Classification of Morphologic Pattern and Fetal State with Cardiotocography Exam

Author: Giuseppe Laruina

Computer and Automation Engineering: Robotics and Automation

Abstract—

Cardiotocography (CTG) is a technical means of recording the fetal heartbeat and the uterine contractions during pregnancy. Main objectives of cardiotocography are to investigate fetal well-being through the study of heartbeat and detect the presence or absence of uterine contractions and their frequency. The dataset consists of measurements of fetal heart rate (FHR) and uterine contraction (UC) features on cardiotocograms classified by expert obstetricians. 2126 fetal cardiotocograms (CTGs) were automatically processed and the respective diagnostic features measured. The CTGs were also classified by three expert obstetricians and a consensus classification label assigned to each of them. Classification was both with respect to a morphologic pattern (A, B, C, ...) and to a fetal state (N, S, P). Therefore the dataset can be used either for 10-class or 3-class experiments.

Source:
Marques de SÃ¡, J.P., jpmdesa'@'gmail.com, Biomedical Engineering Institute, Porto, Portugal.
Bernardes, J., joaobern'@'med.up.pt, Faculty of Medicine, University of Porto, Portugal.
Ayres de Campos, D., sisporto'@'med.up.pt, Faculty of Medicine, University of Porto, Portugal.

1 Dataset Informations

The dataset contains 2126 examples of phetus cardiotocograms. Every element of the dataset is made of 21 features in which: 'LB' is the FHR (Fetal Heart Rate) baseline (beats per minute), 'AC' is the number of acceleration per second, 'FM' the number of fetal movements per second, 'UC' number of uterine contractions per second , 'DL' number of light decelerations per second, 'DS' number of severe decelerations per second, 'DP' number of prolongued decelerations per second, 'ASTV' is the percentage of time with abnormal short term variability, 'MSTV' the mean value of short term variability, 'ALTV' the percentage of time with abnormal long term variability, 'MLTV' the mean value of long term variability, 'Width' is the width of FHR histogram, 'Min' the minimum of FHR histogram, 'Max' the maximum of FHR histogram , 'Nmax' is the number of histogram peaks, 'Nzeros' the number of histogram zeros, 'Mode' the histogram mode, 'Mean' the histogram mean, 'Median' the histogram median, 'Variance' the histogram variance and finally 'Tendency' is the histogram tendency.
Every example has two associated target: one that indicates the Morphologic Patter and one that indicates Fetal Health Status. The latter assumes 10 values that can be: 'A' is the calm sleep, 'B' the REM sleep, 'C' calm vigilance, 'D' active vigilance, 'SH' shift pattern (A or Susp with shifts), 'AD' accelerative/decelerative pattern (stress situation), 'DE' decelerative pattern (vagal stimulation), 'LD' largely decelerative pattern, 'FS' flat-sinusoidal pattern (pathological state) and 'SUSP' is the suspect pattern. The Fethal Healt Status can assume 3 values: Normal, Suspected and Patologic.

2 Classification of Morphologic Pattern

Before the actual Morphologic Pattern classification I tried to visualize class disposition to

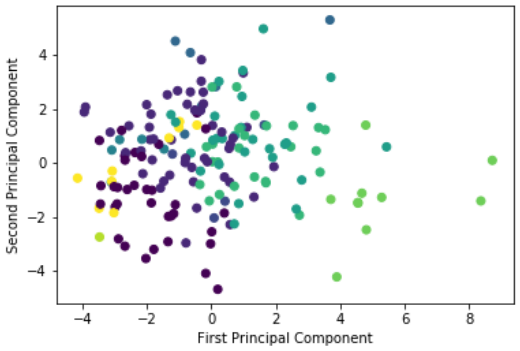
have a better idea of the most suitable classification model and to find out how to streamline the model without losing too much information; then I'll go on using linear models known to be not too much accurate then on using non linear models which will have better accuracy but with some problems like data overfitting. I manually extracted 426 examples (20% of the dataset) the to be used as Test Set. The remaining 1700 examples were divided into 1530 for the Training Set (90% of the total) and the remaining 170 for the Validation Set (10% of the total).

2.1 PCA with Two Principal Components

As said above, let's try to visualize the classes (Morphologic Pattern) using a *Principal Component Analysis*.

