Two Category Classification Using Baysian Decision Rule.

Melton, Chad.A[a,b,c]

[a] *University of Tennessee, Knoxville*
[b] *Bredesen Center for Interdisciplinary Research*
[c] *COSC 522: Machine Learning*

Abstract

Machine learning algorithms are used to gain insight into the large volumes of data that accompany a variety of scientific domains. In particular, Bayesian decision rule is a powerful, supervised machine learning tool that can be used for classification of multiple variables. In this study, my objective was to classify synthetic test data with three types of discriminant functions and evaluate their performance. To accomplish this task, MAP (Maximum A Posteriori) estimation was used with training data to derive and illustrate decision rules for the three cases of discriminant functions. To evaluate the performance, accuracy for each case was tested over prior probabilities ranging from 0.01-1.0 in 0.01 increments. The performance of a one-modal Gaussian was compared on data that were modeled with a two-modal Gaussian. The accuracy of Case 1 was determined to be approximately 0.72 and was at its optimal performance when prior probability was approximately 0.55. The maximum accuracy for Case 2 was calculated to be approximately 0.76 and at a prior probability of 0.46. Case 3 was determined to be optimal at 0.90 and performed best when prior probability was at approximately 0.46. Finally, the accuracy of the one-modal Gaussian method was determined to be 0.879.

1. Introduction

With the up-scaling of computational platforms and faster processing rates, various machine learning algorithms have been used to investigate a wide variety of scientific and social phenomena ranging from automatic cloud detection on Mars [1], to secondary hospital visits due to pediatric asthma patients [2]. Though there are many great machine learning algorithms, Bayesian decision rule makes use of discriminant functions to determine classification decision boundaries. This method includes the training of an algorithm to determine the classification of a data set in question. These discriminant functions can be expressed as three different cases (i.e., Case 1, Case 2, Case 3). Case 1 and Case 2 are considered linear classifiers while Case 3 is a quadratic classifier [3]. For Case 1 ($\mathbf{\Sigma}_i = \sigma^2\mathbf{I}$) , covariance matrices are considered equal, proportional to an identity matrix, and yields the Euclidean distance. With Case 2 ($\mathbf{\Sigma}_i = \mathbf{\Sigma},$) covariance

Email address: chadmeltone@gmail.com (Melton, Chad.A)

matrices are considered in the calculation and the solution yields the squared Mahalanobis distance. Finally, Case 3 ($\mathbf{\Sigma}_i = \mathbf{arbitrary}$) assumes arbitrary covariance matrices and yields a quadratic result [4][3][5].

1.1. Background and Previous Work

Rooted in Bayesian statistics, the mathematics behind discriminant functions were developed in the mid 1700's [6]. Over the last approximately 80 years, discriminant functions have been employed by various scientific domains where pattern recognition could provide insight into a study area. For example, some of these domains include eugenics [7], physical anthropology [8], the human genome [9], and target detection using hyperspectral analysis [10]. Though many works are focused on scientific research, others have been more concerned with accuracy of this methodology. The focus of this work presented in this document is directly concerned with algorithm performance.

1.2. Objective

The first major objective of this work was to determine the decision boundaries of a two-category synthetic data set. The second objective of this work was to extensively evaluate the accuracy of the decision rules based on a range of prior probabilities. The last objective of this study was to evaluate the performance of a one-modal compared to a two-modal Gaussian modeled data set.

2. Technical Approach

2.1. Algorithms and Methodology

To accomplish the objectives for this study, data (synth.tr and synth.te) were obtained from Pattern Recognition and Neural Networks by B.D. Ripley (http://www.stats.ox.ac.uk/pub/PRNN/) and read into python as a CSV. The first step in this process was to separate the training data (synth.tr) into two separate list based on the "y" label attribute which represented Class 0 and Class 1. The parameters of the data were then calculated by using the Maximum Likelihood Estimation method. Training and testing data were then plotted with matplotlib (https://matplotlib.org).

2.2. Decision Rules Calculation

The next milestone of this study involved determining coefficients/decision boundaries for each of the 3 cases. The method for determining these values was similar in that the discriminant function equations for each case were set equal to each other, simplified and solved for coefficients. This step was done by employing Sympy (https://www.sympy.org/), numpy (numpy.org), and manual calculations in Python. One

set of equations used parameters for Class 0 and the set used parameters for Class 1. Equations were broken up into smaller components in order to simplify coding efforts.

