Classifier evaluation and the use of algorithmic classifiers with expert system classifiers

Erach. A. Irani.

February 1993

Contents

1 Introduction **1**

1.1 Classifiers and Expert System Classifiers 1
1.2 Expert Systems 3
1.3 Algorithmic classifiers, expert system classifiers, and experts 3
1.4 Performance evaluation for classifiers 5
1.5 Combining classifier assessments to improve performance 6
1.6 A word on notation used in the thesis 7
1.7 Organization of the thesis 7

2 Criteria for evaluating classifiers **8**

2.0.1 Notation, agreement, and association 8
2.0.2 Criteria for measuring agreement or association 9
2.0.2.1 Proportionate reduction of error (PRE) measures or κ . . . 12
2.0.2.2 Brennan's κ and $\kappa_{b,w}$ 13
2.1 Summary 18

3 Expert and classifier evaluation of POSCH atherosclerosis data **19**

3.1 The expert system, ESCA 21
3.2 Multiple linear regression 30
3.3 Back-propagation 30
3.4 C4 35
3.5 Summary 38

4 Estimating the value of a criterion and its standard error **41**

4.1 Evaluation considering the influence of the number of cases 41
4.2 Theoretically derived estimates for a function of random variables 43
4.3 The resubstitution method, the holdout method, and cross-validation 44
4.4 Obtaining performance estimates using the grouped jackknife 46
4.5 The Bootstrap algorithm to estimate the characteristics of the sampling distribution of a statistic 46
4.5.1 How large should B be? 48
4.5.2 Inaccuracy in computing Expected Excess Error by the bootstrap . 51
4.6 Estimating performance of a trainable classifier 55
4.6.1 Using repeated random halving to estimate the mean and the standard error (RRH). 56
4.6.2 Splitting and recombining the cases 57

4.6.3 Evaluating statistics over a training set using the split recombination procedure . . . 57
4.6.4 The split multinomial method (SM) . . . 58
4.6.5 The split grouped jackknife method (SGJK) . . . 58
4.6.6 The split bootstrap method (SB) . . . 59
4.7 Simulation to show the plausibility of the SM, SGJK, and SB methods . . . 59
4.7.1 The *diff dataset* and the *diff methods* . . . 59
4.7.2 Procedure used in the simulation . . . 60
4.7.3 Determining the number of times the true value falls in the estimated confidence interval about the estimated value . . . 62
4.7.4 p-values for the simulation results . . . 65
4.7.5 Stem and leaf diagrams for the simulation. . . . 69
4.7.6 Scatterplots to show the results of the simulations . . . 88
4.7.6.1 Scatterplots of $\hat{L}$ . . . 88
4.7.6.2 Scatterplots of $\hat{\kappa}_{b,w}$ . . . 95
4.7.6.3 Plots of $SE(L)$ . . . 100
4.7.6.4 Plots of $SE(\kappa_{b,w})$ . . . 105
4.7.7 Choosing among the method of repeated random halving, the SB method, and the SGJK method . . . 116
4.8 Summary . . . 119

5 Improving performance by combining classifiers 121
5.1 Cartesian product of classes or partitions . . . 121
5.2 The Classifier-based Overall Loss Minimization Strategy (COLMS) . . . 122
5.3 Adaptations to the COLMS . . . 124
5.3.1 Overall Loss Minimization Strategy (OLMS) . . . 125
5.3.2 Alternative class-selection strategies for class combination . . . 126
5.4 An Expert-defined Domain-Specific combination Strategy (EDSS) for the coronary angiography application . . . 127
5.5 Rejection and rejection loss . . . 130
5.6 Summary . . . 131

6 Using perturbation-testing for a classifier 133
6.1 Perturbation-testing in the AEPOSCH application . . . 134
6.1.1 Procedure for making changes in arteries . . . 135
6.1.2 Procedure to Perturb Slightly Low (PSL) . . . 136
6.1.3 Procedure to Perturb Slightly High (PSH) . . . 137
6.1.4 Procedure to Perturb Low (PL) . . . 138
6.1.5 Procedure to Perturb High (PH) . . . 140
6.2 Testing for generalization . . . 141
6.3 Summary . . . 141

7 Experiments over the AEPOSCH database 143
7.1 Atherosclerosis data in POSCH . . . 143
7.2 Estimation of L and $\kappa_{b,w}$ over the atherosclerosis data . . . 147
7.2.1 Cross-classification tables for methods over the B3 dataset . . . 150
7.2.2 Cross-classification tables for methods over the BR dataset . . . 159

7.3 Summary figures for the B3 and BR datasets 164
7.4 Summary statistics for the B3 and BR datasets 168
7.5 Interpretation of performance evaluation results 185
7.6 Results of perturbation-testing 185
7.6.1 Expert opinion on perturbation results and interpretation of statistical results 194
7.6.2 Experts' opinion whether MLR is usable instead of ESCA 194
7.7 Summary 195

8 Summary and Future Work **197**

List of Tables

2.1 Table showing perfect agreement and perfect association. 9
2.2 Table showing no agreement and perfect association. 10
2.3 Hypertension probabilities of mis-classification. 17
2.4 p_{ij}/π_i for hypertension. 17
2.5 p_{ij}/p_j for hypertension. 17
2.6 Hypertension probabilities of mis-classification for a homogeneous classifier. 17
2.7 Hypertension probabilities p_{ij} of mis-classification. 17

3.1 Scale for experts' opinion of overall change in ESCA. 21
3.2 Arterial segments with their distal arterial segments. 27
3.3 Mapping of change in stenosis to individual artery points in ESCA. 28
3.4 Mapping of total-points to tentative-assessment in ESCA. 29
3.5 Mapping of variable numbers to arterial segments in multiple linear regression, back-propagation, and C4. 39
3.6 An example training set for C4. 40

4.1 Table showing simulation configurations for testing the SGJK and the SB methods. 62
4.2 Fraction of times the true value lies within the within the 90% and 95% C.I estimated by different methods. $\mu_1 = 1, n = 630, h = 1$ 64
4.3 Fraction of times the true value lies within the 90% and 95% C.I estimated by different methods. $\mu_1 = 1, n = 50, h = 1$ 64
4.4 Fraction of times the true value lies within the within the 90% and 95% C.I estimated by different methods. $\mu_1 = 2.5, n = 630, h = 21$ 64
4.5 Fraction of times the true value lies within the within the 90% and 95% C.I estimated by different methods. $\mu_1 = 2.5, n = 50, h = 1$ 65
4.6 Fraction of times the true value lies within the within the 90% and 95% C.I estimated by different methods. $\mu_1 = 4.0, n = 630, h = 21$ 65
4.7 Fraction of times the true value lies within the within the 90% and 95% C.I estimated by different methods. $\mu_1 = 4.0, n = 50, h = 1$ 66
4.8 Bonferroni values for differences of $SE(\kappa_{b,w})$ estimated using different methods. $R = 200, \mu_1 = 1, n = 50, h = 1$ 66
4.9 Bonferroni values for differences of $SE(\kappa_{b,w})$ estimated using different methods. $R = 200, \mu_1 = 4, n = 50, h = 5$ 67
4.10 Bonferroni values for differences of $SE(\kappa_{b,w})$ estimated using different methods. $R = 200, n = 630, \mu_1 = 1, h = 1$ 67

4.11 Bonferroni values for differences of $SE(\kappa_{b,w})$ estimated using different methods. $R = 200, \mu_1 = 1, n = 630, h = 21, B = 200$. 67
4.12 Probabilities for $X = \chi_F/\sqrt{F}$, $F = 199$. 69

6.1 Points v_{ij} for Perturb Slightly Low in the AEPOSCH application. 137
6.2 Stenoses in arterial segments before and after perturbing using PSL. 137
6.3 Points v_{ij} for Perturb Slightly High in the AEPOSCH application. 138
6.4 Points v_{ij} for Perturb Low in the AEPOSCH application. 139
6.5 Points v_{ij} for Perturb High in the AEPOSCH application. 140