Using a PCA with only 2 components, it leads me to consider only 2 "Principal Component", i.e. passing from 21 to 2 features, losing information but being able to visualize class disposition.
Only Validation Set class are considered.

Figure 1: Two Principal Components and Classes of Validation Set



Making a count of the various classes of Validation Set to which the examples belong:

	Examples for each Class
'A'	34
'B'	48
'C'	4
'D'	6
'SH'	5
'AD'	28
'DE'	25
'LD'	10
'FS'	1
'SUSP'	9

From the chart we notiche that the classes are clearly overlapping and follow a provision that does not follow a linear order. If this is true, a linear model should provide us with poor performances. Let's try applying a simple one *Multiclass Logistic Regression*.

2.1.1 Multiclass Logistic Regression Model (Two Principal Components)

By applying the Multiclass Logistic Regression on the 2-component PCA model we have this results:

	ACCURACY	LOG LOSS
Train Set	0.5150	1.3251
Validation Set	0.5294	1.3749

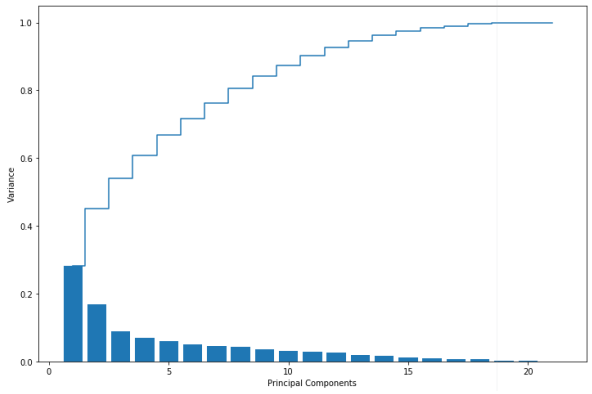
As we can see the Accuracy is very low. The reason is that, beyond the information loss derived from the PCA, the classes have not a linear disposition.

I will not waste any more time trying to improve the hyperparameters, because a linear model is not suitable for this kind of classification problem, but I can display the Variance content of each of the 21 Principal Components (the 21^{th} is very small and and it has not been plotted) to understand how many of them have the most information content and I will try to consider more main components.

2.2 PCA with Eight Principal Components

The plot contains information of every Feature and cumulated variance:

Figure 2: Variance of Features



As we can see with the first 8 principal components it's possible to have circa the 80% of variance.

In fact, considering only 2 main components (previous case) gave bad results, also because I arrived at just under 50% of the total Variance.

2.2.1 Multiclass Logistic Regression Model (Eight Principal Components)

In any case, I will show how the performances improve considering 8 main components and subsequently all components:

	ACCURACY	LOG LOSS
Train Set	0.6876	0.8781
Validation Set	0.7412	0.8544

3 MLR with all Features

We'll now consider all the features and we will show how a Linear Model will improve the perfomances.

Model performance:

	ACCURACY	LOG LOSS
Train Set	0.7954	0.6344
Validation Set	0.8000	0.6764

However, it is not our intention to use a linear classification model for this problem, therefore we pass to a model of Artificial Neural Network.

4 Artificial Neural Network

We will use all the Features for three types of networks, starting from the simplest model, the *Multilayer Perceptron* with only an Hidden Layer then moving on to two types of *Deep Neural Network*.

4.0.1 ANN with One Hidden Layer

Let's start with Artificial Neural Network with one Hidden Layer with 100 Nodes.

After some tests of hyperparameters on the λ Regularization Coefficient (L2 Regularization) of 0.0001, 0.001, and 0.01 respectively (I have noticed a general deterioration of performance for alpha of 0.1, 1 and 10. Therefore I will not use these values of the regularization coefficient in the search for the best hyperparameters) and with the Activation Functions of Hyperbolic Tangent, Sigmoid and Rectified Linear Unit, **we obtained the best results for λ = 0.001 and Activation Function Rectified Linear Unit**. We can see all the results of "Hyperparameters Tuning" of Artificial Neural Network in the last section of the project.

Performance:

	ACCURACY	LOG LOSS
Train Set	0.9392	0.1690
Validation Set	0.8824	0.3780

The Network is being trained in few seconds, so there is no need of a data simplification (some features removal) of a PCA. There no marked Overfitting, the difference between Train Set and Validation Set accuracy is approximately 5%. The model obtains logically better perfomances on the Train Set but in general it has good perfomances.
Let's try to get more accurate results by changing the Number of Nodes and the Number of Hidden Layers.