At this point, Case 1 was calculated with:

$$g_i(\mathbf{x}) = -\frac{1}{2\sigma^2}(\mathbf{x}-\mu_i)^t(\mathbf{x}-\mu_i) + \ln P(w_i), \qquad (1).$$

Once expanded, equations for Class 0 and Class 1 can be set equal to each other to find the decision boundary:

$$g_i(\mathbf{x}) = -\frac{1}{2\sigma^2}(-2\mu_i^t\mathbf{x} + \mu_i^t\mu_i + \ln P(w_i) = \frac{\mu_i^t}{\sigma^2}\mathbf{x} + (-\frac{\mu_i^t\mu_i}{2\sigma^2} + \ln P(w_i), \qquad (2)$$

and then, the components can be simplified further to: $w_i = \frac{\mu_i^t}{\sigma^2}$ (3), and $w_{i0} = -\frac{\mu_i^t\mu_i}{2\sigma^2} + \ln P(w_i), \qquad (4),$ can be represented as a linear equation:

$$g_i(\mathbf{x}) = w_i\mathbf{x} + w_{i0}, \quad (5).$$

After coefficients were determined, the equation for Case 1 was plotted.

Case 2 was calculated with:

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\mu_i)^t\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mu_i) + \ln P(w_i), \quad (6)$$

Once expanded, equations for Class 0 and Class 1 can be set equal to each other to find the decision boundary:

$$g_i(\mathbf{x}) = -\frac{1}{2}(-2\mu_i^t\mathbf{\Sigma^{-1}}\mathbf{x} + \mu_i^t\mathbf{\Sigma^{-1}}\mu_i) + \ln P(w_i) = \mu_i^t\mathbf{\Sigma^{-1}}\mathbf{x} + (\ln P(w_i) - \frac{1}{2}\mu_i^t\mathbf{\Sigma^{-1}}\mu_i), \quad (7)$$

.

Then, the components can be simplified further to:$w_i = \mu_i^t\mathbf{\Sigma^{-1}}\mathbf{x}$ (8), and $w_{i0} = \ln P(w_i) - \frac{1}{2}\mu_i^t\mathbf{\Sigma^{-1}}\mu_i, \quad (9)$ can be represented as a linear equation:

$$g_i(\mathbf{x}) = w_i\mathbf{x} + w_{i0}, \quad (10)$$

After coefficients were determined, the equation for Case 2 was plotted.

The most complex of the three cases, Case 3 was calculated with:

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}^t\mathbf{\Sigma_i^{-1}}\mathbf{x} - 2\mu_i^t\mathbf{\Sigma_i^{-1}}\mathbf{x} + \mu_i^t\mathbf{\Sigma_i^{-1}}\mu_i) - \frac{1}{2}\ln|\mathbf{\Sigma}_i| + \ln P(w_i) =$$

$$= \mathbf{x}^t(-\frac{1}{2}\mathbf{\Sigma_i^{-1}})\mathbf{x} + \mu_i^t\mathbf{\Sigma_i^{-1}}\mathbf{x} + (-\frac{1}{2}\mu_i^t\mathbf{\Sigma_i^{-1}}\mu_i) - \frac{1}{2}\ln|\mathbf{\Sigma}_i| + \ln P(w_i)), \quad (11)$$

Finally, the components can be simplified further to:$W_i = -\frac{1}{2}\mathbf{\Sigma}_i^{-1}$ (12), $w_i = \mathbf{\Sigma}_i^{-1}\mu_i)$ (13), and $w_{i0} = -\frac{1}{2}\mu_i^t\mathbf{\Sigma_i^{-1}}\mu_i) - \frac{1}{2}\ln|\mathbf{\Sigma}_i| + \ln P(w_i)$, (14). These results were different in that the calculation yielded a quadratic equation:

$$g_i(\mathbf{x}) = \mathbf{x}^t W_i\mathbf{x} + w_{i0}\mathbf{x} + w_{i0}, \quad (15)$$

.

2.3. Classification Accuracy and Prior Probability

Classification accuracy was determined by creating functions for G_0 and G_1 for each class. A loop was then constructed that read through the synth.te and classified values according to class and the total classification quantity. This step was achieved by writing a conditional expression that was embedded within the loop. Finally, the number of classified values were divided by the total count of the test data to retrieve overall accuracy. At this point, this step was conducted assuming a prior probability of 0.5 for each class.