7.1 Multiple linear regression coefficients fitted by least squares when training on entire B3 dataset 145
7.2 Upper-triangular cross-classification table of B3-QA dataset. 146
7.3 Upper-triangular cross-classification table of BR-QA dataset. 147
7.4 Loss values ($\times 100$) for ungrouped assessments in the AEPOSCH application. 148
7.5 Grouping scheme for atherosclerosis evaluation. 148
7.6 Loss values ($\times 100$) for grouped assessments in the AEPOSCH application. . 149
7.7 Cross-classification of experts' consensus opinion and ESCA's assessments for the B3 dataset. 152
7.8 Cross-classification of experts' grouped (consensus) opinion and ESCA's grouped assessments for the B3 dataset. 152
7.9 Bonferroni p-value for arterial differences on the B3 dataset. 153
7.10 Bonferroni p-value for arterial differences on the BR dataset. 154
7.11 Cross-classification of experts' assessment (gold standard) and ESCA's assessment for the B3 dataset. 155
7.12 Cross-classification of experts' assessment (gold standard) and MLR's assessment for the B3 dataset. 155
7.13 Cross-classification of experts' assessment (gold standard) and BP's assessment for the B3 dataset. 156
7.14 Cross-classification of experts' assessment (gold standard) and C4's assessment for the B3 dataset. 156
7.15 Cross-classification of experts' assessment (gold standard) and COLMS's assessment for the B3 dataset. 157
7.16 Cross-classification of experts' assessment (gold standard) and EDSS's unrejected assessment for the B3 dataset. 157
7.17 Cross-classification of experts' assessment (gold standard) and EDSS's rejected assessment for the B3 dataset. 158
7.18 Cross-classification of experts' assessment (gold standard) and ESCA's assessment for the BR dataset. 159
7.19 Cross-classification of experts' assessment (gold standard) and MLR's assessment for BR dataset. 160
7.20 Cross-classification of experts' assessment (gold standard) and BP's assessment for BR dataset. 160
7.21 Cross-classification of experts' assessment (gold standard) and C4's assessment for the BR dataset. 160
7.22 Cross-classification of experts' assessment (gold standard) and COLMS's assessment for the BR dataset. 161

7.23 Cross-classification of experts' assessment (gold standard) and EDSS's for BR dataset for unrejected cases. 161
7.24 Cross-classification of experts' assessment (gold standard) and EDSS's for BR dataset for rejected cases. 162
7.25 Cross-classification of experts' grouped assessment (gold standard) and ESCA's grouped assessment for the BR dataset. 163
7.26 Multinomial calculation of $\hat{L}, SE(L)$, and $\hat{\kappa}_{b,w}$ for ungrouped assessments over the B3-QA and BR-QA datasets compared with ESCA's $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ for B3 and BR datasets. 169
7.27 Multinomial calculation of $\hat{L}, SE(L)$, and $\hat{\kappa}_{b,w}$ for grouped assessments over the B3-QA and BR-QA datasets compared with ESCA's $L, SE(L)$, and $\hat{\kappa}_{b,w}$ for B3 and BR datasets. 169
7.28 $\hat{L}$ and $\hat{\kappa}_{b,w}$ computed for ESCA using the bootstrap with 200 bootstrap replications over the B3 dataset. 170
7.29 $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ computed using the SM technique for classifiers over the B3 dataset. 171
7.30 L and $\kappa_{b,w}$ estimated for the training set using the RRH technique over the B3 dataset. 171
7.31 L and $\kappa_{b,w}$ estimated using the RRH technique over the B3 dataset. 172
7.32 L and $\kappa_{b,w}$ estimated using the SGJK method over the B3 dataset. 173
7.33 L and $\kappa_{b,w}$ estimated using the SB method over the B3 dataset. 173
7.34 L and $\kappa_{b,w}$ for grouped assessments estimated using the RRH technique over the B3 dataset. 175
7.35 L and $\kappa_{b,w}$ for grouped assessments estimated using the SGJK technique over the B3 dataset. 175
7.36 L and $\kappa_{b,w}$ for grouped assessments estimated using the SB method over the B3 dataset. 175
7.37 L and $\kappa_{b,w}$ for grouped, randomized assessments estimated using the SGJK technique over the B3 dataset. 176
7.38 L and $\kappa_{b,w}$ for grouped, randomized assessments estimated using the SB method over the B3 dataset. 176
7.39 L and $\kappa_{b,w}$ estimated using the bootstrap over the BR dataset. 177
7.40 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ computed using the bootstrap over the BR dataset. 177
7.41 L and $\kappa_{b,w}$ estimated using the bootstrap over the BR dataset for grouped assessments. 178
7.42 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ computed using the bootstrap over the BR dataset for grouped assessments. 178
7.43 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ computed using the RRH technique over the B3 dataset. 178
7.44 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ computed using the SGJK technique over the B3 dataset. 179
7.45 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ computed using the SB method for the B3 dataset. 179
7.46 L and $\kappa_{b,w}$ for randomized assessments estimated using the SGJK technique over the B3 dataset. 180

7.47 L and $\kappa_{b,w}$ for randomized assessments estimated using the SB technique over the B3 dataset. 180
7.48 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for randomized assessments computed using the SGJK technique over the B3 dataset. 181
7.49 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers computed using the SB method for randomized assessments in the B3 dataset. 181
7.50 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for grouped assessments computed using the RRH technique over the B3 dataset. 183
7.51 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for grouped assessments computed using the SGJK technique over the B3 dataset. . . . 183
7.52 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for grouped assessments computed using the SB method over the B3 dataset. 183
7.53 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for grouped, randomized assessments computed using the SGJK technique over the B3 dataset. 184
7.54 Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for grouped, randomized assessments computed using the SB technique over the B3 dataset.184
7.55 Results of Perturb Slightly Low (PSL) – ESCA. 186
7.56 Results for Perturb Slightly High (PSH) – ESCA. 187
7.57 Results for Perturb Low (PL) – ESCA. 187
7.58 Results for Perturb High (PH) – ESCA. 188
7.59 Results for Perturb Slightly Low (PSL) – MLR. 188
7.60 Results for Perturb Slightly High (PSH) – MLR. 189
7.61 Results for Perturb Low (PL) – MLR. 189
7.62 Results for Perturb High (PH) – MLR. 190
7.63 Results for Perturb Slightly Low (PSL) – BP. 190
7.64 Results for Perturb Slightly High (PSH) – BP. 190
7.65 Results for Perturb Low (PL) – BP. 191
7.66 Results for Perturb High (PH) – BP. 191
7.67 Results for Perturb Slightly Low (PSL) – C4. 192
7.68 Results for Perturb Slightly High (PSH) – C4. 192
7.69 Results for Perturb Low (PL) – C4. 193
7.70 Results for Perturb High (PH) – C4. 193
7.71 Perturbation scores $\hat{V}$ and $SE(V)$ for classifiers on the B3 dataset. 194
7.72 Bonferroni values for differences in perturbation scores $\hat{V}$ on the B3 dataset. 195

List of Figures

1.1 Taxonomy of classifiers. 2

3.1 Independent Film A Form. 22
3.2 Consensus Film A Form. 23
3.3 Independent Film B Form. 24
3.4 Consensus Film B Form. 25
3.5 M14 Form. 26
3.6 A simple unit in back-propagation. 31
3.7 A back-propagation network with one hidden layer and two units in the output layer. 33
3.8 $\hat{L}$ on test cases vs. number of iterations for some promising configurations for back-propagation networks using 1, 2, and 3 hidden layers. 35
3.9 Example decision tree in C4. 37

4.1 $CV(\hat{\sigma}_B)$ vs. B for kurtosis = 0. 49
4.2 $CV(\hat{\sigma}_B)$ vs. B for kurtosis = 1. 50
4.3 $CV(\hat{\sigma}_B)$ vs. B for kurtosis = 2. 50
4.4 $CV(\hat{\sigma}_B)$ vs. B for kurtosis = 4. 51
4.5 Effect of over-training on classification space. 68
4.6 Stem and leaf plot of $\hat{L}(multi-diff\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff} = 0.3198$. n=200, N=50, h=1. 72
4.7 Stem and leaf plot of $\hat{L}(SM\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff} = 0.3198$. n=200, N=50, h=1. 72
4.8 Stem and leaf plot of $\hat{L}(multi-diff\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff} = .3122$. n=200, N=50, h=5. 73
4.9 Stem and leaf plot of $\hat{L}(SM\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff} = .3122$. n=200, N=50, h=5. 73
4.10 Stem and leaf plot of $\hat{L}(multi-diff\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff} = 0.0228$. n=200, N=50, h=1, $\mu_1 = 4$. 74
4.11 Stem and leaf plot of $\hat{L}(SM\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff} = 0.0228$. n=200, N=50, h=1, $\mu_1 = 4$. 74
4.12 Stem and leaf plot of $SE(L, multi-diff\ ,r)/SE(L_{diff})$. $SE(L_{diff}) = .0661$. n=200, N=50, h=1. 75
4.13 Stem and leaf plot of $SE(L, SM\ ,r)/SE(L_{diff})$. $SE(L_{diff}) = .0661$. n=200, N=50, h=1. 75
4.14 Stem and leaf plot of $SE(L, multi-diff\ ,r)/SE(L_{diff})$. $SE(L_{diff}) = .0646$. n=200, N=50, h=5. 76
4.15 Stem and leaf plot of $SE(L, SM\ ,r)/SE(L_{diff})$. $SE(L_{diff}) = .0646$. n=200, N=50, h=5. 76