4.0.2 Deep Neural Network (1)

Deep Neural Network with 2 Hidden Layers and 256 Nodes for Hidden Layer.

After some tests of hyperparameters as done previously, I have the best performance with Activation Function Rectified Linear Unit and Regularization Term equal to 0.0001.

	ACCURACY	LOG LOSS
Train Set	0.9503	0.1480
Validation Set	0.8941	0.3950

The networks is being trained in few seconds. With this model we haven't Overfitting (the tollerance is between 10% between Train Set and Validation Set) with a difference of 6% circa. The perfomances increase with respect to the previous network.
Let's try finally to another Deep Neural Network.

4.0.3 Deep Neural Network (2)

Deep Neural Network with 150 Nodes for Hidden Layer and 3 Hidden Layer.

I obtain the best performance with Activation Funcion Hyperbolic Tangent and Regularization Term equal to 0.001.

	ACCURACY	LOG LOSS
Train Set	0.9954	0.0221
Validation Set	0.8765	0.5113

The network goes quickly in overfitting. There is a difference of circa 12% between Train Set and Validation Set accuracy; the Accuracy and Log Loss on Validation Set is even greater than the other two ANNs, so we can discard this model and use the previous one.

Overfitting is given by the excessive complexity of the Neural Network and the presence of irrelevant features (the machine is not able to tell the difference between important or irrelevant features) which leads the machine to base the final result on irrelevant features.

5 Testing Phase of Morphologic Pattern

For the Test Set, I use the Deep Neural Network with 2 Hidden Layers and 256 Nodes for Hidden Layers, with Activation Function Rectified Linear Unit and Regularization Term equal to 0.0001.

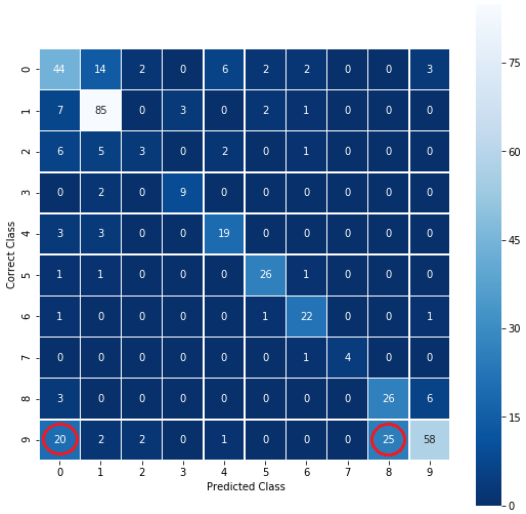
Performance:

	ACCURACY	LOG LOSS
Train Set	0.9412	0.1552
Test Set	0.6526	1.5352

There's a big Overfitting. The Accuracy on the Test Set is too low, let's try to understand why.

The problem may depend on the classes to which the examples belong, which are part of the train set and the test set. It could happen that the data for both sets are not balanced. So let's try to see through a confusing matrix which classes are wrong and if they are only some in particular.

Figure 3: Confusion Matrix of Testing



Note: Legend is 0-based while class are 1-based

In fact it happens that my model makes mistakes in particular on the examples belonging to the Class 10 ('SUSP'), classifying them instead as belonging to the Class 1 ('A') and to the Class 8 ('LD'). Making a count on the classes belonging to the Test set and Train Set.

For the Test Set (426 examples):

	Classes of Test Set
'A'	73
'B'	98
'C'	17
'D'	11
'SH'	25
'AD'	29
'DE'	25
'LD'	5
'FS'	35
'SUSP'	108

There are 108 examples with Label 10 (about 1/4 of total examples).

For the Train Set (1530 examples):

	Classes of Train Set
'A'	277
'B'	423
'C'	32
'D'	64
'SH'	42
'AD'	275
'DE'	202
'LD'	92
'FS'	33
'SUSP'	80

Only 80 examples with Label 10, less than 1/15 of total examples.

Finally it is clear that the problem comes from the division of the dataset carried out at the beginning.

The model created by the neural network is based more on some classes (highlighted in blue) and much less on the 'SUSP' class (highlighted in red) in proportion to the total number of examples. This implies that once passed to the Test phase, we realize that the Test Set contains a very high number of examples belonging to the 'SUSP' class in proportion to the total number of examples and my model is wrong to classify most of them. Mixing the dataset randomly and selecting a new Train, Test and Validation Set could solve the problem.

