Abstract. We ran classification algorithms LDA, SPRT, and a modified SPRT on biological datasets for Parkinson’s disease [4,5], colon cancer [6], and breast cancer [9]. The SPRT algorithms were run with components in decreasing variance order and random order. Results for those in random order were calculated as the majority predictions over 100 runs. Truncation was always set to the total number of components of the dataset. Accuracies for each algorithm were determined using the method of leave-one-out cross-validation. The highest accuracy for the Parkinson’s disease dataset was 0.7128, generated by the MSPRT random function at $\alpha = \beta = 0.175$ and by the SPRT random function at $\alpha = \beta = 0.32$. The highest accuracy for the colon cancer dataset was 0.8871, generated by the MSPRT ordered function at $\alpha = \beta = 0.22$. The highest accuracy for the breast cancer dataset was 0.9648, generated by the MSPRT ordered function at $\alpha = \beta = 0.07$ and by the SPRT ordered function at $\alpha = \beta = 0.09$.

Keywords: classification, discriminant analysis, sequential probability ratio test
1 Introduction to Statistical Classification

Statistical classification is a method of machine learning where items are grouped based on quantitative information that is known about the items and based on a training set of data with known statuses. In proper form, the problem is known as: Given a training dataset \( \{(x_1, y_1), \ldots, (x_n, y_n)\} \) to produce a classifier for this dataset such that \( h: x \rightarrow y \) is able to map any object \( x \in \mathcal{X} \) to its true status \( y \in \gamma \).

Then individual probability is determined of the object being in a certain class. In our project we use a Parkinson’s dataset where we try to identify if a patient has Parkinson’s disease or not, in this unique case, the probability would look like \( P(\text{parkinson} | x) = f(x; \theta) \) where \( x \) is the vector input to the function and \( \theta \) is a parameter. This being such it is easy to determine the opposite status as well \( P(\text{noparkinson} | x) = 1 - P(\text{parkinson} | x) \).

However this is only one approach to the problem, the other approach, known as the Bayesian approach, is to not use a singular parameter vector of \( \theta \) but to integrate over all possible thetas weighted based on the training data, \( D \) in this case:

\[
P(\text{parkinson} | x) = \int f(x; \theta) P(\theta | D) d\theta
\]

Again to find the case in which the status isn’t Parkinson’s is the same because it is a linear case where there are only two outcomes, but for any distinct probability \( P(\text{class}_i | x) = 1 - P(\text{class}_{i+1} | x) \).

The problem then becomes which algorithm should be used to determine the distinct probability. K-nearest neighbor is one of the most commonly used algorithms because it is a simple algorithm. The k-nearest neighbor fit for \( \hat{y} \) is equal to

\[
\hat{y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i,
\]

where \( N_k(x) \) is the neighborhood of \( x \) defined by \( k \) closest points \( x_i \) in the training sample. Another common method is to use logistic regression which tries to fit the probability of data being in a particular class by fitting the data to a logistic curve produced by the Logic function, to which the fit is \( \hat{y} = X(X^T X)^{-1} X^T Y \). Our project uses Linear Discriminant Analysis which relies on some Bayesian principles that are used when the covariances in a dispersion matrix are equal to 0, which implies independence.

In this report we use 3 different data sets and 3 different methods to classify the objects that we use in our training data set using a leave one out method, we then calculate how accurate the methods were on the data sets, and compare and contrast the methods.
2 Methodology

There are three different common algorithms that we have utilized to solve the classification problem: Linear Discriminant Analysis (LDA), Sequential Probability Ratio Test (SPRT), and Modified SPRT (MSPRT). MSPRT is inherently a modified SPRT, while SPRT is essentially a modified (sequential) LDA, which is the most primitive of the algorithms. The details of each classification algorithm and the differences between them are outlined below.

