**Comparative S­­­tudy on Classification of Medical Data through Multilayer Perceptrons and Support Vector Machines**

Joseph Horch

[Joseph.Horch@city.ac.uk](mailto:Joseph.Horch@city.ac.uk) | 220049677

**Abstract**This paper intends to present a critical and even-handed evaluation of two algorithms performance in a supervised classification task on Cardiotocography records from fetuses. The two algorithms compared are Multilayer Perceptron (MLP) and Support Machine Vector(SVM). Multiply experiments were conducted on each model, with differing hyperparameters in grid search. The test of the top models were evaluated and compared using confusion matrices and Receiver Operation Curves. For this classification problem, the SVM algorithm is preferred.

1. **Introduction**

During pregnancy an uneventful delivery, healthy child, and healthy mother is the desire of everyone involved. One method for examining and monitoring the health of a fetuses are fetal cardiograms (CTGs). The use of CTG reduces the incidence of birth asphyxia, where the fetus is born without a heartbeat. While this is desirable, CTG also increases the likelihood the fetus will be delivered via a caesarean section birth, a common but strenuous operation[1]. One of the challenges medical professionals face is accurately interpreting CTG data. Incorrect interpretation could lead to the death of a child or unnecessary operation.

This paper will evaluate two models on their ability to classify CTG into three classes, normal, suspect and pathologic; based off of 23 features which were reduced to seven features. The models considers are a Multilayer Perceptron(MLP) and a Support Vector Machine (SVM).

Section 2 will introduce the models in more detail and hypothesize which will perform better along with initial data exploration. Section 3 will provide description of the training, validation and testing choices as well as the hyperparameters and architecture of the models. Section 4 will summarize the results and analyze them. Lastly the fifth section will conclude this report with lessons learned and future work.

1. **Models**

MLPs, a feedforward network commonly referred to as Artificial Neural Network, are supervised learning classifiers. Developed to tackle the limitations of Rosenblatt’s Perceptron, an MLP has an input layer, output layer, and one or more hidden layers [3]. These layers extract information from features and assign modifiable weighting coefficients to components of the input layers. Each layer operates on the outputs of the its preceding layer.

The first layer of the MLP, the forward pass, receives values from the input which are modified by the weights assigned to the hidden layer to determine an output. The output is then compared to the target output. This creates a signal error, which is backpropgated and the weights are updated on the backward pass. Once the neural network is trained it can produce an output class according to input data [3].

Hidden neurons act as *feature detectors* and play a crucial role in MLP operations. As learning progresses across the MLP these neurons discover the salient features that charactize the training data, ultimately creating a *feature space.* This feature space through supervised learning distinguishes the MLP from Rosenblatt’s Perceptron [3]

SVMs are another supervised learning algorithm and used for classification and regression. SVMs aim to find the optimal hyperplane , or boundary, that maximizes the margin between different classes [2]. The margin is the distance between the closet data points from both classes. In summary a, “SVM constructs a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized” (Haykin)

When faced with a non-linearly separable problem, or multiclass problem, SVM has a novel solution called the Kernel Trick [10]. A dataset that is not linearly separable may be linearly separable in a higher-dimensional space. In other words instead of trying to separate a multiclass problem in a two-dimensional space, SVM can attempt to separate the data in a higher-dimensional space

These models are not without their limitations. According to Haykin MLPs are often desirable for their ease of use but the algorithm tends to converge slowly and lacks optimality. On the other hand SVMs are optimal but suffer from increased cost of computation.

Both models will likely deliver high accuracy due to the steps taken to extract important and relevant feautes. When examining medical data though accuracy is not everything, sensitivity or identifying true negatives is equally important. Prior studies [5] have shown that ANN only achieved a precision of 0.58 classifying the suspect class, but was still the best performing model. Therefore the accuracy of these two models may be fractionally different and I hypothesize SVM will be preferred and have a higher sensitivity with imbalanced classes like suspect and pathologic.

1. **Dataset**

The dataset used to evaluate these algorithms is from the UCI Machine Learning Repository. The dataset contains 2,126 samples with 23 attributes, 21 features and two types of classes. Classification can be performed with respect to a morphologic pattern, with 10 classes, and or fetal state with three classes. For these experiments the three classes, Normal, Suspect and Pathologic were chosen.

Initially the dataset presented an issue. While the official webpage lists 23 features once downloaded the dataset presented 36 features. Furthermore as seen in figure 1 there is a significant imbalance in the number of samples.

