Computer Project: Project Report for the Application of Pattern Recognition with Genomic Expression Data Set

Che-Hao Hu* and Zhiyang Ong†

Abstract

This report commences with a brief description of the pattern recognition framework and techniques for predicting the survival rates of cancer patients. The main stages for our implementation of a pattern recognition system are: feature selection, classifier design, and error estimation. Next, it lists the experimental results that have been obtained, and includes a discussion of these experimental results.

1 Introduction

A pattern recognition system (see Figure 1) is implemented for predicting the survival rates of cancer patients. The pattern recognition uses a genomic expression data set to classify patients into 2 classes: patients would would survive and those who would not survive. The data set is obtained from a microarray-based cancer classification study [4].

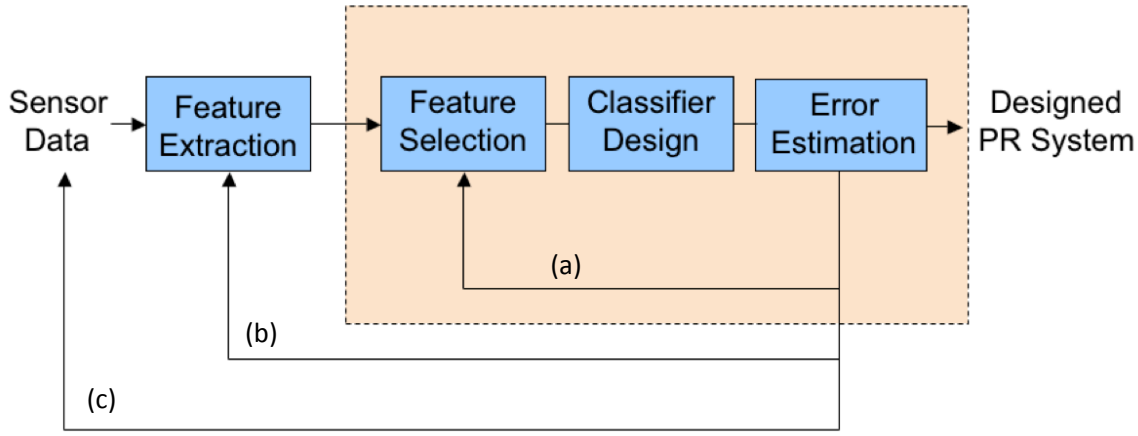

Figure 1: Pattern recognition [1, Presentation slides for Lecture 1, "Introduction", slide 13 on "Pattern Recognition Design Cycle"] [5, §1.1.2, pp. 3, Figure 1.1]

The main stages for our implementation of a pattern recognition system are: feature selection, classifier design, and error estimation. In particular, the classification rules that are implemented include:

*Email correspondence to: pickup0430@cycu.org.tw
†Email correspondence to: ongz@acm.org

diagonal linear discriminant analysis (LDA), 3-Nearest Neighbor algorithm (3NN), and linear support vector machine (SVM). In addition, two types of error estimators are implemented: resubstitution and leave-one-out cross-validation. Also, the feature selection procedure is implemented with exhaustive search and with sequential forward search. Most of the combinations of these techniques for this sequence of 3 stages are linear, and do not involve any feedback from the error estimation stage to the feature selection stage. However, for techniques used by the sequential forward search procedure, data from the error estimation stage needs to be provided to the feature selection stage (see feedback path (a) in Figure 1). Feedback paths (b) and (c) in Figure 1 are not implemented for this pattern recognition system, since we do not carry out any feature extraction nor data acquisition from sensors.

In this report, the following terms are used interchangeably. Gene is used synonymously with features. Also, labels, classes, and groups (denotement of labels/classes in MATLAB [2]) are used synonymously. In addition, patients and samples are used interchangeably.

The goal of the project is to classify the cancer patients correctly patients correctly, in terms of whether they will survive breast cancer.

This report is organized as follows. Section 2 describes how each of the main stages of pattern recognition in the pattern recognition process shown in Figure 1 is implemented. For each stage, a description will be provide for each technique that is used to implement that stage. A list of techniques for each stage is described in the project handout, and are required for grading of the project. Next, in Section 3, a discussion of the experimental results and analysis of the experimental data is carried out. Finally, Section 4 ends the report with some final remarks.

2 Description of Implementation Details

The pattern recognition shown in Figure 1 is implemented in MATLAB [2]. Here, in this section, the implementation of each of the three stages of pattern recognition are described.

The genomic expression data set is randomly partitioned into two sets: training data and testing data. The former set is used to train the classifiers for classifying the data, while the testing data is used to determine the hold-out estimate of the true classification error.

2.1 Feature Selection

The feature selection stage is implemented with the following two methods: sequential forward search (see §2.1.1) and exhaustive search (see §2.1.2). Let the total number of genes in the gene signature be denoted as n, where $n = 70$. Let the number of genes/features to be chosen be denoted by r.

2.1.1 Sequential Forward Search

For this feature selection method, $r = 1$–5. The algorithm for sequential forward search is described by Procedure 2.1.1.

We have discovered that the function `sequentialfs` is part of the Statistics Toolbox's "Exploratory Data Analysis: Cluster Analysis: Hierarchical Clustering" feature. We can use this for sequential forward search. However, due to time constraints and the efficiency in which we have implemented the sequential

forward search algorithm with code reuse, we are not exploring the usage of the `sequentialfs` function for this class project.