To find the best prior probability for optimal accuracy, an additional loop was created to iterate through the synthetic test data (synth.te) over a range of prior probabilities from 0.01 to 1.0 in increments of 0.01. Values from the classification accuracy function were stored in an array and plotted for comparison. Additionally, the numpy function, argmax and max were used to record the highest indices and value (respectively) in the array in order to obtain the prior probability that yielded the highest accuracy.

2.4. Two/One-Modal Gaussian

The first step of the the two/one-modal Gaussian comparison analysis was performed by running MATLAB code provided by Dr. Hairong Qi to plot our synthetic test data. The provided code plotted the two-modal contour model based on previously estimated mean and standard deviation.At this point, an approximate estimation of mean and standard deviation was made visually based on the center of the Class 1 clusters. Additional code was constructed to plot a one-modal contour plot around the Class 1 testing data based on these estimated parameters. After results were visually inspected, code was then constructed in python for each class using the equation:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^t\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right], \quad (16)$$

. Finally, results were analysed in similarly constructed loops described in the previous analyses in this study.

3. Results

3.1. Maximum Likelihood Estimation

Results from the MLE function for Class 0 and Class 1 of the synth.tr were returned for the mean and covariance matrices. The mean array for x1 and x2, for Class 0 was calculated to be: [-0.22147024, 0.32575494].The mean array for x1 and x2, for Class 1 was calculated to be:[0.07595431, 0.68296891]. The covariance matrix for Class 0 was calculated to be:([0.274595, 0.011139],[0.011139 0.03583]. Lastly, the covariance for Class 1 was calculated to be: ([0.15847, -0.01545], [-0.01545, 0.029719]).

3.2. Decision Rules for Case 1, Case 2, and Case 3

After implementing the equations in python and solving with sympy, results for Case 1 yielded -0.832622943054845x + 0.443781977932576. Results for Case 2 returned the equation -0.665898413008571x + 0.455912515037513. Lastly, the quadratic equation for Case 3 returned as: 0.57181875114983e-16*sqrt(-3.83486666203573e+27*x1**2 - 4.99510421701672e+30*x1 + 8.3317316461565e+30) + 2.09062736578341 (see Fig 1).

3.3. Classification Accuracy and Evaluation

The maximum accuracy for Case 1 was determined to be approximately 0.72 and was at its optimal performance when prior probability was approximately 0.55. The maximum accuracy for Case 2 was determined to be approximately 0.76. However, it was at its optimal performance when prior probability was approximately 0.46. Case 3 performed the best out of the tested discriminant functions. The accuracy for Case 3 was determined to be optimal at 0.90 and performed best when prior probability was at approximately 0.46 (see Fig 2).

3.4. Two-Modal/One-Modal Gaussian Performance

Using the visually chosen parameters for the data, the two-modal, one-modal comparison yielded relatively similar results to Case 3. Assuming equal prior probability of 0.5, accuracy was determined to be approximately 0.87 (see Fig 3 and Fig 4).

4. Discussion

4.1. Decision Rules Evaluation

Because Case 1 and Case 3 are linear classifiers, the plots were linear in nature but exhibited different slopes. Case 3 displayed a parabolic plot due to originating from a quadratic equation. All three plots appeared to intersect at a single point. At optimal prior probability (0.5), Case 3 outperformed Case 1 and Case 2 by approximately 0.15. Case 2 generally performed better than Case 1 except for when prior