4.16 Stem and leaf plot of $SE(L, multi-diff\ , r)/SE(L_{diff})$. $SE(L_{diff})$ = .0208. n=200, N=50, h=1. μ_1 = 4.0. 77
4.17 Stem and leaf plot of $SE(L, SM\ , r)/SE(L_{diff})$. $SE(L_{diff})$ = .0208. n=200, N=50, h=1. μ_1 = 4.0. 77
4.18 Stem and leaf plot of $\hat{\kappa}_{b,w}(multi-diff\ , r)/\hat{\kappa}_{b,w,diff}$. $\hat{\kappa}_{b,w,diff}$ = .17523. n=200, N=50, h=1. 78
4.19 Stem and leaf plot of $\hat{\kappa}_{b,w}(SM\ , r)/\hat{\kappa}_{b,w,diff}$. $\hat{\kappa}_{b,w,diff}$ = .17523. n=200, N=50, h=1. 79
4.20 Stem and leaf plot of $\hat{\kappa}_{b,w}(multi-diff\ )/\hat{\kappa}_{b,w,diff}$. $\hat{\kappa}_{b,w,diff}$ = .1752 n=200, N=50, h=5. 80
4.21 Stem and leaf plot of $\hat{\kappa}_{b,w}(SM\ , r)/\hat{\kappa}_{b,w,diff}$. $\hat{\kappa}_{b,w,diff}$ = .1752 n=200, N=50, h=5. 80
4.22 Stem and leaf plot of $SE(\kappa_{b,w}, bst-diff\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff})$ = .2291. n=200, N=50, h=1, B=200. 81
4.23 Stem and leaf plot of $SE(\kappa_{b,w}, SB\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff})$ = .2291. n=200, N=50, h=1, B=200. 82
4.24 Stem and leaf plot of $SE(\kappa_{b,w}, gjk-diff\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff})$ = .2291. n=200, N=50, h=1. 83
4.25 Stem and leaf plot of $SE(\kappa_{b,w}, SGJK\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff})$ = .2291. n=200, N=50, h=1. 84
4.26 Stem and leaf plot of $SE(\kappa_{b,w}, bst-diff\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff})$ = .2439. n=200, N=50, h=5, B=200. 85
4.27 Stem and leaf plot of $SE(\kappa_{b,w}, SB\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff})$ = .2439. n=200, N=50, h=5, B=200. 86
4.28 Stem and leaf plot of $SE(\kappa_{b,w}, gjk-diff\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff})$ = .2439. n=200, N=50, h=50. 86
4.29 Stem and leaf plot of $SE(\kappa_{b,w}, SGJK\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff})$ = .2439. n=200, N=50, h=5. 87
4.30 Scatterplot of $\hat{L}(multi-split)$ vs. $\hat{L}(multi-diff)$. n = 50, h = 1, μ_1 = 1. 89
4.31 Scatterplot of $\hat{L}(multi-split)$ vs. $\hat{L}(multi-diff)$. n = 50, h = 5, μ_1 = 1. 90
4.32 Scatterplot of $\hat{L}(multi-split)$ vs. $\hat{L}(multi-diff)$. n = 50, h = 1, μ_1 = 4. 91
4.33 Scatterplot of $\hat{L}(multi-split)$ vs. $\hat{L}(multi-diff)$. n = 50, h = 5, μ_1 = 4. 92
4.34 Scatterplot of $\hat{L}(multi-split)$ vs. $\hat{L}(multi-diff)$. n = 630 , h = 1, μ_1 = 4. 93
4.35 Scatterplot of $\hat{L}(multi-split)$ vs. $\hat{L}(multi-diff)$. n = 630 , h = 21, μ_1 = 4. 94
4.36 Scatterplot of $\hat{\kappa}_{b,w}(multi-split)$ vs. $\hat{\kappa}_{b,w}(multi-diff)$. n = 50, h = 1, μ_1 = 1. 95
4.37 Scatterplot of $\hat{\kappa}_{b,w}(multi-split)$ vs. $\hat{\kappa}_{b,w}(multi-diff)$. n = 50, h = 5, μ_1 = 1. 96
4.38 Scatterplot of $\hat{\kappa}_{b,w}(multi-split)$ vs. $\hat{\kappa}_{b,w}(multi-diff)$. n = 50, h = 1, μ_1 = 4. 97
4.39 Scatterplot of $\hat{\kappa}_{b,w}(multi-split)$ vs. $\hat{\kappa}_{b,w}(multi-diff)$. n = 50, h = 5, μ_1 = 4. 98
4.40 Scatterplot of $\hat{\kappa}_{b,w}(multi-split)$ vs. $\hat{\kappa}_{b,w}(multi-diff)$. n = 630 , h = 1, μ_1 = 4. 99
4.41 Scatterplot of $SE(L, multi-split)$ vs. $SE(L, multi-diff)$. n = 50, h = 1, μ_1 = 1. 100
4.42 $SE(L, multi-split)$ vs. $SE(L, multi-diff)$. n = 50, h = 1, μ_1 = 1. Random sample two. 101
4.43 Scatterplot of $SE(L, multi-split)$ vs. $SE(L, multi-diff)$. n = 50, h = 5, μ_1 = 1. 102

4.44 Scatterplot of $SE(L, multi-split)$ vs. $SE(L, multi-diff)$. n = 630 , h = 1, $\mu_1 = 1$. 103
4.45 Scatterplot of $SE(L, multi-split)$ vs. $SE(L, multi-diff)$. n = 630 , h = 21, $\mu_1 = 1$. 104
4.46 Scatterplot of $SE(\kappa_{b,w}, bst-diff)$ vs. $SE(\kappa_{b,w}, gjk-diff)$. n = 50, h = 5, B = 200, $\mu_1 = 1$. 105
4.47 Scatterplot of $SE(\kappa_{b,w}, bst-diff)$ vs. $SE(\kappa_{b,w}, gjk-diff)$. n = 50, h = 1, B = 1000, $\mu_1 = 1$. 106
4.48 Scatterplot of $SE(\kappa_{b,w}, gjk-split)$ vs. $SE(\kappa_{b,w}, gjk-diff)$. n = 50, h = 1, $\mu_1 = 1$. 107
4.49 Scatterplot of $SE(\kappa_{b,w}, gjk-split)$ vs. $SE(\kappa_{b,w}, gjk-diff)$. n = 50, h = 5, $\mu_1 = 1$. 108
4.50 Scatterplot of $SE(\kappa_{b,w}, gjk-split)$ vs. $SE(\kappa_{b,w}, gjk-diff)$. n = 50, h = 5, $h_s = 1$, $\mu_1 = 1$. 109
4.51 Scatterplot of $SE(\kappa_{b,w}, bst-split)$ vs. $SE(\kappa_{b,w}, bst-diff)$. n = 50, h = 5, B = 200, $\mu_1 = 1$. 110
4.52 Scatterplot of $SE(\kappa_{b,w}, bst-split)$ vs. $SE(\kappa_{b,w}, bst-diff)$. n = 50, h = 1, B = 1000, $\mu_1 = 1$. 111
4.53 Scatterplot of $SE(\kappa_{b,w}, gjk-split)$ vs. $SE(\kappa_{b,w}, gjk-diff)$. n = 630 , h = 1, $\mu_1 = 1$. 112
4.54 Scatterplot of $SE(\kappa_{b,w}, gjk-split)$ vs. $SE(\kappa_{b,w}, gjk-diff)$. n = 630 , h = 21, $h_s = 1$, $\mu_1 = 1$. 113
4.55 Scatterplot of $SE(\kappa_{b,w}, gjk-split)$ vs. $SE(\kappa_{b,w}, gjk-diff)$. n = 630 , h = 21, $\mu_1 = 1$. 114
4.56 Scatterplot of $SE(\kappa_{b,w}, bst-split)$ vs. $SE(\kappa_{b,w}, bst-diff)$. n = 630 , h = 1, $\mu_1 = 1, B = 200$. 115
4.57 Scatterplot of $SE(\kappa_{b,w}, bst-split)$ vs. $SE(\kappa_{b,w}, bst-diff)$. n = 630 , h = 1, $\mu_1 = 1, B = 1000$. 116
4.58 Scatterplot of $SE(\kappa_{b,w}, bst-split)$ vs. $SE(\kappa_{b,w}, bst-diff)$. n = 630 , h = 21, $\mu_1 = 1$, B = 200. 117
4.59 Scatterplot of $SE(\kappa_{b,w}, bst-split)$ vs. $SE(\kappa_{b,w}, bst-diff)$. n = 630 , h = 21, $\mu_1 = 1$, B = 1000 118