6 Testing Phase of Morphologic Pattern (new sets)

Once the Training Set, Validation Set and the Test Set have been shuffled more carefully, splitting the dataset not manually but through train_test_split of scikit-learn in Train Set (70% of the data), Test Set (20%) and Validation Set (10%), I have repeated the hyperparameter tuning and I obtain consistent results between

the performance of the Validation Set and the Test Set.
In addition, the best Neural Network model is the simplest one: **ANN with only 1 Hidden Layer and 100 Nodes with Activation Function Rectified Linear Unit and Regularization Term equal to 0.0001**. It was observed that using a Deep Neural Network the overfitting was easily verified when hyperparameters changed, so the simplest model was chosen.

Performance for the Validation Set:

	ACCURACY	LOG LOSS
Train Set	0.8987	0.2726
Validation Set	0.8216	0.6142

Finally performance for the Test Set:

	ACCURACY	LOG LOSS
Train Set	0.9261	0.2173
Test Set	**0.8486**	0.6222

7 Classification of Fetal Health

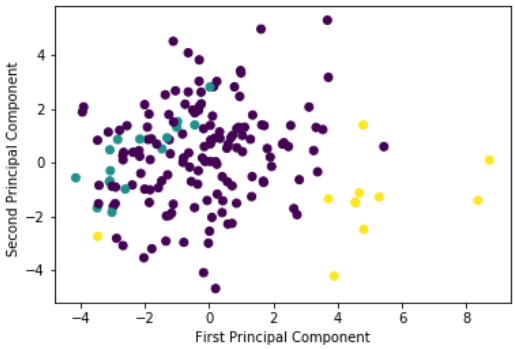
Given the fact the we analyzed the perfomances of the model in previous section, in this section we will analyze the problem at a graphical level, because the classes are only 3 and using a PCA it's possible to visualize the labels of date. Using only 2 principal components the the information content of the features will be lower, so we will see that the model performances will be not high.

We will start by using a PCA and then we will use classification model only on 2 principal components, without taking care of the their real information content (Variance analysis saw before. Given the same dataset, it will be again used a manually split in 70% of dataset as Train Test, 20% as Test Set and 10% as Validation Set (the same "wrong" principle to separate the various sets and we will see in retrospect that we will have the same problem as the Morphologic Pattern).

7.1 Principal Component Analysis

Let's make a PCA on 2 principal components and show the relative classes in Validation Set.

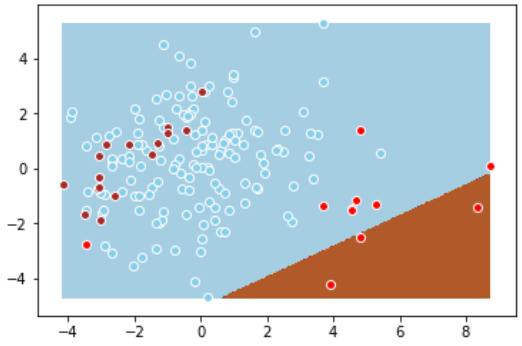
Figure 4: Two Principal Components and Correct Classes of Validation Set



7.1.1 *Multiclass Logistic Regression and Graphical Results*

Let's apply a Multiclass Logistic Regression and show the model Decision Boundaries.

Figure 5: Validation Set - Colored points: Correctly associated data; Colored zones: Multiclass Logistic Regression zones created results



Graphical Validation Set performances:

	ACCURACY
Validation Set	0.8588

We can see that the model classify in perfect way the light blue class ('Normal Fetus Health'), while it's completely wrong on the brown class ('Suspect Fetus Health') and red class ('Pathological Fetus Health')
The model is not useful in this case, despite an accuracy of 85%.

7.1.2 *SVM with Gaussian Kernel*

Given the non-linearity of the classes shown above, we can use a non-linear classification model. In this case we will use a Kernel Support Vector Machine and we will focus on optimizing the latter, without considering other possible models. We will show the performances and Decision Boundary (with the uncertainty).

Let's start with Gaussian kernel.