A picture containing screenshot, rectangle, colorfulness, square

Description automatically generatedChart, bar chart

Description automatically generated

Figure 1 non SMOTE & SMOTE classes

The imbalance was too great to be ignored so Synthetic Minority Over-Sampling Technique (SMOTE) was implemented. SMOTE is an oversampling approach in which the minority class is over-sampled creating “synthetic” examples [8]

High input dimensions of data is a prevalent problem in classification. High dimensions are computationally expensive and there is no guarantee that more inputs will improve accuracy. Therefore two experiments on the models were ran, one with SMOTE applied to the dataset and one with SMOTE was withheld.

To increase accuracy and reduce computational expense some features were eliminated. In order to determine which features were the most relevant correlation coefficient matric and K-means algorithm [4].

Chart, bar chart

Description automatically generatedFigure 2 shows the results of the K-means algorithm which used a function to perform analysis of the variance (ANOVA) on the input data (6). ANOVA is often used to identify feature that have a significant impact on the outcome variable. Not only does ANOVA help in reducing the dimensionality of the input data it also detects interactions between features that may be missed if only using a correlation matrix.

Figure 2: ANOVA

Features DP, ASTV, ALTV, Mode, Mean, Median, and CLASS were the features extracted. It is worth mentioning that SUSP feature was removed. While the feature showed correlation and had a high ANOVA score it was removed due to there being no documentation on the UCI page of the class. Thus it was determined to be an outlier.

1. **Methodology**

After extracting the feature for the MLP and SVM model the dataset was split 80% for testing and validation, 20% for testing using a stratified split methods [4].

Selecting the models entailed a grid search to adjust the hyperparameters of both MLP and SVM. The training and validation for the models was done in a 10-fold cross-validation. Chamidah demonstrated that 10-fold cross alongside the feature extraction will likely boost accuracy [4].

Early stopping, a technique used to prevent overfitting was implemented on both models. Early stopping prevents a neural network from continuing to learn by monitoring the validation set during training and halting training if the validation performance does not improve after a threshold [3]

The MLP model was constructed with one input layer, two hidden layers and an output layer. The classifier’s two hidden layers had 128 and 64 neurons respectively, with the output layer having three neurons corresponding to the three classes. The input layer size is seven for the seven input features. The hidden neuron sizes were chosen using a grid search from an earlier experiment that combined multiple suggestions including; (10,10,10), (8, 16, 32, 64) & (100, 100, 100) [12].

Rectified linear unit (ReLU) was chosen as the activation function. Traditionally the sigmoid activation functions were used but through the mid to late 2010s were shown to be susceptible to problems during training – the so-called vanishing gradients problem. ReLU is common function because of its simple implementation and effectiveness overcoming the vanishing gradient issue [10] This often leads to models performing better and is less computationally expensive. ReLU became more adopted because it allows better optimization using stochastic gradient descent, more efficient computation and is scale-invariant – meaning its characteristics are not affected by the scale of the input.

As mentioned Early stopping was implemented to prevent overfitting. This was implemented after witnessing earlier experiments were overfitting and computing for upwards of 45 minutes. A Cross-Entropy Loss Function was used to adjust the model weights during the training with the aim of minimizing the loss function. An Adam optimizer was chosen to update the MLPs weights during training.

To determine the optimal hyperparameters for model a grid search was run to determine the learning rate, batch size, weight decay, momentum and dropout rate. Weight decay is a procedure that groups weights in the network into two categories: those that have a significant influence and those that have practically little to no influence. The weights in the latter category are referred to as excess weights. These excess weights are given values close to zero to prevent the model from overfitting during training and reduce complexity [3]

These regularization techniques can help prevent overfitting which was a concern after initial experiments appeared to show SMOTE overfitting the training set.

The SVM model was more straightforward to set up. The kernel function is the first parameter specified. Utilizing a grid search it was determined the Radial Basis Function (RBF). Grid search also determined the hyperparameters of C and gamma.

1. **Results and evaluation**

During the model selection process the best MLP model (MLP I) could achieve an accuracy score of 92.28% and the best SVM (SVM I)could achieve a score of 98.28%. However these models were for all intents and purposes trained on different datasets.

|  |  |  |
| --- | --- | --- |
| Accuracy scores | MLP | SVM |
| SMOTE | 82.50% | 98.28% |
| non-SMOTE | 89.20% | 97.82% |

The MLP model with both feature extraction and SMOTE (MLP II) produced an accuracy score of 82.5% compared to the SVM model’s aforementioned 98.5%.

The SVM tested on a dataset that included feature extraction, but did not include SMOTE (SVM II), still managed to outperform MLP posting an accuracy score of 97.8%.