SEQUENTIAL FORWARD SEARCH(α, n, r)

// Input α: 2-D data set which rows are samples and columns are genes.
// Input n: Total number of features in the data set.
// Input r: Number of features to be selected.

// Set of selected features.
1 $F = \varnothing$

// Select the first feature.
// For each of the n features
2 **for** $i = 1$ **to** n
3 Design a classifier c_i for feature i, f_i.
4 Estimate the error for c_i.
5 Pick feature f_1 among these features such that its classifier c_i' for f_i has the minimum estimated error for all c_i.
// Add f_1 to the set of selected features.
6 $F = F \cup f_1$

// Select the second feature.
// For each of the remaining $(n-1)$ features
7 **for** $j = 1$ **to** $(n-1)$
// Design a classifier for (f_1, f_j).
8 Design a classifier c_j for feature j, f_j, and the features in F.
9 Estimate the error for c_j.
10 Pick feature f_2 among these features such that its classifier c_j' for (f_1, f_j) has the minimum estimated error for all c_j.
// Add f_2 to the set of selected features.
11 $F = F \cup f_2$

// ...
// Select the r^{th} feature.
// For each of the remaining $(n-r+1)$ features
12 **for** $m = 1$ **to** $(n-r+1)$
// Design a classifier for $(f_1, f_2, \ldots, f_{m-1}, f_m)$.
13 Design a classifier c_m for feature m, f_m, and the features in F.
14 Estimate the error for c_m.
15 Pick feature f_r among these features such that its classifier c_m' for $(f_1, f_2, f_3, \ldots, f_m)$ has the minimum estimated error for all c_m.
// Add f_r to the set of selected features.
16 $F = F \cup f_r$
17 **return** The combination $(f_1, f_2, f_3, \ldots, f_r)$ as the selected feature set.

2.1.2 Exhaustive Search

For this feature selection method, $r = 1$–3. This is carried out by enumerating each combination of r genes from the C_r^n combinations of genes. Since exhaustive search requires a classifier with the minimum estimated error for all these combinations, a classifier is designed for each combination of 3 genes. That is, the combination of features (1, 2, 3) has a unique classifier. Likewise, the combination of features (1, 2, 4) has another associated unique classifier. Subsequently, the best classifier is selected from all designed classifiers. Next, the best classifier is used to classify the testing data. Finally, the error estimation would be carried out on the testing data to determine an estimate for the true classification error.

Therefore, the process of applying exhaustive search would result in exploring C_r^n combinations of features.

2.2 Classifier Design

The classification rules that are implemented include: diagonal linear discriminant analysis (LDA) (see §2.2.1), 3-Nearest Neighbor algorithm (3NN) (see §2.2.2), and linear support vector machine (SVM) (see §2.2.3).

2.2.1 Diagonal Linear Discriminant Analysis (LDA)

The MATLAB function `classify(testing/sample data, training data, training data label, type)` is used to implement the diagonal Linear Discriminant Analysis (LDA) classifier.

2.2.2 3-Nearest Neighbor Algorithm (3NN)

The following MATLAB functions are used to implement the 3-nearest neighbor algorithm (3NN).

The MATLAB function `KNN = ClassificationKNN.fit(X,Y,'PARAM1',val1)`, where `X` is the training data, `Y` is the set of labels for the training data, `PARAM1` refers to 'NumNeighbors' and value `val1` = 3, <'NumNeighbors', 3> is the <PARAM, value> pair that is used to indicate the value of "k" in our 3-NN (or k-NN, for k = 3) algorithm implementation.

The MATLAB function `label = predict(mdl,Xnew)` returns a vector of predicted class labels for a matrix `Xnew`, based on mdl, a model returned by the `ClassificationKNN.fit()` function that was customized for the 3NN algorithm.

2.2.3 Linear Support Vector Machine (SVM)

Two MATLAB functions from the Statistics Toolbox [3] are used to implement the linear support vector machine (SVM) classifier. They are: `svmtrain` – train a support vector machine classifier; and `svmclassify` – classify data using a support vector machine.

2.3 Error Estimation

Two types of error estimators are implemented: resubstitution (see §2.3.1) and leave-one-out cross-validation (see §2.3.2).

2.3.1 Resubstitution

To implement the resubstitution method of error estimation, we use the entire set of training data for designing a classifier for each combination of selected features. Subsequently, use the entire set of training data to calculate the estimated error of each classifier.

2.3.2 Leave-One-Out Cross-Validation

The leave-one-out cross-validation method is described in Procedure 2.3.2.

LEAVE-ONE-OUT CROSS-VALIDATION(n)

// Input n: Number of patients in the data set.

// For each of the 60 patients/samples.

1 **for** $i = 1$ **to** 60
2 Design a classifier for the other 59 patients.
3 Use the aforementioned classifier to label the i^{th} patient.
4 Estimate the classification error for label the i^{th} patient.
5 **return**

3 Experimental Results

The experimental results are tabulated in Figures 2, 3, and 4. They are also plotted in Figures 5, 6, 7, 8, 9, 10, 11, 12.