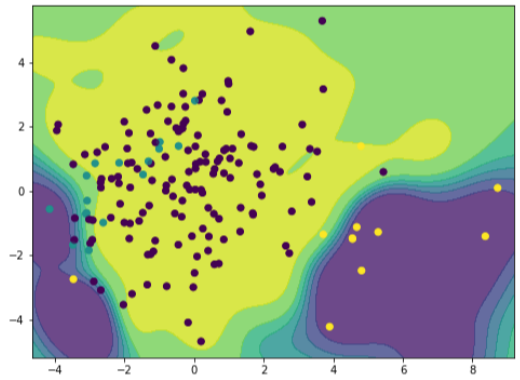
Note on graphs: beside yellow, light green, purple colors of 3 class, we will see some color gradients indicating zones where the model is uncertain. The points in a given color zone are predicted as the relative class with probability varying in dependency of the color intensity; the color of the points is the correct class of various example.

Performance:

	ACCURACY
Train Set	0.8935
Validation Set	0.8824

How the predictions were made:

Figure 6: Test Set - Colored points: correctly associated data; Colored Zones: Zones predicted by the SVM



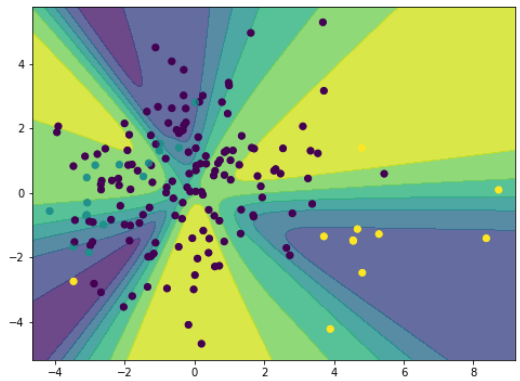
It's shown how the model despite high uncertainty has a good accuracy. We will see that gaussian kernel has the best results.

7.1.3 SVM with Sigmoid Kernel

Performance:

	ACCURACY
Train Set	0.7621
Validation Set	0.8000

Figure 7: Test Set - Colored points: correctly associated data; Colored Zones: Zones predicted by the SVM



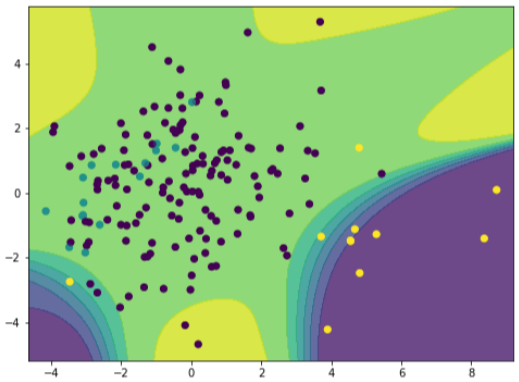
We can see the performances are way worse.

7.1.4 SVM with Polynomial Kernel

Perfomance:

	ACCURACY
Train Set	0.8725
Validation Set	0.8882

Figure 8: Test Set - Colored points: correctly associated data; Colored Zones: Zones predicted by the SVM



It has better results only than the sigmoid kernel.
textbfI'll use Polynomial Kernel for the Test Set.

8 Testing Phase of Fetal State

For the test Set I use the SVM with Polynomial Kernel.

Performance:

	ACCURACY
Train Set	0.8935
Test Set	0.7019

I obtain bad results for the Test Set and a lot of Overfitting. We already know what the problem is, because the morphologic pattern classes are linked to the health status of the fetus; consequently the various unbalanced sets cause the same problem in this part of classification. We can solve the problem in the same way as the previous case.

For the Test Set (426 examples):

	Classes of Test Set
'1'	246
'2'	141
'3'	39

For the Train Set (1442 examples):

	Classes of Train Set
'1'	1265
'2'	138
'3'	126

The Class 2 is unbalanced (1/3 for the Test Set and only about 1/10 for the Train Set)

For the Validation Set (176 examples):

	Classes of Validation Set
'1'	149
'2'	16
'3'	11

9 Testing Phase of Fetal State (new sets)

The phase of hyperparameters tuning showed significantly higher performance with the textbfGaussian Kernel. Let's see the results obtained.

Performance for the Validation Set:

	ACCURACY
Train Set	0.9399
Validation Set	0.9108

Finally performance for the Test Set:

	ACCURACY
Train Set	0.9399
Test Set	**0.8956**