This can be explained by SVM’s ability to perform well with high-dimensional data but is more likely the MLP model suffering from the ‘curse of dimensionality [11]. Applying SMOTE created an equilibrium within the imbalanced classes but it increased the dimensionality. Along with a lower accuracy the MLP predicted more False Negatives and False Positives.

A screenshot of a graph

Description automatically generated with low confidenceWhen examining medical data it is equally important to classify a person who needs surgery as it is a person who does not. In these cases only the Suspect and Pathologic classes are relevant, Normal class indicates no need for surgery. Both non-SMOTE models were outperformed their SMOTE counterparts considerably as seen in the figure 3. Of particular interest was SVM II to correctly classify the pathologic class (this could also be overfitting).

Figure 3 testing confusion matrices

A Receiver Operation Characteristic Curve (ROC) was plotted (Figure 4) to check the quality of the classifiers. The Area Under the ROC Curve, measure classification accuracy.

**A picture containing text, line, diagram, parallel

Description automatically generated**Both models have similar AUCs for the normal and pathologic classes but diverge on the suspect class. The suspect class is the hardest to classify [8]. Therefore the model that more accurately classifies the suspects true positive and false positive rate should be preferred.

Figure 4 ROC/AUC of best models

1. **Conclusion and future learning**

This report reviewed how accurately an MLP and SVM models can classify multiclass data. The classes were normal, suspect or pathologic.

The conclusion is that while both models performed well and had comparably accuracies for some of the classes, SVM showed to be the better algorithm for this problem with higher accuracy and less computational run time. SVM after all tends to handle high dimension data better than MLP.

Both ROC and confusion matrices are strong tools for evaluating classifiers. They measure accuracies well and provide compelling visualizations for the sensitivity and specificity of the models.

I learned that without SMOTE and feature engineering the accuracy results would not have nearly been as high. It would be interesting to investigate if PCA would have impacted the scores. Future experiments could also include under-sampling the majority class (normal ‘1’ in this case) to increase the sensitivity of the classifiers to minority classes [8].

1. **References**

[1] Steer PJ. Has electronic fetal heart rate monitoring made a difference. Semin Fetal Neonatal Med. 2008 Feb;13(1):2-7. doi: 10.1016/j.siny.2007.09.005. PMID: 18271079.

[2] Ng, A. (n.d.). CS229 Lecture notes, Support Vector Machines. [online] pp.1–25. Available at:  
https://moodle.city.ac.uk/pluginfile.php/3032217/mod\_resource/content/1/svm-notes.pdf

[3] Haykin, S.S. (2016). *Neural networks and learning machines*. Delhi ; Chennai: Pearson.

[4] Chamidah, N. and Wasito, I. (2015b). *Fetal state classification from cardiotocography based on feature extraction using hybrid K-Means and support vector machine*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICACSIS.2015.7415166.

[5] Sundar, C., M. Chitradevi, & G. Geetharami (2013), Overview of Research Challenges for Classification of Cardiotocogram Data *Journal of Computer Science*, 9(2), pp.198–206. <https://thescipub.com/abstract/10.3844/jcssp.2013.198.206>

[6] Bichen Zheng, Sang Won Yoon, Sarah S. Lam, Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms, Expert Systems with Applications, Volume 41, Issue 4, Part 1, 2014, Pages 1476-1482, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2013.08.044>

[7] Kim, E. (2017). *Everything You Wanted to Know about the Kernel Trick (But Were Too Afraid to Ask)*. [online] Available at: <https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html>

[8] Chawla, N., Bowyer, K., Hall, L. and Kegelmeyer, P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. [online] Available at:https://moodle.city.ac.uk/pluginfile.php/3032216/mod\_resource/content/1/smote.pdf.

[9] Brownlee, J. (2020). *How to Perform Feature Selection With Numerical Input Data*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/feature-selection-with-numerical-input-data/>.

[10] Brownlee, J. (2021). *How to Choose an Activation Function for Deep Learning*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/>.

[11] Wang, Z., Wang, Y., Xuan, J., Dong, Y., Bakay, M., Feng, Y., Clarke, R. and Hoffman, E. (2005). Optimized multilayer perceptrons for molecular classification and diagnosis using genomic data. *Bioinformatics*, [online] 22(6), pp.755–761. doi:https://doi.org/10.1093/bioinformatics/btk036.

[12] Subasi, Abdulhamit. (2020). *Practical Machine Learning for Data Analysis Using Python.* Elsevier. Retrieved from https://app.knovel.com/hotlink/toc/id:kpPMLDAUP1/practical-machine-learning/practical-machine-learning