2.1 LDA

Linear discriminant analysis (LDA) [1] is a well-studied classification algorithm. It assigns an instance described by \( \mathbf{x} = (x_1, x_2, ..., x_p)^T \), a \( p \)-dimensional feature vector, to \( \arg\max_g \Pr(G = g|\mathbf{X} = \mathbf{x}) \), the class with the highest posterior probability. By Bayes’ theorem, we know that

\[
\Pr(G = g|\mathbf{X} = \mathbf{x}) \propto \Pr(\mathbf{X} = \mathbf{x}|G = g) \Pr(G = g)
\]

and if we further assume that \( \Pr(G = g) \)'s are the same for all \( g \)

\[
\Pr(G = g|\mathbf{X} = \mathbf{x}) \propto \Pr(\mathbf{X} = \mathbf{x}|G = g).
\]

Therefore, class label assignment amounts to finding \( \arg\max_g \Pr(\mathbf{X} = \mathbf{x}|G = g) \).

LDA assumes that, conditioned on the class label, the feature vector of an instance is distributed as a multivariable normal distribution. That is,

\[
\Pr(G = g|\mathbf{X} = \mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^2} \exp\left(-\frac{1}{2} (\mathbf{x} - \mu_g)^T \Sigma^{-1} (\mathbf{x} - \mu_g) \right),
\]

where \( \mathbf{x} \) is the component vector of an instance belonging to class \( g \) with mean vector \( \mu_g \) and dispersion matrix \( \Sigma \) common to all classes. The parameters \( \mu_g \)'s and \( \Sigma \) can be estimated with a training dataset. In the binary case, finding \( \arg\max_g \Pr(\mathbf{X} = \mathbf{x}|G = g) \) is equivalent to computing the likelihood ratio

\[
\Lambda = \frac{\Pr(\mathbf{X} = \mathbf{x}|G = 1)}{\Pr(\mathbf{X} = \mathbf{x}|G = 2)}.
\]

The instance \( \mathbf{x} \) is assigned to class 1 if \( \Lambda > 1 \) or class 2 if \( \Lambda < 1 \). Binary LDA is therefore closely related to the probability ratio test.
2.2 SPRT

Fu [2] assumes that the components of \( \mathbf{x} \) are independent and can be observed sequentially. Consequently,

\[
\Lambda = \frac{\Pr(\mathbf{X} = \mathbf{x} | G = 1)}{\Pr(\mathbf{X} = \mathbf{x} | G = 2)} = \prod_{i=1}^{p} \frac{\phi\left(\frac{X_i - \mu_{1i}}{\sigma_i}\right)}{\phi\left(\frac{X_i - \mu_{2i}}{\sigma_i}\right)}
\]

where \( \phi(\cdot) \) is the pdf of the standard normal distribution, \( \mu_{1i} \)'s and \( \mu_{2i} \)'s are components of \( \boldsymbol{\mu}_1 \) and \( \boldsymbol{\mu}_2 \), respectively, and \( \sigma_i \)'s are the standard deviations of the \( x_i \)'s. Equivalently, we have

\[
\log \Lambda = \sum_{i=1}^{p} \log \left( \frac{\phi\left(\frac{X_i - \mu_{1i}}{\sigma_i}\right)}{\phi\left(\frac{X_i - \mu_{2i}}{\sigma_i}\right)} \right) = \sum_{i=1}^{p} z_i
\]

where \( z_i = \log\left(\frac{\phi\left(\frac{X_i - \mu_{1i}}{\sigma_i}\right)}{\phi\left(\frac{X_i - \mu_{2i}}{\sigma_i}\right)}\right) \). Wald’s sequential probability ratio test (SPRT) [3] is then readily applicable to binary classification problems. Setting the desired classification error rates, we obtain the decision boundaries \( b \) and \( a(> b) \). Given a new instance \( \mathbf{x} \), we sample its components one at a time and compute \( \sum_{i=1}^{j} z_i \) until \( \sum_{i=1}^{j} z_i \) falls out of range \((b, a)\). Upon termination, we assign \( \mathbf{x} \) to class 1 if \( \sum_{i=1}^{j} z_i > a \) or to class 2 if \( \sum_{i=1}^{j} z_i < b \). Unlike many other classification algorithms, the number of components examined before making a decision is not a constant but depends on the given new instance.

The decision boundaries \( a \) and \( b \) are computed from inputs \( \alpha \) and \( \beta \), which are the desired error rates for class 1 and class 2, respectively, such that \( 0 < \alpha, \beta < 1 \). The decision boundaries are then given by

\[
a = \log\left(\frac{1 - \beta}{\alpha}\right) \quad \quad \quad b = \log\left(\frac{\beta}{1 - \alpha}\right)
\]

where the log’s are taken because \( a \) and \( b \) are the boundaries for \( \log \Lambda \) as opposed to simply \( \Lambda \).