Classification Rule	Feature Selection	Error Estimators	Selected Feature	Estimated Error	Estimat of the True Classification Error
3NN	Exhaustive search (1)	Resubstitution	66	0.1	0.2255
3NN	Exhaustive search (2)	Resubstitution	7, 66	0.0333	0.2128
3NN	Exhaustive search (3)	Resubstitution	1, 8, 49	0.0167	0.3277
3NN	Exhaustive search (1)	Leave-One-Out	49	0.1833	0.2255
3NN	Exhaustive search (2)	Leave-One-Out	7, 66	0.1	0.2128
3NN	Exhaustive search (3)	Leave-One-Out	2, 37, 64	0.0833	0.2426
3NN	Sequential Forward Search (1)	Resubstitution	66	0.1	0.2255
3NN	Sequential Forward Search (2)	Resubstitution	66, 22	0.05	0.234
3NN	Sequential Forward Search (3)	Resubstitution	66, 22, 5	0.0667	0.234
3NN	Sequential Forward Search (4)	Resubstitution	66, 22, 5, 24	0.0333	0.2638
3NN	Sequential Forward Search (5)	Resubstitution	66, 22, 5, 24, 35	0.0333	0.234
3NN	Sequential Forward Search (1)	Leave-One-Out	49	0.1833	0.2255
3NN	Sequential Forward Search (2)	Leave-One-Out	49, 37	0.1167	0.1957
3NN	Sequential Forward Search (3)	Leave-One-Out	49, 37, 24	0.1167	0.2
3NN	Sequential Forward Search (4)	Leave-One-Out	49, 37, 24, 8	0.0833	0.2383
3NN	Sequential Forward Search (5)	Leave-One-Out	49, 37, 24, 8, 5	0.0833	0.2426

Figure 2: Experimental results for the 3-Nearest Neighbor (3NN) classifier and its associated combinations of feature selection (exhaustive search and sequential forward search) and error estimation (resubstitution and leave-one-out cross-validation) techniques. It also includes the set of selected features, and the hold-out estimate of the true classification error.

Classification Rule	Feature Selection	Error Estimators	Selected Feature	Estimated Error	Estimat of the True Classification Error
Diagonal LDA	Exhaustive search (1)	Resubstitution	12	0.1333	0.2809
Diagonal LDA	Exhaustive search (2)	Resubstitution	57, 66	0.1	0.1915
Diagonal LDA	Exhaustive search (3)	Resubstitution	12, 20, 33	0.0667	0.2511
Diagonal LDA	Exhaustive search (1)	Leave-One-Out	12	0.1667	0.2809
Diagonal LDA	Exhaustive search (2)	Leave-One-Out	57, 66	0.1	0.1915
Diagonal LDA	Exhaustive search (3)	Leave-One-Out	2, 60, 66	0.0833	0.1787
Diagonal LDA	Sequential Forward Search (1)	Resubstitution	12	0.1333	0.2809
Diagonal LDA	Sequential Forward Search (2)	Resubstitution	12, 20	0.1333	0.2681
Diagonal LDA	Sequential Forward Search (3)	Resubstitution	12, 20, 33	0.0667	0.2511
Diagonal LDA	Sequential Forward Search (4)	Resubstitution	12, 20, 33, 16	0.0833	0.2468
Diagonal LDA	Sequential Forward Search (5)	Resubstitution	12, 20, 33, 16 , 38	0.0833	0.2128
Diagonal LDA	Sequential Forward Search (1)	Leave-One-Out	12	0.1667	0.2809
Diagonal LDA	Sequential Forward Search (2)	Leave-One-Out	12, 18	0.15	0.2894
Diagonal LDA	Sequential Forward Search (3)	Leave-One-Out	12, 18, 24	0.1167	0.2511
Diagonal LDA	Sequential Forward Search (4)	Leave-One-Out	12, 18, 24 ,23	0.1	0.2596
Diagonal LDA	Sequential Forward Search (5)	Leave-One-Out	12, 18, 24, 23, 13	0.0833	0.2468

Figure 3: Experimental results for the linear support vector machine (SVM) classifier and its associated combinations of feature selection (exhaustive search and sequential forward search) and error estimation (resubstitution and leave-one-out cross-validation) techniques. It also includes the set of selected features, and the hold-out estimate of the true classification error.

4 Conclusions

4.1 Comparison of the different classification rules based on the combination of selected genes and its corresponding hold-out estimate of the true classification error

For diagonal LDA, the minimum hold-out estimate of the true classification error is 0.1787 when we use exhaustive search, Leave-One-Out, and choose the 2^{nd}, 60^{th}, and 66^{th} genes/features.

For 3NN, the minimum hold-out estimate of the true classification error is 0.1957 when we use sequential forward search, Leave-One-Out, and choose the 49^{th}, and 37^{th} genes/features.

For linear SVM, the minimum hold-out estimate of the true classification error is 0.1319 when we use sequential forward search, resubstitution, and choose the 49^{th}, 18^{th}, 17^{th}, and 60^{th} genes/features.

4.2 For each classification, comparison of feature selection methods and error estimation methods

For the diagonal LDA, we draw the following conclusions. If the error estimator is the same, exhaustive search can perform better than sequential forward search as the number of selected features increase. For exhaustive search, leave-one-out can perform better than resubstitution for sufficiently large sets of selected features. For sequential forward search, resubstitution can perform better than leave-one-out for sufficiently large sets of selected features.

