logistic regression that was chosen for this approach modified cost function looks as follows: $J(\mathbf{w}, T) = 1/|T| \sum_{i=1}^{|T|}(-z_1 \cdot y^{(i)} \cdot log(\hat{y}^{(i)}) - z_0 \cdot (1 - y^{(i)}) \cdot log(1 - \hat{y}^{(i)}))$ where z_1 and z_0 are weights applied on minority and majority classes respectively. $z_j = |T|/(2 \cdot n_j)$ where 2 is a number of classes and n_j is a number of samples in a class j.

TABLE I

OVER/UNDER SAMPLING METHODS COMPARISON FOR A RANDOM FORESTS MODEL

Over/under sampling method	cross validation g_mean score
RandomOverSampler	0.518
SMOTE	0.606
RandomUnderSampler	0.61
ClusterCentroids	0.34

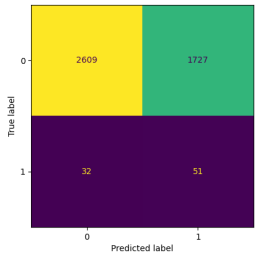

Fig. 1. confusion matrix computed over cross validation performed on random forests model with dataset balanced by RandomUnderSampler method

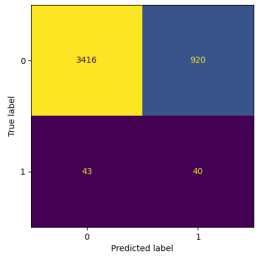

Fig. 2. confusion matrix computed over cross validation performed on random forests model with dataset balanced by SMOTE method

Ensemble learning

Ensemble methods combine the predictions of several base estimators built with a given learning algorithm in order to improve robustness over a single estimator.

To try out this approach two types of ensemble models were tested: random forests that combine predictions of decision trees and bagging classifier with logistic regression model as a basic estimator.

Models evaluation

All the models described farther in the text were part of a pipe=Pipeline(steps=[("sc", sc),("vt", vt),("clf", clf)]), where sc = StandardScaler(), vt = VarianceThreshold(threshold=$(0.9 \cdot (1 - 0.9))$) and clf is a classifier model to be specified.

Logistic regression classifier (Cost-sensitive learning)

For implementing cost-sensitive learning clf=LogisticRegression(class_weight="balanced", max_iter=200) was used initially.

Grid search:

In the Table.2 some results of GridSearchCV(pipe, parameters, scoring=scorer), where: parameters are in the first column; scorer = make_scorer(g_mean_score, greater_is_better=True). In the column "g_mean score" results of the custom cross validation function mentioned earlier in the article are presented.

Manual tuning:

While according to grid search lbfgs solver with parameter C=0.3 is best, better g_mean score = 0.6706 was achieved with C=1.

Final model after manual tuning: clf = LogisticRegres-

sion(class_weight=”balanced”, max_iter=200, solver=’lbfgs’, C=1)

Figure Fig.3 represents confusion matrix computed for the final model. Sensitivity is 0.613 specificity is 0.753 and mean g_mean score over all cross validation test subsets is 0.67.

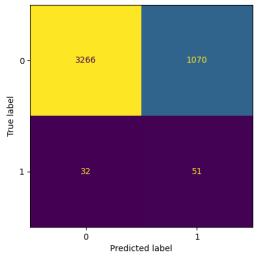

Fig. 3. confusion matrix computed over cross validation on test subsets performed on a logistic regression model

Random forests classifier (Data-level preprocessing + ensemble learning)

Grid search:

By grid search through parameters: ’n_estimators’:range(4,12,2), ’max_depth’:range(2,8,2) best parameters found were: ’max_depth’: 6, ’n_estimators’: 8 that at best produced mean g_mean score over all cross validation test subsets equal to 0.65.

Manual tuning:

By additional manual tuning any better results were not achieved. When increasing either max_depth or n_estimators or both parameters model starts to suffer from overfitting. If same parameters are decreased then model does not seem to be able to learn well enough.

Final model is clf = RandomForestClassifier(n_estimators=8, max_depth=6)

Figure Fig.4 represents confusion matrix computed for the final model. Sensitivity is 0.686 specificity is 0.632 and mean g_mean score over all cross validation test subsets is 0.65.

Bagging classifier (Data-level preprocessing + Cost-sensitive learning + ensemble learning)

Grid search:

By grid search through parameters: ’n_estimators’:range(2,8,2) best parameters found were: ’n_estimators’: 6 that at best produced mean g_mean score over all cross validation test subsets equal to 0.675.

TABLE II

LINEAR REGRESSION PARAMETERS TUNING VIA GRIDSEARCHCV(..)

Parameters scope	Best found	test g_mean score
’solver’:[’newton-cg’, ’lbfgs’, ’liblinear’, ’sag’], ’C’:[0.1,1,10]	’C’: 0.1, ’solver’: ’sag’	0.632
’solver’:[’newton-cg’, ’lbfgs’, ’liblinear’], ’C’:[0.1,0.5,1]	’C’: 0.5, ’solver’: ’newton-cg’	0.657
’solver’:[’lbfgs’, ’liblinear’], ’C’:[0.1,0.3,0.5,0.7]	’C’: 0.3, ’solver’: ’lbfgs’	0.658

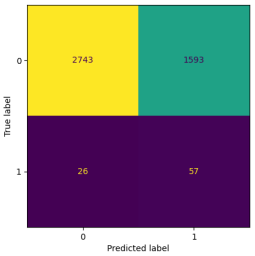

Fig. 4. confusion matrix computed over cross validation on test subsets performed on a random forests model

Manual tuning:

By additional manual tuning any better results were not achieved.