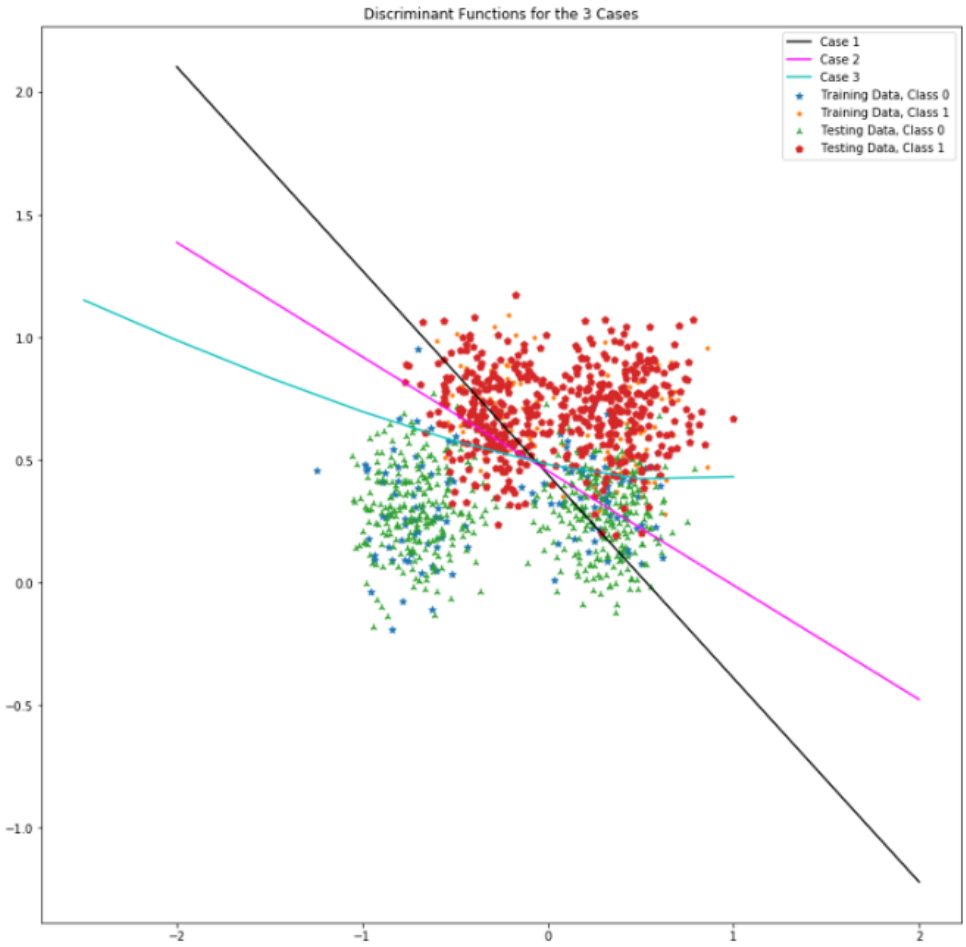

Figure 1: Cases 1, 2, and 3 decision boundaries overlaying synthetic test and training data. Case 1 is represented by the black line. Case 2 is represented by the magenta line, and Case 3 is represented by the cyan parabola.

probability was at approximately 0.65-0.75. In this range, the accuracy of case one briefly increases and then sharply decreases. Overall, these results were somewhat expected because quadratic functions can usually capture more nuances from a data that are not linearly separated or linearly stretched. This fact was directly observable in the plot of all three cases. In this plot, Case 3 divided the cases more clearly and approximately captured the curvature of the boundary between Class 1 and Class 2, where as the lines for Case 1 and 2 did not seem to capture this nuance (see Fig 3).

4.2. Accuracy vs Prior Probability

My analysis of accuracy compared to prior probability provide interesting insight into how prior probability can affect the results of a study. Such results could prove useful in future studies when choosing a methodology for dealing with data that may have a skewed or less than "ideal" prior probability. These presented results also show the importance of a thorough understanding of the domain knowledge that is being investigated so that a correct model can be chosen based on prior probability. Nonetheless, if given a choice of a discriminant