For the 3NN, we draw the following conclusions. For leave-one-out error estimation, sequential for-

Classification Rule	Feature Selection	Error Estimators	Selected Feature	Estimated Error	Estimat of the True Classification Error
Linear SVM	Exhaustive search (1)	Resubstitution	49	0.1333	0.1745
Linear SVM	Exhaustive search (2)	Resubstitution	60, 66	0.0667	0.1702
Linear SVM	Exhaustive search (3)	Resubstitution	13, 57, 66	0.0333	0.1787
Linear SVM	Exhaustive search (1)	Leave-One-Out	12	0.1333	0.2766
Linear SVM	Exhaustive search (2)	Leave-One-Out	60, 66	0.0833	0.1702
Linear SVM	Exhaustive search (3)	Leave-One-Out	Calculating	Calculating	Calculating
Linear SVM	Sequential Forward Search (1)	Resubstitution	49	0.1333	0.1745
Linear SVM	Sequential Forward Search (2)	Resubstitution	49, 18	0.1167	0.1745
Linear SVM	Sequential Forward Search (3)	Resubstitution	49, 18 ,17	0.1167	0.1745
Linear SVM	Sequential Forward Search (4)	Resubstitution	49, 18 ,17 ,60	0.1	0.1319
Linear SVM	Sequential Forward Search (5)	Resubstitution	49, 18 ,17 ,60, 3	0.0833	0.1702
Linear SVM	Sequential Forward Search (1)	Leave-One-Out	12	0.1333	0.2766
Linear SVM	Sequential Forward Search (2)	Leave-One-Out	12, 20	0.1167	0.2468
Linear SVM	Sequential Forward Search (3)	Leave-One-Out	12, 20, 16	0.0833	0.2511
Linear SVM	Sequential Forward Search (4)	Leave-One-Out	12, 20, 16, 37	0.0667	0.2383
Linear SVM	Sequential Forward Search (5)	Leave-One-Out	12, 20, 16, 37, 7	0.05	0.1915

Figure 4: Experimental results for the diagonal linear discriminant analysis (LDA) classifier and its associated combinations of feature selection (exhaustive search and sequential forward search) and error estimation (resubstitution and leave-one-out cross-validation) techniques. It also includes the set of selected features, and the hold-out estimate of the true classification error.

ward search performs better than exhaustive search as the number of selected features increase. For resubstitution error estimation, we cannot conclude if sequential forward search or exhaustive search is better. For exhaustive search, leave-one-out can perform better than resubstitution as the number of selected features increase. For sequential forward search, leave-one-out can perform better than resubstitution as the number of selected features increase from 1 to 4.

For linear SVM, we draw the following conclusions. For resubstitution, regardless of the feature selection procedure, the hold-out estimate of the true classification error does not change significantly as the number of selected features increase. For leave-one-out, exhaustive search can perform better than sequential forward search as the number of selected features increase. For exhaustive search, resubstitution performs better than leave-one-out. For sequential forward search, resubstitution performs much better than leave-one-out, especially for small sets of selected features.

4.3 Discussion of how experimental results would change if more training samples or genes were available

If more training samples or genes were available, the results for diagonal LDA would not change much since diagonal LDA is not data hungry [1, Lecture 5 on "Parametric Classification Rules", slides 11, 14, and 17]. Likewise, for 3NN, its results would not change much either, since it is not data hungry [1, Lecture 14 on "Model Selection", slide 50].

However, for linear SVM , which is a data hungry method, the results change significantly. Consequently, dimensionality reduction is needed to reduce 70 genes to 60 (at least), since we only have 60 samples/patients in the data set.