7.1 $\hat{L}$ and $SE(L)$ for various classifiers computed over the B3 dataset. 164
7.2 $\hat{L}$ and $SE(L)$ for various classifiers computed over the B3 dataset. The assessments are grouped. 165
7.3 $\hat{\kappa}_{b,w}$ and $SE(\kappa_{b,w})$ for various classifiers computed over the B3 dataset. . . . 166
7.4 $\hat{\kappa}_{b,w}$ and $SE(\kappa_{b,w})$ for various classifiers computed over the B3 dataset. The assessments are grouped. 166
7.5 $\hat{L}$ and $SE(L)$ computed using the bootstrap for classifiers over the BR dataset. 167
7.6 $\hat{\kappa}_{b,w}$ and $SE(\kappa_{b,w})$ computed using the bootstrap for classifiers over the BR dataset. 167
7.7 $\hat{L}$ and $SE(L)$ computed using the bootstrap for classifiers over the BR dataset. The assessments are grouped. 168
7.8 $\hat{\kappa}_{b,w}$ and $SE(\kappa_{b,w})$ computed using the bootstrap for classifiers over the BR dataset. The assessments are grouped. 169

Abstract

Classification is an important task in the real-world. It is important to develop techniques for the comparison of classifiers. This has to be done statistically and in terms of well-defined criteria. This is a major goal of this thesis. This thesis discusses such techniques specifically in the context of comparing *algorithmic classifiers (algorithms)* and *expert system classifiers (expert systems)*. The basic criterion used in this thesis is L, the weighted loss of mis-classification. L is a well-established criterion in the statistical and pattern-recognition literature. A relatively unknown criterion by Brennan, kappa base (κ_b) that adjusts for agreement attainable due to chance, is introduced from the biomedical and psychological literature. κ_b is extended to $\kappa_{b,w}$ for the case when the mis-classifications are of varying degrees of severity. The importance of using statistical techniques such as hypothesis testing and confidence intervals is stressed.

The use of cross-validation, the grouped jackknife, and the bootstrap for estimating the value, standard error, and other properties of a criterion is shown. A deficiency in the bootstrap method for estimating Expected Excess Error of a trainable classifier is proved. The split-recombination (SR) conjecture is introduced to estimate properties of a trainable classifier and three new techniques, the split multinomial (SM), split grouped jackknife (SGJK), and the split bootstrap (SB), are introduced. A simple experiment on artificial data consisting of two classes from two uni-variate normal populations is done to demonstrate the plausibility of the SR conjecture.

The Classifier-based Overall Loss Minimization Strategy (COLMS) and the Overall Loss Minimization Strategy (OLMS) are introduced to combine classifier evaluations to improve L and $\kappa_{b,w}$. The COLMS is an independent discovery and development of the *coupling procedure* for classifiers introduced by Wernecke. The OLMS extends the ideas in the COLMS to accommodate arbitrary partitioning algorithms. These ideas are demonstrated on real-world atherosclerosis evaluation data from the Program On the Surgical Control of the Hyperlipidemias (POSCH) and four real-world classifiers viz. an expert system ESCA, multiple linear regression, back-propagation, and C4.5 (an implementation of ID3). An Expert Defined Domain-Specific Strategy (EDSS) for combining classifiers is compared with the COLMS.

The use of an expert-defined test (EDT) is introduced to further increase confidence in a classifier, especially an algorithmic classifier. The EDT examined was based on perturbation testing. The domain experts agreed after viewing the results of the rigid statistical evaluation and the performance on the EDT of ESCA and multiple linear regression that multiple linear regression could be used instead of ESCA for the subset of the POSCH atherosclerosis evaluation task that ESCA performed.

Chapter 1

Introduction

A classifier does the task of classifying samples from a population into one of several subpopulations, that is, into one of several categories or classes. An *expert system classifier (expert system)* is a computer program making use of domain-specific knowledge in a manner intended to mimic the reasoning process a knowledgeable expert would use. An *algorithmic classifier (algorithm)* is a classifier based on a non-domain-specific computational algorithm such as multiple linear regression. The algorithmic classifier is not domain-specific but follows a generic, pre-defined classification strategy based on its underlying mathematical model. The model is independent of the task under consideration. This thesis studies techniques to compare classifiers with specific application to algorithmic classifiers and expert system classifiers. This is because, while algorithmic classifiers can often be cheaper to implement than an expert system, it is often hard to convince domain-experts that an algorithmic classifier can do the classification task at hand – this was an important motivation in the development of expert systems.

1.1 Classifiers and Expert System Classifiers

Classification is a frequently performed task in the real world. It is practised every day by people, machines, and animals. For instance, a person has to classify an apple as good or bad to eat, machines can sort parts on a conveyor belt, and an animal has to classify another animal as dangerous or friendly. A taxonomy of classifiers is given in Fig. 1.1. A human being may be an expert in a particular field. Being an expert requires expertise that is gained only after study and experience in that field [81, 35]. Doctors are a well-known example of experts in the field of medicine. Expert systems are one attempt to capture the expertise of human experts in a form where it can be readily available [30, 35]. An expert system that carries out the task of classification normally requiring an expert is an expert system classifier. The importance of classification can be recognized from the fact that one of the grandfathers of all expert systems, MYCIN, has classification as an integral part of its function [68, 67, 66]. Other classificatory expert systems are ETA [49, 70] and ESCA [48, 50]. ETA and ESCA have been developed by the Program On the Surgical Control of the Hyperlipidemias (POSCH) to assess clinical trial data by classifying individual patient data.

Figure 1.1: Taxonomy of classifiers.

1.2 Expert Systems

Expert systems are increasingly popular tools of the information age. A typical expert system is an interactive computer program based on the knowledge of experts for performing a task such as classifying an object, analyzing data, or designing mechanisms in engineering. There are thousands of commercial applications of expert systems. Many Fortune 500 companies are conducting research and development involving expert systems. This effort is motivated by a desire to capture knowledge in a usable form and distribute it. The potential savings of using an expert system are enormous. An equally important motivation is to capture the scarce expertise of an expert before he/she retires or leaves the company.

The early expert systems were implemented directly in higher-level programming languages such as INTERLISP [68]. Expert system shells were developed later to speed up the implementation process and provide generic help and other facilities. This is possible because an expert system shell allows domain-specific knowledge to be expressed in a domain-independent framework. The cost of developing an expert system can be reduced by using expert system shells [30, 81, 35].

Examples of academic expert system shells are EMYCIN [68, 81], the shell derived from MYCIN, and AGNESS [72], the shell used to develop ETA [71, 70]. KEE [41] is an example of a commercial knowledge engineering environment that, besides being an expert-system-building tool, provides several facilities such as graphical-interface-building capabilities [40, 39]. A moderately complicated expert system may require six person-years of effort to create, whereas the more complex ones require a large expenditure of effort, sometimes as many as 15-30 person-years [81].

