B(E)3M33UI — Semester project 1: Fetal acidosis detection

Jiří Spilka

April 2, 2019

1 Introduction

Can you help obstetricians to early detect fetal acidosis during labor?

Fetal Heart Rate (FHR) monitoring is routinely used in clinical practice to help obstetricians assess fetal health status during delivery. However, early detection of fetal acidosis that allows relevant decisions for operative delivery remains a challenging task. A comprehensive set of features is computed for FHR description on a large and well documented database.

The goal is to create a machine learning model for fetal heart rate classification that is able to predict whether fetuses suffer from acidosis.

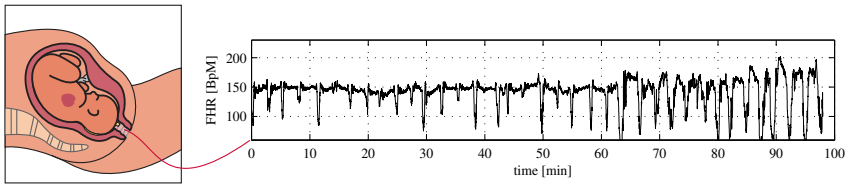

Figure 1: Recording of fetal heart rate during labor

2 Requirements

You have to submit the python scripts, which demonstrate what you have achieved, and a report describing the methods and results

1. module `fetal_acidosis_classification.py` (mandatory), containing a working classifier, which will be used to assess the quality of your solution,
2. any other Python scripts that are needed to reproduce the results, and
3. PDF file containing the report (mandatory), and
4. (optionally) other Python scripts, data, etc. used to create the report, especially the code that generates the results and figures used in the report, e.g. when comparing several classifiers. The code must be organized such that it is clear what parts of the code are related to what parts of the report. These files will not be evaluated, they serve as a reference: if any part of the report would raise any doubts, these files will be consulted. If they are missing, the questionable part of the report may be evaluated with 0 points.

Remember to submit your own work! Do not commit plagiarism! The plagiarism detection feature will be on for this task, and any confirmed plagiat may be a reason to assign 0 points from this task! This applies to both the PDF reports and python scripts.

2.1 Code

Your code should be clean, readable, and understandable. This will form one criterion of the evaluation. Please, take the time to come up with meaningful names of variables, functions, constants, etc. Keep the functions short, spanning several lines only, if possible. Make appropriate comments.

2.2 Report

The report can be written in Czech or in English. It should have a form of a scientific article. You can assume the reader has a general background in machine learning but does not know the individual methods. The report should contain an adequate description of the data, principles, methods used and results achieved in your work. By adequate description we mean a **concise description** sufficient to **understand and replicate** the work done by you. You should not only present the results but also try to **interpret and discuss** them. You can use the provided templates in LaTeX or MS Word.

3 Classifier: mandatory part

Create a fetal acidosis detection algorithm, describe it, and submit it for the quality evaluation.

3.1 Specifications

In module `fetal_acidosis_classification.py`, implement functions `train_model()` and `predict()`. Do not change arguments of these functions, as they will be imported for quality evaluation.

1. `train_model(X,y)`, which takes the training data (features) X, and labels (classes pH $\leq$ 7.05 or pH $>$ 7.05) y, trains a classifier (or a pipeline with preprocessors and a classifier), and returns the trained classifier (or rather the whole pipeline).
2. `predict(model,X)` takes the model (classifier, pipeline) and the features X, and returns the predictions for this data.

3.2 Working example

In the module `fetal_acidosis_classification.py`, you can see an example of a dummy classifier. It uses the `DummyClassifier`.

4 Data

The FHR recordings and computed features are available and described in detail below. It can be used for: algorithm selection, hyper-parameter tuning, and performance estimation. Use them as you wish. In addition, the quality evaluation will be done on different test data.

5 Evaluation

The classifiers are evaluated using a geometric mean of sensitivity (SE) and specificity (SP)

$$gmean = \sqrt{SE \cdot SP}, \tag{1}$$

where $SE = TP/(TP + FN)$ and $SP = TN/(TN + FP)$. It is already implemented in the supplied module as the function `g_mean_score`. It uses confusion matrix to compute TP, FP, TN, FN.

6 Model tuning

Use a grid search feature to search for optimal settings of each model. When you construct a pipeline and need to tune some parameters of transformations and model inside that pipeline, you can "address" the parameters hidden in the pipeline as `<estimator>__<parameter>`. See the documentation of pipeline with the example of `GridSearch`.

When using the grid search facility, you may also need some other information about how to construct your own scoring function and use it e.g. in grid search.

7 Comparison of several models

Compare several types of models using ROC curves, *gmean*, AUC and/or, learning curves .... You can then choose the best model as the final one and submit it. You can use any classifiers, e.g.

- k nearest neighbors,
- logistic regression,
- SVM with different kernels (linear, polynomial, RBF),
- naive Bayes with multinomial distribution,
- decision trees, adaboost, random forests,
- neural networks, ...