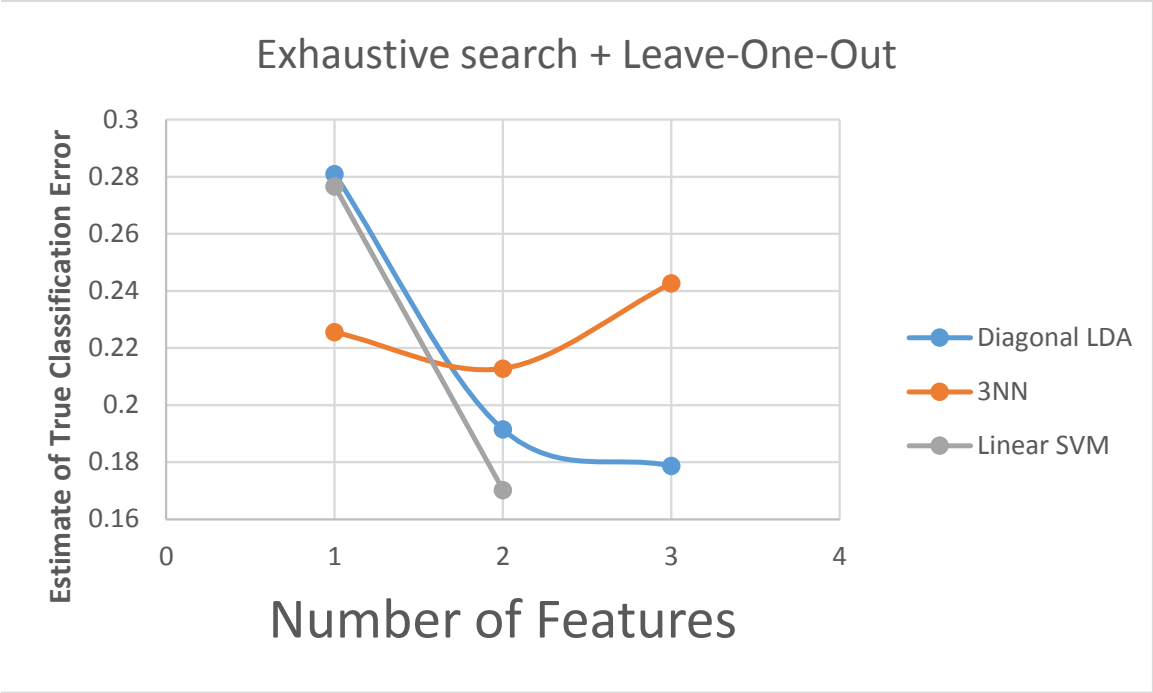

Figure 5: Comparison of classifiers for the combination of exhaustive search-based feature selection and leave-one-out-based error estimation, with regards to the hold-out estimate of the true classification error.

References

[1] Ulisses de Mendonça Braga-Neto. ECEN 649 Pattern Recognition: Course Material. Published online via eLearning, 2014.

[2] The MathWorks, Inc. MATLAB. Available online at: http://www.mathworks.com/products/matlab/; April 2, 2014 was the last accessed date, 2014.

[3] The MathWorks, Inc. Statistics toolbox. Available online at: http://www.mathworks.com/products/statistics/; April 2, 2014 was the last accessed date, 2014.

[4] Marc J. van de Vijver, Yudong D. He, Laura J. van 't Veer, Hongyue Dai, Augustinus A. M. Hart, Dorien W. Voskuil, George J. Schreiber, Johannes L. Peterse, Chris Roberts, Matthew J. Marton, Mark Parrish, Douwe Atsma, Anke Witteveen, Annuska Glas, Leonie Delahaye, Tony van der Velde, Harry Bartelink, Sjoerd Rodenhuis, Emiel T. Rutgers, Stephen H. Friend, , and René Bernards. A gene-expression signature as a predictor of survival in breast cancer. New England Journal of Medicine, 347(25):1999–2009, December 19 2002.

[5] Andrew R. Webb and Keith D. Copsey. Statistical Pattern Recognition. John Wiley & Sons, Chichester, West Sussex, England, U.K., third edition, 2011.

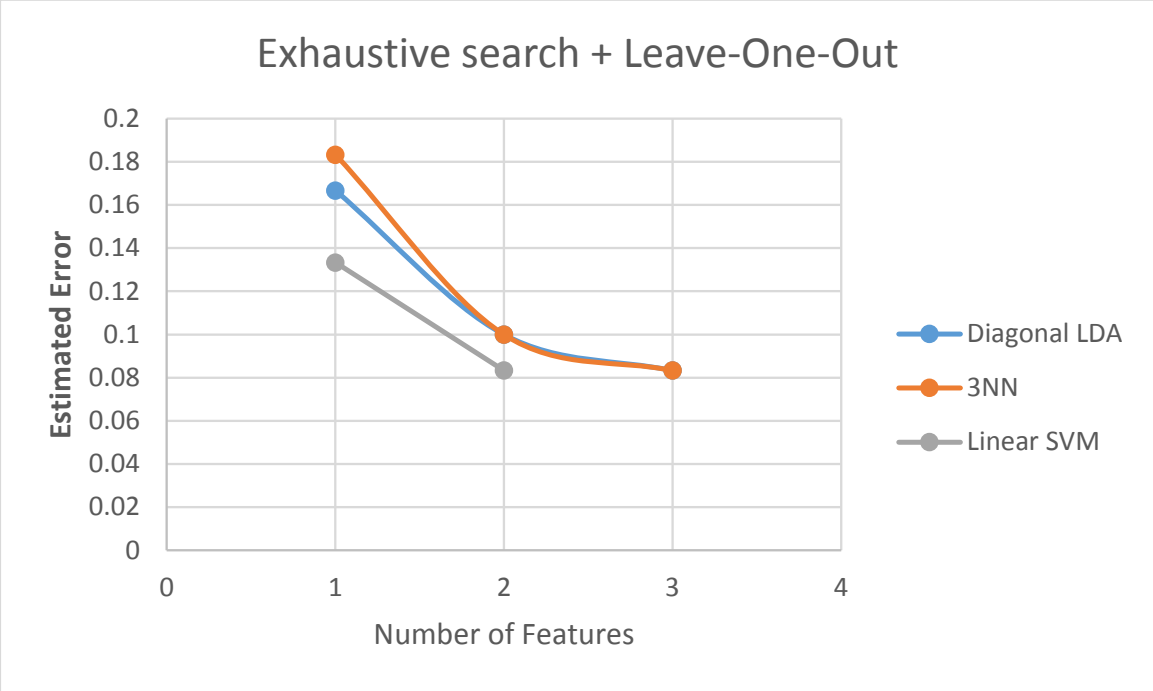

Figure 6: Comparison of classifiers for the combination of exhaustive search-based feature selection and leave-one-out-based error estimation, with regards to the estimated error.