Validation and performance evaluation of expert systems is an important issue. The importance of statistical tests of performance to consider the effects of sample size is stressed in [57, 58] when discussing the validation of expert systems. Validation and performance evaluation are defined by O'Leary et al. [59] as two distinct issues. Validation is the process of determining that an expert system accurately represents an expert's knowledge in a particular problem domain. Performance evaluation is the process of examining an expert system's ability to solve real-world problems in a particular problem domain. This requires the use of one or more techniques to measure performance. It is observed in [3] that most expert systems are inadequately validated. MYCIN [68] is an example of a system that has been carefully validated. Even so, the performance is not adequately reported in all articles describing MYCIN. For example, it is reported that in 75% of cases MYCIN's recommended course of treatment is judged adequate but the number of cases used is not reported [69]. The number of cases used is important because it enables us to determine the influence of random fluctuations on the proportion of acceptable treatments observed. Statements such as "...once the experts approve its performance in at least 90% of cases" appearing in [67] do not take into account the random fluctuations taking place due to chance.

1.3 Algorithmic classifiers, expert system classifiers, and experts

In contrast to the high cost of building an expert system, the cost of building an algorithmic classifier is usually substantially lower because an algorithmic classifier, unlike an expert system classifier, does not require careful handcrafting of knowledge for its development.

Numerous studies comparing clinical versus statistical prediction in a variety of clinical contexts have found that statistical models have outperformed or performed as well as the clinicians [47, 13, 38, 33, 12]. Viewing the evidence, one researcher suggests that acquisition of expert knowledge should be limited to the identification of key process parameters for subsequent use in relatively simple models [55, 64].

Algorithmic methods tend to do as well or better than the experts for several applications. For instance, in evaluations of the expert system ETA, multiple linear regression, and human doctors; the human doctors, multiple linear regression, are comparable[1] [71]. Artificial Neural Nets and statistical discriminant analysis have been used to classify the nature of back pain and perform well in comparison with human experts [52]. It is observed in [76] that the accuracy of radiologists diagnosing breast cancer was increased by using statistically based decision aids. It is observed in [22] (a reprint of [21]) that several other studies [80, 43, 51] have observed that departures from the principles of normative decision theory [64] are a regular feature of human decision-making, even among experts.

Despite these facts, algorithmic classifiers have tended to be neglected in expert systems research. This is because experts tend to distrust the algorithmic classifiers, especially if they cannot understand the algorithm. This distrust can be overcome by a rigorous evaluation of the expert system and the algorithm and showing that the algorithm does the task at an acceptable level of performance. In addition, another way of overcoming such distrust is by explaining to the expert how the algorithm works. In some cases, the algorithm will perform at the same (or better) level of performance than the expert system.

Several arguments have been advanced for not using algorithms instead of expert systems but they can be countered.

1. The expert may still be reluctant to use the algorithmic classifier because he/she cannot "trust" it.

   We can explain that in many situations, where crucial decisions may have to be taken, a human being has to make the final decision. Neither the algorithm nor the expert system is likely to be trusted to make crucial decisions especially because of legal issues, such as liability, of using an expert system [24, 85]. In cases where the decision is not crucial, if a performance analysis shows that the algorithm performs acceptably there is no reason not to accept the algorithm.

2. Algorithmic classifiers do not provide help facilities like an expert system.

   This is incorrect as help facilities can be constructed around several algorithms such as linear discriminant analysis [56] or multiple linear regression [15, 56] that illustrate how the algorithm's decision is made.

3. The expert system follows the same reasoning strategy as used by the expert and is therefore better than an algorithm.

[1] Evaluations for this application are given on a seven-point scale (-3, -2, -1, 0, +1, +2, +3) with -3 being a code for *much worse* and +3 being a code for *much better*. Two assessments are *within one* if they occupy adjacent categories on the scale such as 0 and +1. Human doctors agreed exactly with each other in 42% of 42 cases and agreed within one category in 76% of the cases. Multiple linear regression agreed exactly with human doctors in 34% of 100 cases and agreed within one category in 81.5% of 100 cases. ETA agreed exactly with human doctors in 42% of 100 cases and agreed within one category in 84% of 100 cases.

It can be pointed out that in fact, an algorithm may be more understandable, as it follows a well-defined approach for training and for arriving at assessments. The expert system, on the other hand, is often constructed using an ad-hoc approach. In creating a successful expert system, we are carving out a set of structures with narrowed focus to represent reality [10]. Besides, expert systems need not follow the reasoning process used by the expert, who may use intuition and recognition [11].

Classifier evaluation is important because it may enable the researcher to answer the questions below.

1. Can a classifier perform well enough for the required task, where performance may be based on any well-defined criterion ?
2. Is one classifier better than another for the given task?
3. Has a scheme to combine two classifiers resulted in a classifier with better performance than any one of the constituent classifiers?

The above questions have to be answered when deciding whether an algorithmic classifier or an expert system classifier can be used, and whether combining two classifiers makes sense. Answering these questions is important so that one can increase acceptance of an expert system or algorithmic classifier. Better performance evaluation is especially important for gaining acceptance of an algorithmic classifier because algorithmic classifiers are distrusted by experts.

1.4 Performance evaluation for classifiers

The concepts of standard error and confidence intervals from statistics can be used to express the effects of uncertainty in an estimate [14]. Typically, for a classifier to be acceptable its actual performance must exceed some given threshold τ. Either confidence intervals or hypothesis testing can be used to compare its estimated performance with the minimum level. The statistical concept of a p-value is used in hypothesis testing.

The p-value is the probability, computed under the assumption that a null hypothesis H_0 is true, that a test-statistic value at least as extreme as the one observed would occur by chance. If the p-value is sufficiently small (typical threshold values are .10, .05, and .01), the null hypothesis is discredited in favor of the alternative hypothesis. Equivalently, the p-value is the smallest significance level at which the null hypothesis can be rejected [14].

For example, if the criterion is the probability π that a classifier gives a result acceptable to the expert, we can express the uncertainty in whether the performance is satisfied in terms of a p-value of a test of a null hypothesis H_0 against an alternative H_a. Instead of specifying that the performance is $100\pi\%$, we specify the maximum p-value of the null hypothesis

$$H_0 : \pi \leq \tau$$

in a one-tailed hypothesis test [14] with

$$H_a : \pi > \tau$$

as the alternative hypothesis, where τ is the minimum acceptable value for π.

Performance can be reported using a variety of criteria. Measuring performance in more precise terms using more than one criterion helps us to characterize the classifier better. Besides reporting the proportion of samples (or cases) accurately classified, we can use a weighted measure, that gives an overall figure considering the severity of the various disagreements between classifiers or between a classifier and the *gold standard.* Chapter 2 defines the expected loss, L, where L is the weighed loss of mis-classification. Other performance measures that can be used adjust for the agreement that can be expected due to guessing alone. For instance, when we predict a toss of a coin by guessing heads all the time we are expected to be correct 50% of the time so that $\frac{\pi-.5}{1-.5}$ might be an index of the improvement over guessing. Chapter 2 defines Brennan's kappa (κ_b) and $\kappa_{b,w}$ where $\kappa_{b,w}$ is the extension of Brennan's kappa to the case when the mis-classifications are of varying degrees of severity. $\kappa_{b,w}$ adjusts for agreement attainable due to chance when measuring the agreement between the classifier and a gold standard. Other performance measures that can be used adjust for the agreement that can be expected due to guessing alone

A model for the classifier, expert system or algorithmic, needs to be assumed for our research [9]. We assume that irrespective of the inputs to the classifier, the classifier classifies the input into one of several categories or classes possibly defined numerically. This covers a wide variety of expert systems whose primary task is classification. Expert systems, such as MYCIN, that do other tasks besides classifications can be judged within this model by having an expert rate the recommendations of the system in terms of a classificatory model (such as acceptable, not acceptable, quite acceptable).

Besides determining general criteria such as L and $\kappa_{b,w}$ for a classifier, one can ask the domain expert to define a domain-specific test for a classifier. An example of an expert-defined test (EDT) that tests the classifier on domain-specific tests is given in Chapter 6. An EDT can be used to increase confidence in a classifier. The tests studied in this thesis are based on perturbing the input data and observing the response of the classifier.

1.5 Combining classifier assessments to improve performance

With the availability of algorithmic classifiers one may wish to combine classifiers in order to improve performance (reduce L or increase $\kappa_{b,w}$). The Classifier-based Overall Loss Minimization Strategy (COLMS) and the Overall Loss Minimization Strategy (OLMS) are two domain-independent, criterion-based strategies introduced in this thesis. The COLMS is an independent discovery and development of the work by Wernecke on *coupling* classifiers to improve performance. The OLMS can use partitioning algorithms that divide the data-space as belonging to a set of *cells.* A classifier is a specific instance of a partitioning algorithm where the cells are the classes reported by the classifiers. Besides criterion-based domain-independent strategies, domain-specific strategies can be used that are based on a set of rules defined by an expert. Such a strategy for combining classifiers evaluating the atherosclerosis data in POSCH is given in Chapter 5.