Mukhopadhyay [3] addresses the aspect of truncation in SPRT in order to set a maximum number of components desired to be examined. Even if a constant less than \( p \) is not specified, truncation must be utilized if the algorithm has not made a decision after examining the last component. That is, \( b < \sum_{i=1}^{p} z_i < a \). In this case, the decision boundary is truncated to \( \frac{1}{2}(a + b) \), such that the given instance \( \mathbf{x} \) is assigned to class 1 if \( \sum_{i=1}^{k} z_i > \frac{1}{2}(a + b) \), or to class 2 if \( \sum_{i=1}^{k} z_i < \frac{1}{2}(a + b) \). The same truncation can be applied at any specified constant \( k \) such that \( 1 < k \leq p \) and \( b < \sum_{i=1}^{k} z_i < a \). However, parameters \( a \), \( b \), and \( k \) are somewhat interdependent.
Mukhopadhyay [3] acknowledges that if \( k \) is specified, the class error rates (which determine \( a \) and \( b \)) may not be closely met. Since we were not particularly concerned with minimizing the running time of our algorithms, we simply truncated at the total number of components \( p \) each time.

### 2.3 MSPRT

Using Fu’s [2] modified SPRT method, the computation of \( \log \Lambda \) is the same as that of SPRT. However, the decision boundaries are not constant with each iteration of observing the components of instance \( x \). The decision boundaries at the \( i \)th component examined are \( b_i \) and \( a_i(>b_i) \) given by

\[
\begin{align*}
    a_i &= a \left(1 - \frac{i}{k}\right)^{r_2} \\
    b_i &= b \left(1 - \frac{i}{k}\right)^{r_1}
\end{align*}
\]

where \( 0 < r_2, r_1 \leq 1, a > 0, b > 0 \), and \( k \) is the truncation specification. When running our algorithms, we simply set \( r_1 = r_2 = 1 \) each time. As \( i \to k \), \( a_i, b_i \to 0 \), thus ensuring that a decision will be made by the \( k \)th component. MSPRT is generally distinct from SPRT such that its decision boundaries are gradually decreasing to become 0 at iteration \( k \), while the decision boundaries of SPRT remain constant, with an abrupt decision made at iteration \( k \).

### 2.4 Component Sampling

All components are considered when predicting classification with the LDA function, while SPRT and MSPRT may use one or all of the components. This proposes the question of how to order the components when sampling in SPRT and MSPRT. We performed the functions with the components sorted in two different ways: randomly and ordered by decreasing variance. Since performing the functions with the components in random order does not return pre-determined results, we calculated the results for SPRT and MSPRT random by taking the majority prediction over 100 runs for each \( \alpha, \beta \) tested.

The classification algorithms described above assume that the components of the instance to be classified are independent. Practically, however, this is usually not the case. In order to obtain an independent set, we apply a linear transformation to the components, which we can assume follow a multivariate normal distribution. The linear transformation is generated by first calculating the corresponding eigenvalues and eigenvectors of the covariance matrix of the training data. The transformation is then the set of eigenvectors whose eigenvalues are greater than \( 1/1000^{th} \) of the sum of eigenvalues. This linear transformation is applied to each instance before it is classified.
2.5 Leave-One-Out Cross-Validation

We performed leave-one-out cross-validation with the datasets in order to determine the accuracy of each function. The process consists of excluding the first instance of the dataset while using the rest of the instances as training data. The excluded instance is then predicted using that training data on the function being tested. This is repeated for each instance, each time comparing the predicted class label with the actual, pre-known label. Finally, the accuracy of the function is calculated as the percentage of instances that were classified correctly.

3 Results and Discussion

We ran our programs on three different binary biological datasets. The first was a Parkinson’s disease dataset [4,5], having 195 instances with 22 components. The second was a colon cancer dataset [6] preprocessed by [7], having 63 instances with 2000 components. We ranked the genes of the colon cancer dataset by using a simple index (BSS/WSS) as described in [8], narrowing the dataset down to only 500 components. The third dataset was for breast cancer [9], having 683 instances with 10 components. The results of our algorithms run on each biological dataset are shown below in Figures 1, 2, and 3, respectively.