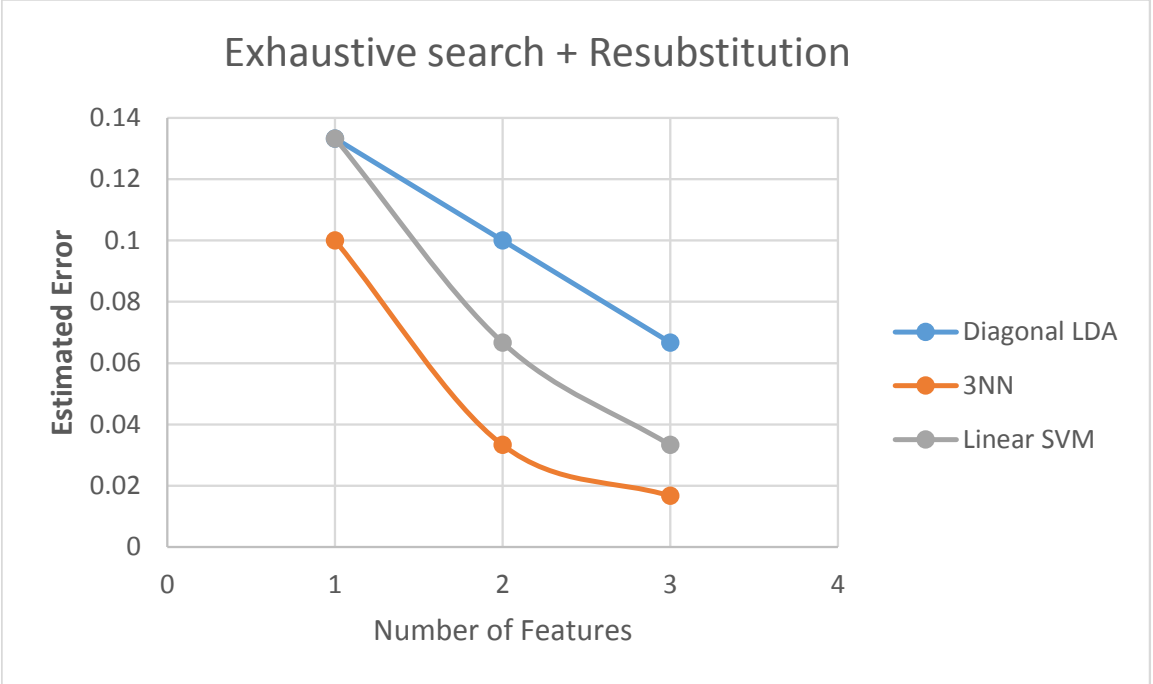

Figure 7: Comparison of classifiers for the combination of exhaustive search-based feature selection and resubstitution-based error estimation, with regards to the hold-out estimate of the true classification error.

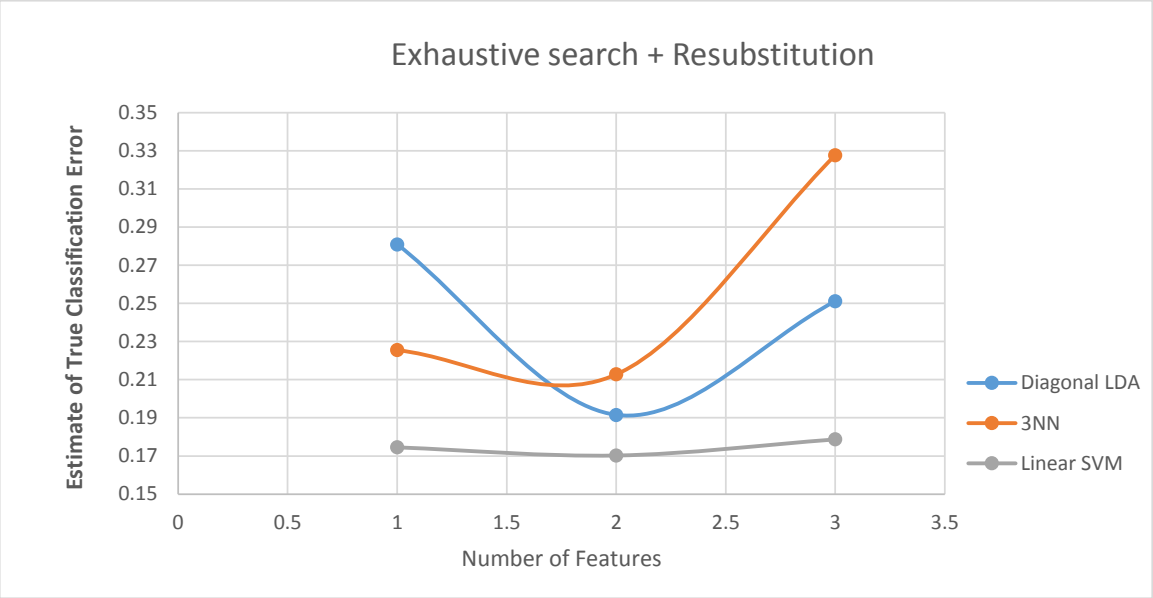

Figure 8: Comparison of classifiers for the combination of exhaustive search-based feature selection and resubstitution-based error estimation, with regards to the estimated error.