With the availability of algorithmic classifiers we might wish to combine assessments from two or more classifiers in an effort to improve performance unless we know that one of the classifiers is a best possible classifier. First, we have to define what we mean by performance, as performance can be measured using several criteria such as weighted

loss [16, 78, 32, 31], kappa [8, 5], or weighted kappa [23]. Second, we have to select a combination strategy for assessments from the classifiers. The combination strategy may be based on a set of rules defined by an expert or we may use a statistically-defined strategy. Strategies of these types are studied in Chapter 5.

1.6 A word on notation used in the thesis

In evaluating or comparing classifiers on the basis of these performance criteria on real or simulated data, estimates of criteria are subject to random variation. This necessitates the use of statistical concepts and procedures to address the various questions in the previous paragraphs. We will use the notation $E[X]$ to refer to the expectation of the random quantity X and $SE(X)$ to refer to its standard error [45]. A quantity computed from data to estimate a criterion θ will often be denoted as $\hat{\theta}$. $P(A)$ is the probability of the event A. The word statistic as a noun will usually be used to refer to a quantity computed from the data. $\hat{\theta}$ denotes the estimated value of a parameter or criterion θ from a sample. $SE(\theta)$ denotes the estimated standard error of $\hat{\theta}$.

$$SE(\theta) = \hat{\sigma}_{\hat{\theta}}$$

Usually X denotes a scalar random variable, x is a scalar observation from a population, $\boldsymbol{x}$ denotes a vector observation, and $\boldsymbol{X}$ denotes a random vector. There will be some exceptions to this convention, but they will be clear from the context.

1.7 Organization of the thesis

Chapter 2 introduces criteria for evaluating classifiers and the motivations for having more than one criterion. Chapter 3 describes a real-world application and classifiers used in it. This real-world data provides a source for testing our ideas in addition to testing them on ideal data. Chapter 4 introduces procedures to estimate the properties of a classifier from the given dataset. Chapter 5 introduces the Classifier-based Overall Loss Minimization Strategy (COLMS) and the Overall Loss Minimization Strategy (OLMS). The COLMS and the OLMS are two domain-independent criterion-based strategies for combining classifier evaluations to reduce L. The COLMS is an independent discovery and development of the ideas by Wernecke [83].The OLMS can handle general partitioning algorithms that partition the data into categories other than the classes. A classifier is a special case of a partitioning algorithm. A specific ad-hoc expert-defined domain-specific strategy (EDSS) is introduced in Chapter 5. An example of an expert-defined test (EDT) that tests the classifier on domain-specific tests is given in Chapter 6. An EDT can be used to increase confidence in a classifier. The tests studied are based on perturbing the input data. Chapter 7 gives the results of performance evaluation using different techniques for various classifiers and classifier combination techniques on the AEPOSCH dataset. The AEPOSCH dataset is a subset of the atherosclerosis evaluation data collected by POSCH. Chapter 8 gives observations from the research and indicates directions for future research.

Chapter 2

Criteria for evaluating classifiers

This chapter introduces criteria for evaluating classifiers and the motivations for having more than one criterion. The result of classifying data by two or more classifiers is known as a cross-classification [36]. When two classifiers are classifying the data, the classification is known as a two-way classification. Several criteria have been used to study two-way classifications. If classifiers A and B classify the samples into C categories, the resulting classifications can be arranged in a square $C \times C$ table where entry (k, l) in the table corresponds to the i'th classification by classifier A and the j'th classification by classifier B. Such a table is known as a bivariate contingency table [36] or contingency table, for simplicity.

2.0.1 Notation, agreement, and association

Many of the definitions below can be applied when one has a rectangular $I \times J$ contingency table.

Let n be the sample size. For reference (throughout the rest of this thesis) define the following quantities:[1]

$$p_{ij} = P(\text{random case in cell } (i,j))$$

$$\pi_i = p_{i,.} = \sum_{j=1}^{C} p_{ij} = \text{P(cell } i \text{ in right margin)}$$

$$p_j = \sum_{i=1}^{C} p_{ij} = \text{P(cell } j \text{ in bottom margin)}$$

$p_{i,.}$ and p_j are known as the marginals.

$$f_{ij} = \text{frequency (count) of cases in cell } (i,j)$$

$$f_{i.} = \sum_{j=1}^{C} f_{ij} = \text{Number of cases in cell } i \text{ in right margin}$$

$$f_{.j} = \sum_{i=1}^{C} f_{ij} = \text{Number of cases in cell } j \text{ in bottom margin}$$

[1] As stated in Chapter 1 $P(A)$ is the probability of the event A.

	1	2	3	4
1	*			
2		*		
3			*	
4				*

Table 2.1: Table showing perfect agreement and perfect association.

and

$$\hat{p}_{ij} = f_{ij}/n = \text{proportion of cases in cell } (i,j)$$

$$\hat{\pi}_i = \hat{p}_{i.} = f_{i.}/n = \sum_{j=1}^{C} \hat{p}_{ij} = \text{proportion of cases in cell } i \text{ in right margin}$$

$$\hat{p}_j = \hat{p}_{.j} = f_{.j}/n = \sum_{i=1}^{C} \hat{p}_{ij} = \text{proportion of cases in cell } j \text{ in bottom margin}$$

Then $E[\hat{p}_{ij}] = p_{ij}$, and if the two classifications are statistically independent,

$$p_{ij} = p_{i,}p_{.,j} = \pi_i p_j$$

is the expected proportion of cases in cell (i,j). If the categories are defined by the choices of two classifiers, then π_i and p_i are the probabilities that the classifiers classify a randomly selected case as belonging to class ω_i. If one classifier's classification is defined as the "gold standard" or actual class, then π_i is the population proportion of cases in the class ω_i and p_j is the probability that the classifier classifies a randomly selected case as belonging to class ω_j.

An agreement criterion ought to be insensitive to scaling of all the numbers in the contingency table by the same amount. The agreement criterion should be easy to interpret [36]. It is usually important for the agreement criterion to be sensitive to the pattern of the marginals. This is because agreement between the classifiers will increase as the proportion of samples that belong to the classes that the raters agree on better (than other classes) increases. This fact should be reflected in an increasing value of the agreement criterion.

There are several differences between agreement and association. High agreement implies high association but the reverse need not be true. If two classifiers agree perfectly then all the entries will lie on the main diagonal entry of the table. This shows high association and high agreement [Table 2.1].

If we have ordered categories, then we might say that there is perfect disagreement because all the entries will lie on the cross-diagonal entry of the table. There is high association in this case [Table 2.2].

2.0.2 Criteria for measuring agreement or association

There are many criteria for measuring agreement or association. Several criteria for measuring association, such as the coefficient of contingency [32], are based on the chi-square

	1	2	3	4
1				*
2			*	
3		*		
4	*			

Table 2.2: Table showing no agreement and perfect association.

(χ^2) statistic.

$$\chi^2 = n \sum_{i=1}^{C} \sum_{j=1}^{C} \frac{(f_{ij} - f_{i.}f_{.j}/n)^2}{f_{i.}f_{.j}}$$

Criteria based on χ^2 do not distinguish between association and agreement. This is an important reason for not using them in studies of agreement [36, 23].

Other criteria are used for measuring association [32] but they cannot be used for measuring agreement. For example, Yule's Q [36, 74, 32] can be used to measure association between two variables. Q cannot be used to measure agreement, at least in part because it is insensitive to scaling of the marginals $p_{i,.}$ and $p_{.,j}$ [36]. Agreement criteria based on a log-linear model are given in [25, 2, 77]. The log-linear model can be used for measuring association among more than two variables and is equivalent to Q for the two-variable case. Criteria based on the log-linear model are hard to interpret, and suffer from the handicap that they are insensitive to scaling of the marginals.

Not all errors are equally serious. For instance, MYCIN's recommendations were judged as "accurate," "acceptable," and "unacceptable" and no effort was made to arrive at a pooled performance evaluation. If we have a system that rates a patient on a scale from "much better" to "much worse" as compared to his[2] previous condition, with several intermediate categories, then we need to have a pooled performance evaluation. If the system rates as "much worse" while the expert rates as "much better" then we have a serious error. This is in contrast to the situation where the system rates as "better" and the expert rates as "marginally better." The second error is less serious than the first.

