Reproducing the Paper: BREAST CANCER DIAGNOSIS FROM FINE-NEEDLE ASPIRATION USING SUPERVISED COMPACT HYPERSPHERES AND ESTABLISHMENT OF CONFIDENCE OF MALIGNANCY*

Alpamys Tolegen†
(Dated: April 21, 2019)

This paper aims to replicate another paper called 'Breast Cancer: Diagnosis from the Fine-Needle Aspiration Using Supervised Compact Hyperspheres and Establishment of Confidence of Malignancy"

I. EXECUTIVE SUMMARY

This paper aims to reproduce the work on Breast Cancer Detection by using machine learning algorithms. Nowadays, majority of research papers' goal in medicine is to diagnose the disease with the aid of computers and get beyond human specialists' ability in recognizing the disease. In addition, some papers introduce algorithms that help to classify the disease since each disease can have several or more types that should treated differently, as in case of breast cancer.

Breast cancer is a tumor that can be classified as benign and malignant. Benign tumor do not threaten the life of the patient whereas malignant tumor can have lethal consequences. Breast cancer affects approximately 10% of all women at some stage of their life. Moreover, this disease is the second most common cause of cancer deaths among women. Therefore, having accurate methods of diagnosing and classifying the disease is essential to save millions of life.

II. DATA DESCRIPTION

In this paper, the Breast Cancer Wisconsin Data set is used. It has 569 observations of tumors and has 30 features of a tumor marked as B(benign) or M(malignant). The input variables are:

- *Radius* - computed by averaging the length of radial line segments from the center of mass of the boundary to each of the boundary points.
- *Perimeter* - measured as the sum of the distances between consecutive boundary points.
- *Area* - measured by counting the number of pixels on the interior of the boundary and adding one-half of the pixels on the perimeter, to correct for the error caused by digitization.
- *Compactness* - combines the perimeter and area to give a measure of the compactness of the cell, calculated as squared perimater over area
- *Smoothness* - is quantified by measuring the difference between the length of each radial line and the mean length of the two radial lines surrounding it
- *Concavity* - captured by measuring the size of any indentations in the boundary of the cell nucleus.
- *Concave points* - similar to concavity, but counts only the number of boundary points lying on the concave regions of the boundary, rather than the magnitude of such concavities
- *Symmetry* - measured by finding the relative difference in length between pairs of line segments perpendicular to the major axis of the contour of the cell nucleus
- *Fractal dimension* - approximated using the coastline approximation. The perimeter of the nucleus is measured using increasingly larger rulers. As the ruler size increases, the precision of the measurement decreases, and the observed perimeter decreases. Plotting these values on a log-log scale and measuring the downward slope gives the negative of an approximation to the fractal dimension.
- *Texture* - measured by finding the variance of the grayscale intensities in the component pixels.

Using these 10 input variables, 30 were made by taking mean, standard deviation and extreme values.

III. PREPROCESSING

Authors of the original paper have only done dimensionality reduction by Genetic Algorithm (GA) as preprocessing. However, it seems more things should be done with the dataset before deploying models. Firstly, multicollinearity should be removed. By multicollinearity we mean two or more features that are extremely correlated ($\rho \geq 0.9$). Such severe multicollinearity is quite problematic since it increases the variance of the coefficient

* A footnote to the article title

† Also at Mathematics Department, Nazarbayev University.

estimates. To remove it, only one of the highly correlated columns should be left. In this dataset 10 such columns were removed leaving with 20 features. Further, Genetic Algorithm (GA) was used for feature selection resulting in reduced dataset with only 10 features left. For GA, code of the user *kaushalshetty* from GitHub was used with some modification that made results reproducible. For GA, the following parameters were used:

- *Population size* - 50
- *Crossover rate* - 0.8
- *Mutation rate* - 0.05
- *Number of generations* - 300

GA with these parameters was also used in the original paper, however, I could not get the same results since the subset of selected features depends on the random seed. The number of selected features varies depending on seed (from 7 to 13). Thus, this part of the paper is difficult to reproduce.

Authors of the original paper also have included PCA plots of train and test in their paper. However, the weird thing is that the distribution of benign and malignant masses in plots of train and test are dramatically different. Such thing should not happen if random splitting of data was performed. At first, there was a suspicion that may be it is because of some weird random seed that was used for splitting, however, it seems not the case since 30 different seeds were examined and none of them resulted in such dramatic differences in distributions. This can mean several things: either authors have not done random splitting or manipulated the data splitting in order to maximize metrics of their method. Figure 1 is the illustration of how should well randomly split samples' plots should look.

IV. MODELLING

I was not able to implement authors' method called Supervised Compact Hyperspheres which is based on Minimum Enclosing Ball algorithm. Nevertheless, it seems their algorithm was not good enough since even after more than 10 years it was not used in other papers.

I have constructed 3 models and they are Linear Discriminant Analysis, Kernel Fisher's Discriminant Analysis and Support Vector Classifier. I could not find Python modules or open-source codes that have implemented KFDA, thus I have used mine Also, I was not able to implement it in a way that is able to give prediction, thus, after KFDA transformation I was using KNeighbours method to do the classification. Moreover, authors were using rbf kernels for KFDA and SVC, and then tuning other parameters with the help of cross-validation. Mine experiments with Grid Search Cross Validation method for parameter choosing show that rbf is not always the

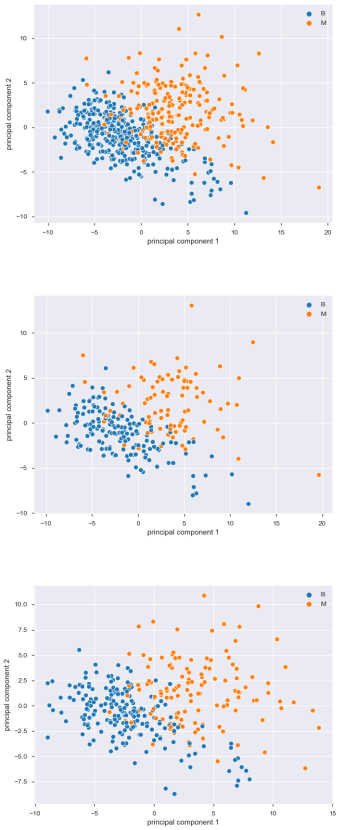

FIG. 1: Three simple graphs

best choice. SVC benefits most from the linear kernel with C=100. F1-score was chosen as the metric to judge these models. After tuning of the models and getting predictions on test, the following results were obtained: LDA - 0.93703, SVC - 0.94898 and KFDA - 0.9113.

V. CONCLUSION

To conclude, it seems the paper in not fully reproducible. First of all, GA was used for feature selection and this method does not give always the same results if random seed is not set, so this is the first obstacle in reproducing the paper. Second one is data splitting into train and test. As it was mentioned, authors' splitting was not correct. Finally, their algorithm called Supervised Compact Hyperspheres is difficult to implement.