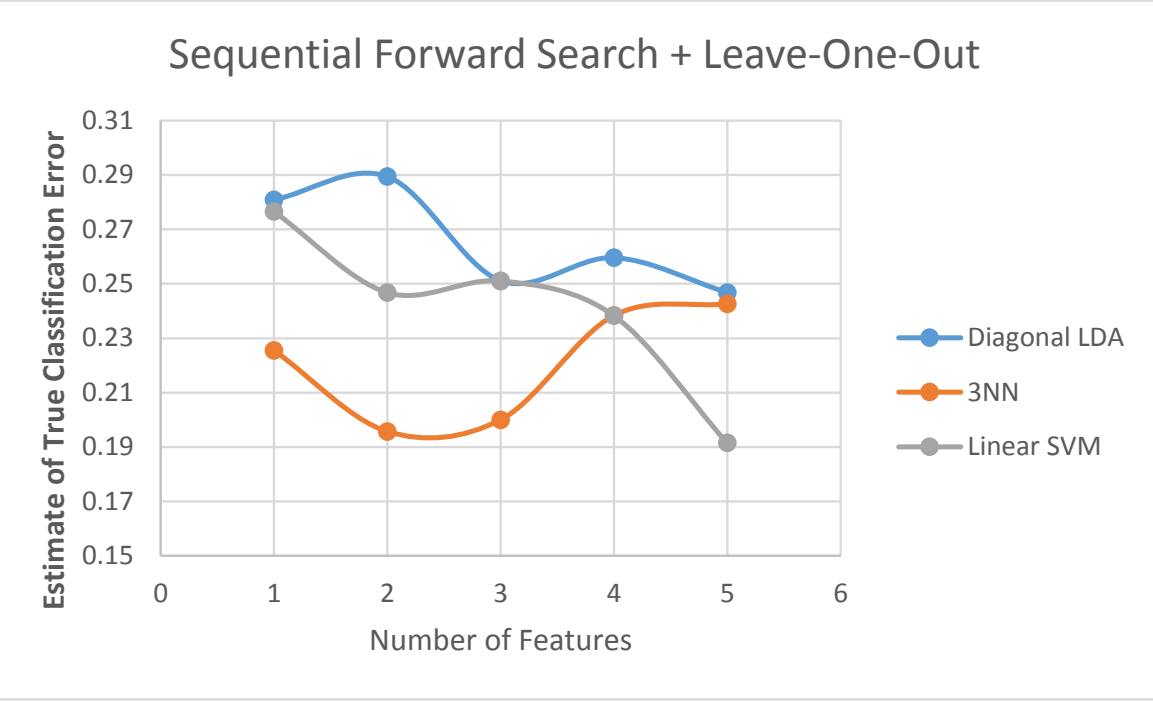

Figure 9: Comparison of classifiers for the combination of sequential forward search-based feature selection and leave-one-out-based error estimation, with regards to the hold-out estimate of the true classification error.

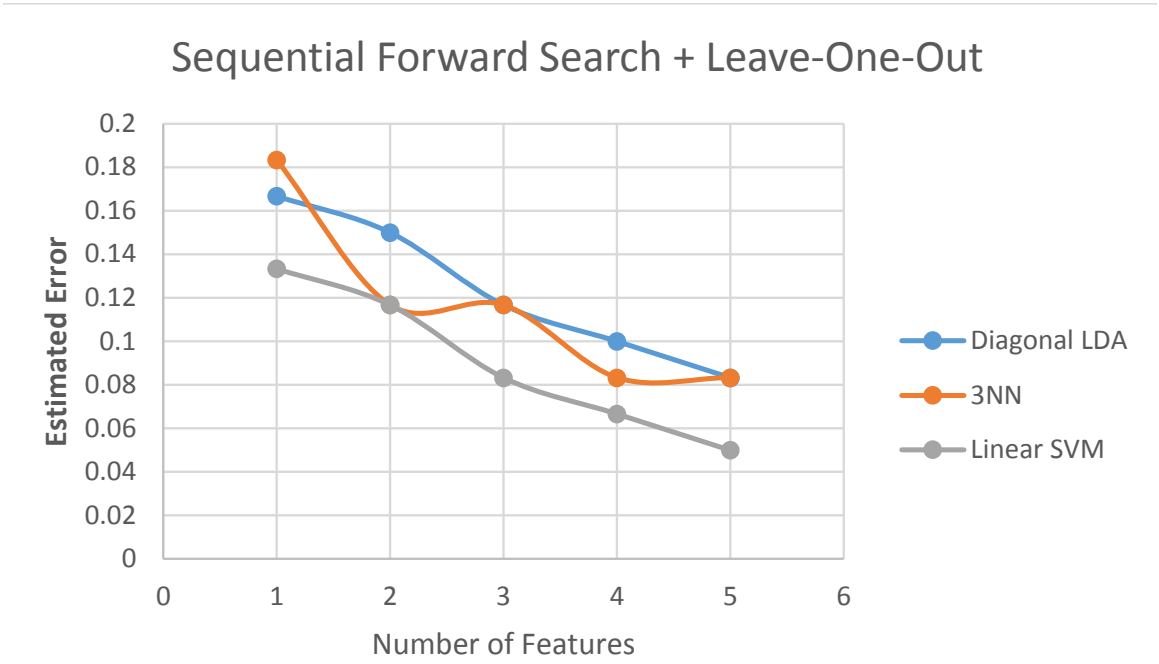

Figure 10: Comparison of classifiers for the combination of sequential forward search-based feature selection and leave-one-out-based error estimation, with regards to the estimated error.