To represent the differing degrees of seriousness of errors one can use a loss function as in [16, 78, 31] that assigns loss values L_{ij} when classifiers A and B classify an object in classes ω_i and ω_j, respectively.

In our application we will usually assume that classifier A is the correct classification (gold standard) and B is the classifier being evaluated. Moreover, we will always have $I = J = C$. Then π_i is the proportion of objects in the class ω_i which is represented by the k'th category, $(1 \leq k \leq I)$, in the population and p_j is the proportion of objects classified as belonging to the class ω_j which is the l'th category, $(1 \leq l \leq J)$, by the classifier.

Without loss of generality, assuming that the loss is bounded, we can assume that L_{ij} satisfy

$$L_{ii} = 0, \forall i = 1, \ldots, C$$

[2] In this thesis we will say "his" whenever we wish to refer to "her/his" and "him" whenever we wish to refer to "him/her."

$$0 \leq L_{ij} \leq 1, \forall i, j = 1, \ldots, C$$

As a matter of convention, a positive loss represents a true loss and a zero loss represents a perfect classification. The higher the value of L_{ij}, the more serious the error.

Instead of a loss function, we can use a weight or agreement function [23] that assigns weights w_{ij} to each object classified as ω_i by A and ω_j by B. If C is the number of classes, without loss of generality, assuming that the weights are bounded, we can assume that $\{w_{ij}\}$ satisfies

$$w_{ii} = 1, \forall i = 1, \ldots, C$$

$$0 \leq w_{ij} \leq 1, \forall i, j = 1, \ldots, C$$

As a matter of convention, a positive weight represents a partial agreement and a weight of unity represents a perfect agreement. The lower the value of w_{ij}, the more serious is the consequence of misclassification.

A loss function defines a weight function and vice versa by

$$w_{ij} = 1 - L_{ij}, \forall i, j = 1, \ldots, C$$

In some cases, the two classifiers are considered symmetrical when computing error [23]. In this case, we will assume

$$L_{ij} = L_{ji}$$

and equivalently

$$w_{ij} = w_{ji}$$

If all that matters is exact correctness of classification then

$$w_{ij} = 0 \quad (L_{ij} = 1), \forall i \neq j$$

is appropriate. A criterion for measuring agreement between two classifiers is the expected loss, L [46, 79, 16, 34, 78, 19, 18, 17], when classifying a randomly selected object

$$L = \sum_{i=1}^{C} \sum_{j=1}^{C} L_{ij} p_{ij} = \sum_{i=1}^{C} \sum_{j=1}^{C} L_{ij} p_{i|j} p_j \tag{2.1}$$

where

$$p_{i|j} = p_{ij} / p_j$$

The actual average loss entailed in classifying a sample is

$$L_o = \hat{L} = \sum_{i=1}^{C} \sum_{j=1}^{C} L_{ij} \hat{p}_{ij} = \sum_{i=1}^{C} \sum_{j=1}^{C} L_{ij} \hat{p}_{i|j} \hat{p}_j \tag{2.2}$$

where

$$\hat{p}_{i|j} = f_{ij} / f_{.j}$$

Alternatively, we may call $\hat{L}$ the estimated expected loss. In this thesis we shall use $\hat{L}$ to refer to the estimated expected loss unless stated otherwise. We similarly define the expected agreement as

$$W = \sum_{i=1}^{C} \sum_{j=1}^{C} w_{ij} p_{ij} = 1 - L \tag{2.3}$$

and the actual or observed agreement is

$$W_o = \hat{W} = \sum_{i=1}^{C}\sum_{j=1}^{C} \hat{p}_{ij} w_{ij} = 1 - \hat{L}$$

The standard error of the estimated loss and observed agreement can be computed using known theoretical results for multinomials [84].

$$SE(\hat{W}) = SE(\hat{L}) = \sqrt{\frac{1}{n}(\sum_{i=1}^{C}\sum_{j=1}^{C} L_{ij}^2 p_{ij} - (\sum_{i=1}^{C}\sum_{j=1}^{C} L_{ij} p_{ij})^2)} \tag{2.4}$$

L is a raw figure for disagreement. It does not take into account that a certain amount of agreement takes place with the classifier used as the gold standard because of chance alone. The gold standard is the method whose classification is accepted as correct. The gold standard might be a human expert, the result of a laboratory test, or a classifier. The gold standard is often an expert in expert systems research. Chance agreement can be illustrated by the tossing of a fair coin. A person who guesses "heads" all the time will on the average have an agreement rate of 0.5.

2.0.2.1 Proportionate reduction of error (PRE) measures or κ

The proportionate reduction of error (PRE) measures or kappa criteria were explicitly designed to measure agreement [36, 8, 23, 4] adjusting for the degree of agreement due to chance alone. The PRE measures are of the form

$$\kappa = \frac{W - W_c}{1 - W_c}$$

or equivalently

$$\bigtriangledown = \frac{L_c - \text{Ł}}{\text{Ł}}$$

where

W is the expected observed agreement between the classifiers,

L is the expected observed loss,

W_c is the expected agreement due to chance alone (under a strategy of guessing), and

L_c is the expected loss due to chance alone (under a strategy of guessing).

Some strategies of guessing are:

1. The classifier ignores the data completely and selects randomly with probabilities $p_1, \ldots, p_C$ where C is the number of the classes. This idea leads to the definition of Cohen's kappa (κ_c) [23] and Hildebrand's $\bigtriangledown$ [36].

2. The classifier selects the one class with the lowest a priori expected loss $\hat{L}$, that is class ω_j that minimizes $\sum_{i=1}^{C} \pi_i L_{ij}$. This idea leads to the definition of Brennan's kappa (κ_b) [5] or kappa base.

Natural estimators of κ and $\bigtriangledown$ are

$$\hat{\kappa} = \frac{\hat{W} - \hat{W}_c}{1 - \hat{W}_c}$$

and

$$\hat{\bigtriangledown} = \frac{\hat{L}_c - \hat{L}}{\hat{L}_c}$$

If the classifiers are independent, as would be the case if a classifier is guessing independently of the data,

$$p_{ij} = \pi_i p_j$$

then the expected loss due to chance alone would be

$$L_c = \sum_{i=1}^{C} \sum_{j=1}^{C} \pi_i p_j L_{ij}$$

and the expected agreement due to chance alone would be

$$W_c = \sum_{i=1}^{C} \sum_{j=1}^{C} \pi_i p_j w_{ij}$$

In this case,

$$\kappa_c = \frac{\sum_{i=1}^{C} \sum_{j=1}^{C} p_{ij} w_{ij} - \sum_{i=1}^{C} \sum_{j=1}^{C} \pi_i p_j w_{ij}}{1 - \sum_{i=1}^{C} \sum_{j=1}^{C} \pi_i p_j w_{ij}} \tag{2.5}$$

and

$$\bigtriangledown = \frac{\sum_{i=1}^{C} \sum_{j=1}^{C} \pi_i p_j L_{ij} - \sum_{i=1}^{C} \sum_{j=1}^{C} p_{ij} L_{ij}}{\sum_{i=1}^{C} \sum_{j=1}^{C} \pi_i p_j L_{ij}}$$

The above form of L_c (and W_c) is appropriate when we are trying to develop an index for the agreement of two raters after adjusting for the agreement due to random agreement between the raters when the marginals of the raters are assumed to be fixed. In the literature, κ_c is known as Cohen's weighted kappa [23] and $\hat{\bigtriangledown}$ as Hildebrand's $\bigtriangledown$ [36]. The equivalence of Cohen's kappa and Hildebrand's $\bigtriangledown$ is proved in [26].

2.0.2.2 Brennan's κ and $\kappa_{b,w}$

Brennan [5] argues that the correct form of p_e to be chosen when using an agreement criterion to describe the degree of agreement between a classifier's judgments about objects and the known "criterion" status of these objects is determining the improvement with respect to a "best" strategy using only the marginal probabilities. This best strategy allocates all samples to the class that maximizes the "benefit" as determined by the knowledge of the marginals without knowledge of the data in the individual case. We will refer to such a guesser as a best uninformed guesser. Brennan then derives a criterion (κ_b) which is applicable when perfect classifications receive a score of 1, and any mis-classifications receive a score of 0.