![Graph](image)

**Fig. 1.** Accuracies for Parkinson's disease dataset. SPRT ordered and MSPRT ordered were run with $\alpha = \beta$ from 0.01 to 0.40 with a step size of 0.01. The accuracies for SPRT random and MSPRT random were calculated by the majority predictions over 100 runs each at $\alpha = \beta = 0.175$ and $\alpha = \beta = 0.32$. The highest accuracy for the Parkinson’s disease dataset was 0.7128, generated by the MSPRT random function at $\alpha = \beta = 0.175$ and by the SPRT random function at $\alpha = \beta = 0.32$. 
Fig. 2. Accuracies for the colon cancer dataset. SPRT ordered and MSPRT ordered were run with $\alpha = \beta$ from 0.01 to 0.40 with a step size of 0.01. The accuracies for SPRT random and MSPRT random were calculated by the majority predictions over 100 runs each at $\alpha = \beta = 0.12$ and $\alpha = \beta = 0.22$. The highest accuracy for the colon cancer dataset was 0.8871, generated by the MSPRT ordered function at $\alpha = \beta = 0.22$. 
Fig. 3. Accuracies for breast cancer dataset. SPRT ordered and MSPRT ordered were run with $\alpha = \beta$ from 0.01 to 0.40 with a step size of 0.01. The accuracies for SPRT random and MSPRT random were calculated by the majority predictions over 100 runs each at $\alpha = \beta = 0.07$ and $\alpha = \beta = 0.09$. The highest accuracy for the breast cancer dataset was 0.9648, generated by the MSPRT ordered function at $\alpha = \beta = 0.07$ and by the SPRT ordered function at $\alpha = \beta = 0.09$.

In the Colon Cancer dataset, MSPRT ordered reached the maximum accuracy rate in the set, while SPRT ordered was just under its accuracy rate. However, in the case of MSPRT and SPRT random, they only reached the LDA accuracy rate and MSPRT random fell below the accuracy rating of LDA.

Finally, in the Breast Cancer dataset, the accuracy of SPRT and MSPRT ordered both reached peaks in accuracy above the LDA accuracy rate at different $\alpha$ and $\beta$ values, but both reached the same peak in accuracy. SPRT and MSPRT random both fell short of the LDA accuracy rate, MSPRT random getting lower accuracies both times the tests were run.

The results of the datasets might suggest that in sets that have high accuracy using the LDA method, have a greater accuracy rate from SPRT and MSPRT ordered. In the Parkinson’s dataset, the set where we had the lowest accuracy, our SPRT and MSPRT rates were fairly low compared to the accuracy found in the other datasets. Another conclusion that can possibly be rendered is that in datasets with low accuracy MSPRT random and SPRT random have higher accuracy than a standard LDA.

Another thing that can be noticed in the datasets is that SPRT and MSPRT both random and ordered may have some correlation. In lower accuracy datasets such as the Parkinson’s set MSPRT ordered has a lower accuracy rate than SPRT ordered, however in higher to mid-high accuracy datasets such as the breast and colon cancer datasets, MSPRT ordered has somewhat higher accuracy rates than SPRT ordered. However with MSPRT and SPRT random, with high to mid-high accuracy datasets,
the accuracy of MSPRT random seems to always be lower than the SPRT random. It’s difficult to make any conclusion off of this further point however, as the Parkinson’s dataset is hard to glean any results off of.

4 Conclusion

A future project might be warranted to suggest what high accuracy is using datasets that have varying accuracy rates, as we were only able to test three datasets. The more data sets that can be tested the more accurate estimates we can get of the correlation between SPRT, MSPRT both ordered and random and the accuracy rates of LDA.

Another future project that might be of some interest would be looking into the accuracy of various other datasets, also increasing the number of tests for the random datasets may prove to have different results. The more trials ran of the different datasets, the closer we can get to a ‘true’ value.

The conclusion that can be learned from all of this though is that accuracy rates of various testing methods is largely dependent on the dataset that you are using and the accuracy rate and correlation between status and other variables.

References