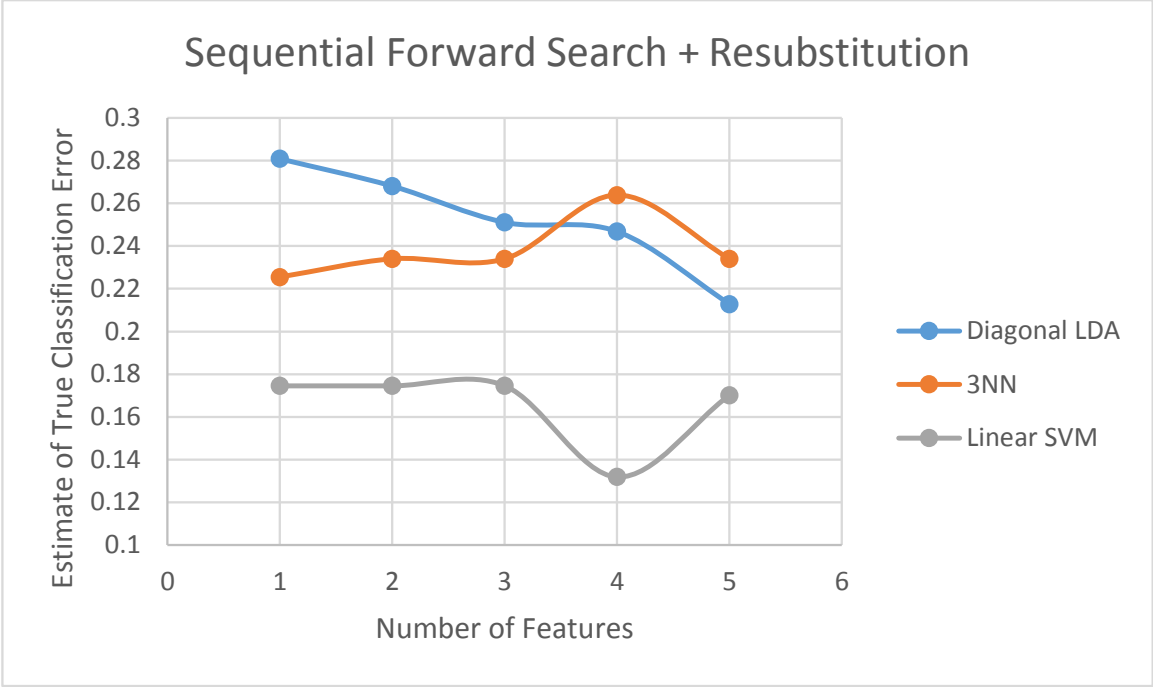

Figure 11: Comparison of classifiers for the combination of sequential forward search-based feature selection and resubstitution-based error estimation, with regards to the hold-out estimate of the true classification error.

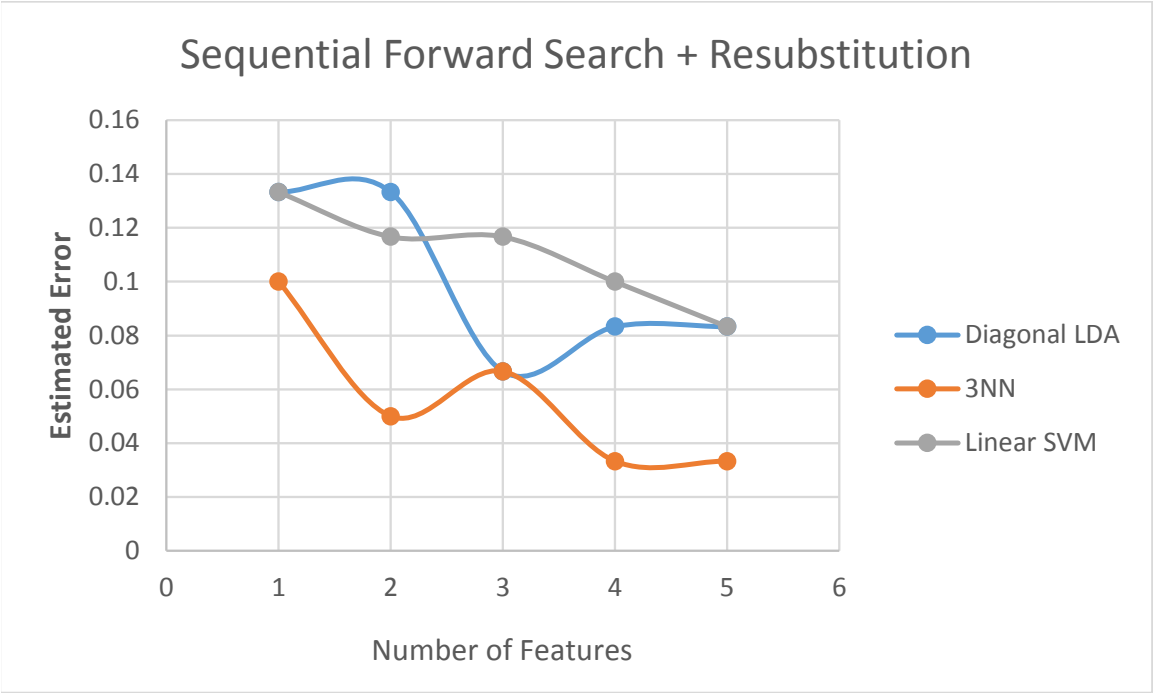

Figure 12: Comparison of classifiers for the combination of sequential forward search-based feature selection and resubstitution-based error estimation, with regards to the estimated error.