8 Additional ideas

To gain additional points, you can try to elaborate on the following ideas:

- Add more features (e.g. use packages tsfresh, kymatio, or ask Jiri Spilka for additional features).
- Incorporate information about different labour stages (first stage, second stage – see below for details)
- Create a more robust model using data from more FHR segments (see below for details).
- Use also uterine contractions signal together with FHR (the link to signals is in additional materials).
- Interpret the classification model, what features are important or on which examples the model makes mistake.
- Select only relevant features using (feature selection or L_1 regularization).

9 Scoring

The final score for the task will be composed of the following components:

Component	**Regular points**
Code quality (readability, understand-ability)	0-2
Report in LaTeX	0-1
Adequate description of the final model chosen for competition	0-2
Model tuning via grid search + adequate description in the report	0-2 per classifier type, max. 5
Model comparison + adequate description in the report	0-4
Results and model interpretation, discussion	0-3
Evaluation on hidden test data	0-3
Component	**Bonus points**
Additional (non-trivial) ideas tried + description in the report	0-1 per idea, max. 2
Model contest	0-3
Total	Max. 20 regular points + max. 5 bonus points

The necessary condition for the assessment is to get at least 10 regular points from this project. It is recommended to try to improve score in several ways and describe them adequately in the report. This way, you will

1. gain experience in model building and build a better model, which will score higher in the during quality assessment and in the contests (thus bringing you more points), and
2. have the chance to get additional points for extra effort.

9.1 Quality scoring

The *gmean* will be used to measure the quality of your model. The score will be translated to points using the following table:

gmean	Points
$gmean < 0.50$	0
$0.50 \leq gmean < 0.60$	1
$0.60 \leq gmean < 0.70$	2
$0.70 \leq gmean$	3

9.2 Contests scoring

As a bonus, your model will enter a contest. Based on the results on the test dataset, the model will be ranked from the best to the worst.

Quartile	Points
1st	3
2nd	2
3rd	1
4th	0

10 Additional materials

10.1 Database

The intrapartum CTG database consists in total of 4546 intrapartum recordings, which were acquired between April 2010 and September 2017 at the obstetrics ward of the University Hospital in Brno, Czech Republic. The data also contains clinical information, notably providing pH after delivery that is used to define fetal acidosis. Subject inclusion criteria are singleton, gestational age $\geq$ 37 (weeks), maternal age $\geq$ 18. Further, data quality requirements are signal quality (length $\geq$ 60 min, amount of missing data $\leq$ 15%, the time between the end of a recording and birth $\leq$ 10 min). The data were further divided into two classes: *acidotic*, $N_+ = 90$, with pH $\leq$ 7.05 and *normal*, $N_- = 4456$, with pH $>$ 7.05. The dataset of fetal heart rate records is available at CTU_UHB_2017.zip.

10.2 Feature computation

The features were computed using a sliding window of 20 minutes with 5 minutes overlap. The code for feature computation is available at CTG-database-2017. There are several additional attributes: `segStage` (values 1, 2, or 12) marks stage of labor, `segIndex` (values from 1 to number of segments) marks segment index from end (the last segments equals to 1), `segStageI_index` (values from 1 to number of segments in the first stage) marks segments of first labor stage (the last segment of the first stage equals to 1), `segStageII_index` (values from 1 to number of segments in the second stage) marks segments of second labor stage (the last segment of the first stage equals to 1),

10.2.1 Automated FIGO-like features (9).

Rather than trying to implement the precise FIGO criteria, 9 FIGO-inspired features, are used, $(\beta_0, \beta_1, LTV, STV, \#acc, \#dec, MAD_{dtrd}, T_{stress}, A_{dec})$, quantifying: baseline $B(t)$ level and evolution as $B(t) = \beta_0 + \beta_1 t$; long and short term variabilities (LTV, STV), accelerations/decelerations via their numbers (#*acc* and #*dec*), average depth MAD_{dtrd}, average duration T_{stress} and average area A_{dec}.

10.2.2 Spectral features (5).

Spectral estimation is conducted over BpM time series using the standard Welch periodogram procedure. The following frequency bands were used: E_{VLF} ($[0.003, 0.04]$ Hz), low frequency E_{LF} ($[0.04, 0.15]$ Hz), and high frequency E_{HF} ($[0.04, 0.15]$ Hz). Further, the ratio LF/HF of E_{LF} and E_{HF}, and the spectral index α, estimated over both LF and HF bands are computed.

10.2.3 Scale-free dynamics features (6).

Scale-free dynamics and multifractal features were recently shown to offer relevant and robust alternatives to the classical measurements of long and short term variabilities as quantified by STV and LTV. The Hurst parameter H is a linear feature, as it describes both the autocorrelation function or the Fourier spectrum, yet in a scale-free spirit. It has been shown that it can be efficiently and robustly computed from discrete wavelet transforms applied to FHR time series. The multifractal parameters $(h_{min}, c_1, c_2, c_3, c_4)$ produce advanced scale-free characterization of the variability.

11 Good luck and have fun!