Final model is clf = BaggingClassifier(base_estimator=LogisticRegression(class_weight=”balanced”, max_iter=200, solver=’lbfgs’, C=1), n_estimators=6)

Figure Fig.5 represents confusion matrix computed for the final model. Sensitivity is 0.613 specificity is 0.76 and mean g_mean score over all cross validation test subsets is 0.675.

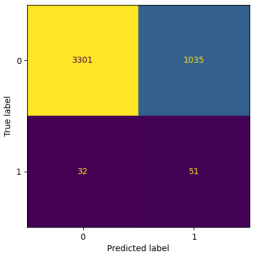

Fig. 5. confusion matrix computed over cross validation on test subsets performed on a bagging classifier model

Final comparison:

Tuned models were finally tested on a bigger chunk of the original dataset in order to understand which model is more robust. All previous tests were run on one segment of a dataset. Fig.6, Fig.7 and Fig.8 contain confusion matrices computed over cross validation on 5 segments of provided dataset. Table.4 contains g_mean score achieved by each classifier.

TABLE III

BEST RESULTS OVER CROSS VALIDATION ON 1 SEGMENT OF ORIGINAL DATASET

Classifier	cross validation mean train g_mean score	cross validation mean test g_mean score	sensitivity
Bagging classifier	0.731	0.6743	0.613
Random forests classifier	0.78	0.6501	0.686
Logistic regression	0.728	0.6705	0.613

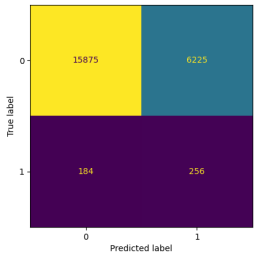

Fig. 6. confusion matrix computed over cross validation on test subsets of 5 segments of original dataset performed on a bagging classifier model

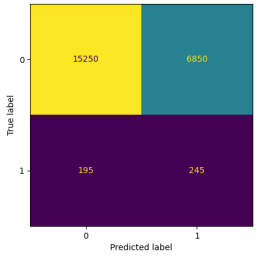

Fig. 7. confusion matrix computed over cross validation on test subsets of 5 segments of original dataset performed on a random forests classifier model

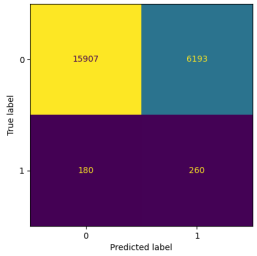

Fig. 8. confusion matrix computed over cross validation on test subsets of 5 segments of original dataset performed on a logistic regression model

V. Discussion

According to the Table.3 none of the tested models suffers from overfitting.

g_mean score achieved on one segment of dataset is very close for all 3 tested models according to the Table.3. At the same time best sensitivity, that is the preferred metric in diagnostic problems, is higher for random forests. But Table.3 presents best achieved results over running cross validation several times. While logistic regression model proved to be deterministic and bagging classifier quite stable, results of random forests classifier varied with each new run of cross validation. Fig.9 represents close to worst results for random forests classifier with specificity equal to 0.578 and g_mean score equal to 0.596. Due to it's random nature random forests classifier could be considered unreliable when it is crucial to deliver consistently good results.

TABLE IV
Results over cross validation on 5 segments of original dataset

Classifier	cross validation mean test g_mean score
Bagging classifier	0.6402
Random forests classifier	0.6121
Logistic regression	0.6457

For a final choice of a model it is necessary to understand how robust each model is. To achieve this goal all three tuned models were tested on 5 segments of original dataset as described in the previous section. From the results in Table.4 we can see that logistic regression proved to be more robust than random forests classifier. Also bagging classifier that combined the results of 6 logistic regressors did not improve g_mean score of logistic regression model as was intended. The idea behind the choise of bagging classifier with logistic regressor as basic estimator was that by learning individually well performing logistic regressors on random subsets some percent of errors that was caused by randomness of outliers in a training dataset, but not by theoretical inability of logistic regression model to distinguish between classes will be cancelled out by several regressors. The idea was proven wrong, at least on the provided dataset with chosen parameters of tested models.

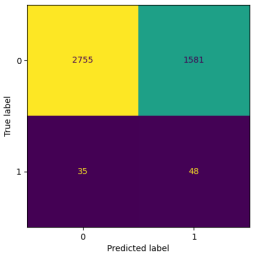

Fig. 9. confusion matrix computed over cross validation on test subsets performed on a random forests model (close to worst case)

VI. Conclusion

From the results presented and discussed above it can be concluded that logistic regression model is the most robust not suffering from overfitting model reliably achieving high g_mean score and good enough sensitivity compare to other tested models.

One of the conclusions that could be made based on the conducted experiments is that in case of learning on an imbalanced dataset it is highly important to either choose model that could learn well on such dataset (such as logistic regression model with cost-sensitive learning) or to preprocess dataset and to test different methods of preprocessing on the model that is to be used for predictions, i.e. choise of model and data-level preprocessing could be of equal importance when dealing with imbalanced dataset.

As was shown on a random forests classifier with randomized algorithms it is not advisable to rely on one run of cross validation. Due to randomization with each run of cross

validation results could differ significantly and variety of these results should be considered.

Another less important conclusion is that combination of several estimators learned on random subsets of preprocessed data as was done with a use of bagging classifier and logistic regression model as a basic classifier will not necessarily lead to better results than achieved by one independent basic classifier.

REFERENCES

[1] Catherine S Bobrow, Peter W Soothill: *Causes and consequences of fetal acidosis*, University of Bristol, Department of Obstetrics, St Michael's Hospital, Bristol.

[2] Author Petr Posik: *Learning.Linear Methods for Regression and Classification.*, Handouts. Czech Technical University in PragueFaculty of Electrical EngineeringDept. of Cybernetics.