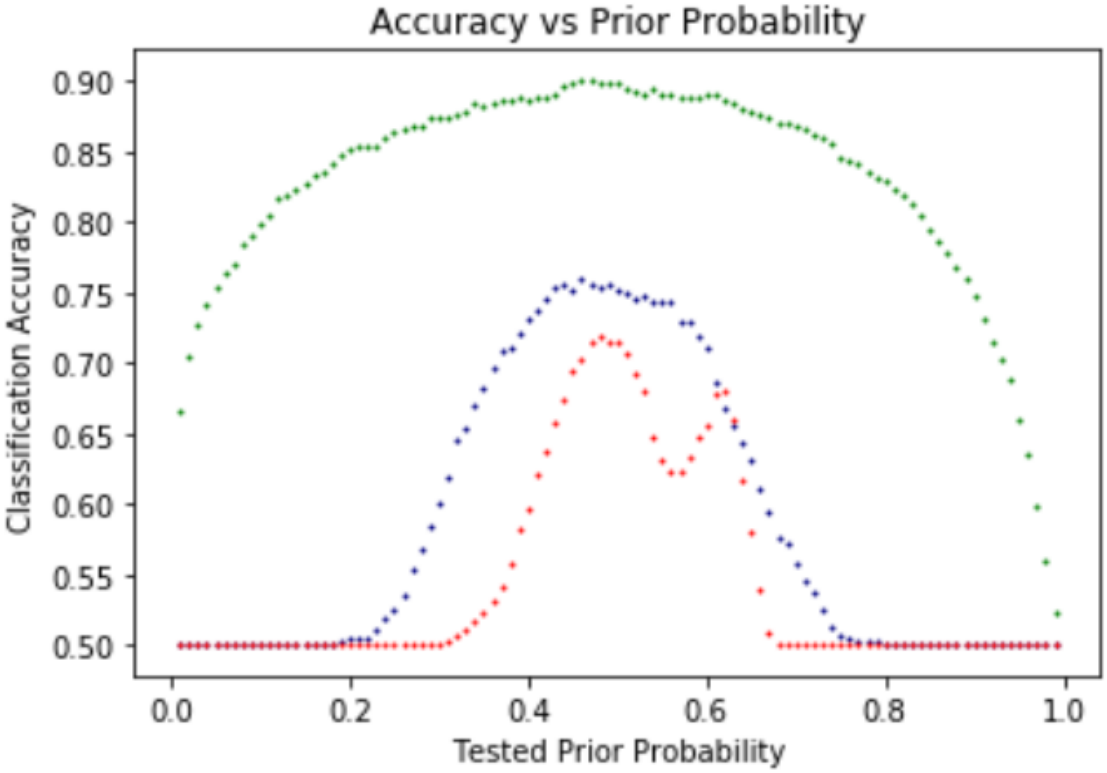

Figure 2: Accuracy vs prior probability. The red doted line represents Case 1 accuracy. The navy dotted line represents Case 2 accuracy. Finally, the green dotted line represents Case 3 accuracy.

function, I would choose to use Case 3 due to the overall better and well-rounded performance by the algorithm across a range of prior probabilities. Generally, all algorithms tested in this study performed better when probabilities were approaching 0.5. Accuracy tended drop off continually as prior probability values approached 0.01 and 1.

5. Conclusion

Synthetic test data was classified using three cases of discriminant functions that were trained with synthetic training data. A two-modal Gaussian also modeled the data and used to compare the performance with a one-modal Gaussian. The accuracy of these 4 models was evaluated at a range of prior probabilities from 0.01 to 1.0. The accuracy of Case 3 was evaluated to be the highest while the one-modal Gaussian was measured to be the 2nd highest. The accuracy of Case 2 ranked 3rd and finally Case 1 had the lowest accuracy. It would be interesting for future studies to employ similar methodology on different data set from a real example, perhaps with more variables. Other future studies could incorporate other methods to test algorithm performance such as ROC plots.

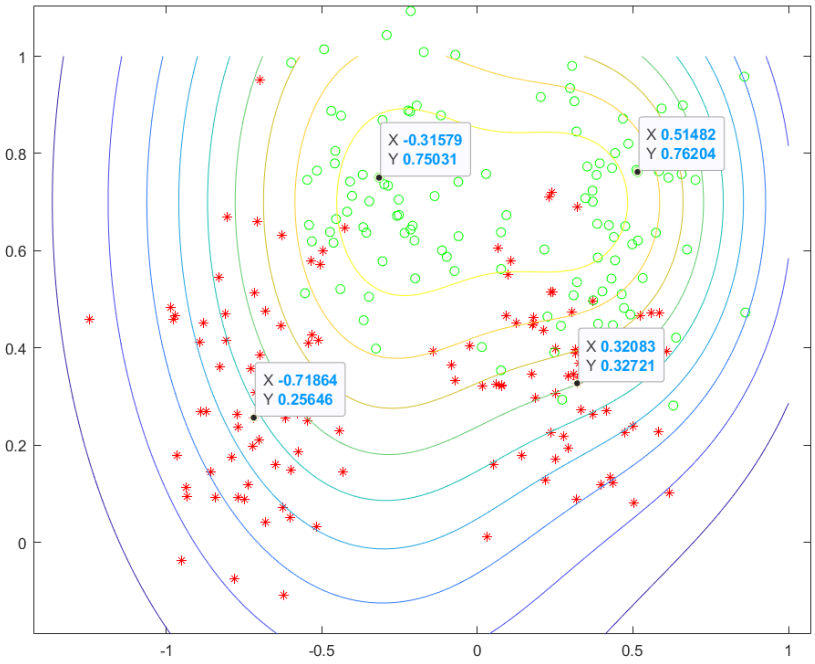

Figure 3: One-modal Gaussian plotted in MATLAB. The boxes containing x,y values represent mean chosen for analysis. The boxes closest to the red data points approximately represent parameters chosen by Dr.Qi and the values in boxes within the yellow contours over the green data points were qualitatively chosen by the author.

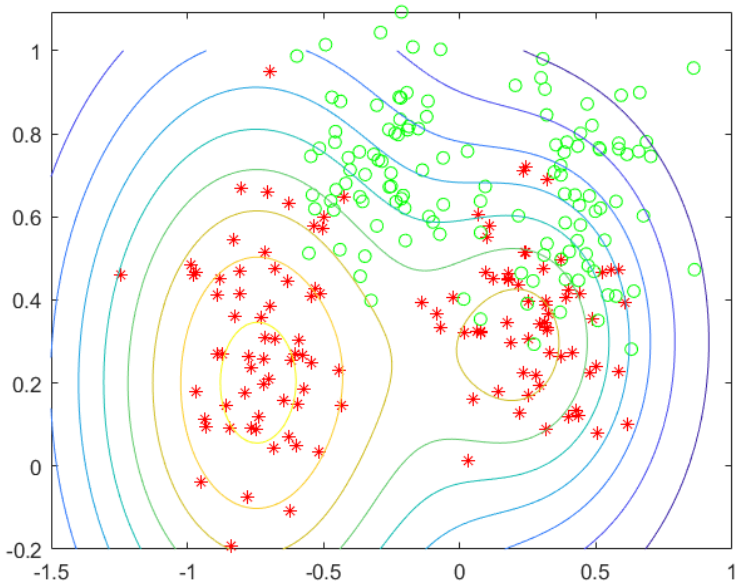

Figure 4: Two-modal Gaussian plot from MATLAB. Courtesy of Dr. Qi.

References

[1] A. Castano, A. Fukunaga, J. Biesiadecki, L. Neakrase, P. Whelley, R. Greeley, M. Lemmon, R. Castano, S. Chien, Automatic detection of dust devils and clouds on mars, Machine Vision and Applications 19 (5-6) (2008) 467–482.

[2] E. K. Shin, R. Mahajan, O. Akbilgic, A. Shaban-Nejad, Sociomarkers and biomarkers: predictive modeling in identifying pediatric asthma patients at risk of hospital revisits, npj Digital Medicine 1 (1) (2018) 50.

[3] R. O. Duda, P. E. Hart, D. G. Stork, Pattern classification, John Wiley & Sons, 2012.

[4] H. Qi, Lecture 3 - discriminant function and normal density (2019).
URL `http://web.eecs.utk.edu/~hqi/cosc522/lecture03-discriminant.pdf`

[5] C. Y, iscussion about discriminant functions for the multivariate normal density (2014).
URL `https://www.projectrhea.org/rhea/index.php/Discussion_about_Discriminant_Functions_for_the_Multivariate_Normal_Density`

[6] T. Bayes, An essay towards solving a problem in the doctrine of chances. 1763., MD computing: computers in medical practice 8 (3) (1991) 157.

[7] R. A. Fisher, The precision of discriminant functions, Annals of Eugenics 10 (1) (1940) 422–429.

[8] E. H. Ashton, M. Healy, S. Lipton, The descriptive use of discriminant functions in physical anthropology, Proceedings of the Royal Society of London. Series B-Biological Sciences 146 (925) (1957) 552–572.

[9] V. V. Solovyev, A. A. Salamov, C. B. Lawrence, Identification of human gene structure using linear discriminant functions and dynamic programming., in: Ismb, Vol. 3, 1995, pp. 367–375.

[10] E. Lo, Hyperspectral target detection based on classification algorithms.