We extend Brennan's observation to the case where weights are used to represent partial agreement. Let

$$W = \sum_{i=1}^{C} \sum_{j=1}^{C} w_{ij} p_{ij}$$

$$W_j = \sum_{i=1}^{C} w_{ij} \pi_i$$

W_j is the expected benefit of selecting class ω_j for a random case without observing the data. Define

$$W_b = max_j W_j$$

that is,

$$W_b = max_j \sum_{i=1}^{C} w_{ij} \pi_i, j = 1, \ldots, C$$

$$\begin{aligned} \kappa_{b,w} &= \frac{W - W_b}{1 - W_b} \\ &= \frac{\sum_{i=1}^{C} \sum_{j=1}^{C} w_{ij} p_{ij} - max_j \sum_{i=1}^{C} w_{ij} p_{i,.}}{1 - max_j \sum_{i=1}^{C} w_{ij} p_{i,.}} \end{aligned} \tag{2.6}$$

For the case of tossing a biased coin, let the outcome of heads be denoted as "H", and that of tails as "T",

$$P(H) = 0.7$$

$$P(T) = 0.3$$

$$w_{H|H} = w_{T|T} = 1$$

$$w_{T|H} = w_{H|T} = 0$$

$$W_1 = 0.7$$

$$W_2 = 0.3$$

$$W_b = max(W_1, W_2) = 0.7$$

$$W = 0.7 \times 1 + 0.3 \times 0 = 0.7$$

We compute $\kappa_{b,w}$ as follows:

$$\kappa_{b,w} = (0.7 - 0.7)/(1 - 0.7) = 0$$

Then a best partially informed guesser (who is guessing based solely on only the prior marginal probabilities of the classes) will always choose "Heads (H)." The above example illustrates that $\kappa_{b,w}$ is defined so that a best guesser guessing based on a priori probabilities of occurrence of the classes will get a score of 0. Positive (negative) values of $\kappa_{b,w}$ reflect better (worse) "guesses" than that obtained from guessing based on a priori probabilities alone.

Criteria such as L and $\kappa_{b,w}$ describe the properties of a classifier applied to a particular population. The ratios $\{p_{ij}/\pi_i\}$ are characteristics of the classifier that are not changed if the

population proportions $\{\pi_i\}$ change. Finally, the values $\{\pi_i\}$ together with the distribution of data for each category characterize the population.

Consider this contrived case of hypertension (Ht) to give a better idea of the concepts involved. Suppose hypertension (Ht) occurs in a population with $P(Ht) = .3$. Information from family history may be used to classify a person with the following success rate

If 0 represents the class non-HT and 1 represents the class HT then the symmetric loss and gain functions are given by

$$L_{0,0} = L_{1,1} = 0; w_{0,0} = w_{1,1} = 1$$

$$L_{0,1} = L_{1,0} = 1; w_{0,1} = w_{1,0} = 0$$

The ratio of the cell-probabilities to the marginals of the true state of hypertension (p_{ij}/π_i) is given in Table 2.4.

The ratio of the cell-probabilities to the marginals of the predicted state of hypertension (p_{ij}/p_j) based on family history is given in Table 2.5.

$$W_c = 0.3 \times 0.45 \times 1 + 0.3 \times 0.55 \times 0 + 0.7 \times 0.45 \times 0 + 0.70 \times 0.55 \times 1 = 0.52$$

Using Equation 2.5 we get

$$\kappa_c = 0.438$$

For computing $\kappa_{b,w}$ note that a best guessing strategy based on knowledge of the marginals is to select "non-HT" each time. For this case,

$$W_c = 0.70$$

Using Equation 2.6 we get

$$\kappa_{b,w} = \frac{.73 - .70}{1 - .70} = .10$$

By computing $\hat{\kappa}_{b,w}$ of any classifier for a given dataset, we estimate how the classifier classifies with respect to a best guesser using a-priori probabilities π_i and the loss values L_{ij}. Using L and $\kappa_{b,w}$ gives a clearer idea of the classification performance of a classifier than L alone. In this example, $\kappa_{b,w}$ shows that there is little to gain from guessing by using the given classifier. When choosing classifiers, usually the classifier with the lowest value of L will be chosen. This is equivalent to choosing the classifier with the highest value of $\kappa_{b,w}$ because if two classifiers C_1 and C_2 are used on the same population

$$\kappa_{b,w}(C_1) > \kappa_{b,w}(C_2) \text{ iff } L(C_1) < L(C_2)$$

$\kappa_{b,w} = 1$ requires $W = 1$ or $L = 0$ and represents perfect validity. $\kappa_{b,w} = 0$ represents no agreement beyond that due to a best uninformed strategy, and negative values of $\kappa_{b,w}$ represent agreement poorer than that due to a best uninformed strategy.

We do not need to test for the marginals being homogeneous (having identical distributions) when comparing two values of $\kappa_{b,w}$. In fact, a test of homogeneity of marginals [86] is not desirable, because, however the classifier chooses to optimize its performance against the gold standard, we would be justified in using the classifier with the highest value of $\kappa_{b,w}$ if our loss function is well-defined and unlikely to change. In Table 2.3 the marginals are

not homogeneous and the hypertension probabilities of mis-classification given in Table 2.3 give rise to a loss:

$$L = 0.24 \times 0 + 0.21 \times 1 + 0.06 \times 1 + 0.49 \times 0 = 0.27$$

Consider the homogeneous marginal probabilities of the classifier [Table 2.6]. L for the homogeneous classifier [Table 2.6] is:

$$L = 0 \times 0 + 0.3 \times 1 + 0.3 \times 1 + 0.4 \times 0 = 0.6$$

L for the classifier with homogeneous marginals is greater than L for the classifier with non-homogeneous marginals. We may wish to group the classes into broader classes. In this case we can use the new classes when computing and comparing agreement criteria.

It is important to note that any of the above performance criteria depend on the population under study and the classifier. For instance, if the population under study is changed, the same classifier will yield different values of L, W, and $\kappa_{b,w}$, as a continuation of the previous hypertension example shows.

In the previous hypertension example, we keep the probabilities of mis-classification, P(HT | non-HT) and P(non-HT | HT), and the probabilities of correct classification, P(HT | HT) and P(non-HT | non-HT), unchanged in the new population. The loss function is not changed. However, assume the incidence of hypertension is increased to .60 in the population. The new cell-probabilities are given in Table 2.7.

The new values of κ_c and $\kappa_{b,w}$ are as below:

$$\kappa_c = 0.5$$

$$\kappa_{b,w} = 0.400$$

There is a substantial change in the value of $\kappa_{b,w}$ between the two populations, although the classification knowledge used is the same.

True State	Classified as Ht	Classified as non-Ht	π_i
Ht	.24	.06	.30
non Ht	.21	.49	.70
p_j	.45	.55	1.00

Table 2.3: Hypertension probabilities of mis-classification.

True State	Classified as Ht	Classified as non-Ht	Sum
Ht	0.8	0.2	1.0
non Ht	0.3	0.7	1.0

Table 2.4: p_{ij}/π_i for hypertension.

True State	Classified as Ht	Classified as non-Ht
Ht	0.533	0.109
non Ht	0.467	0.891
Sum	1.000	1.000

Table 2.5: p_{ij}/p_j for hypertension.

True State	Classified as Ht	Classified as non-Ht	π_i
Ht	0	.30	.30
non Ht	.30	.40	.70
p_j	.30	.70	1.00

Table 2.6: Hypertension probabilities of mis-classification for a homogeneous classifier.

True State	Classified as Ht	Classified as non-Ht	π_i
Ht	.48	.12	.60
non Ht	.12	.28	.40
p_j	.60	.40	1.00

Table 2.7: Hypertension probabilities p_{ij} of mis-classification.

2.1 Summary

This chapter suggests two criteria from the literature for evaluating classifiers. L is a well-established criterion in the pattern recognition literature. Using a loss function enables one to grade the severity of mis-classification. The importance of using $SE(L)$ when evaluating a classifier or making comparisons of classifiers is stressed. Cohen's kappa (κ_c) [8, 23, 5] is inappropriate to compare how well a classifier does as compared to a gold standard after one takes into account the agreement due to chance alone. "Kappa base" ($\kappa_{b,w}$) or Brennan's kappa (κ_b) [5] compares how well a classifier does as compared to the best classifier guessing solely on the knowledge of the loss values and the a priori distribution of the classes in the population. $\kappa_{b,w}$ and $SE(\kappa_{b,w})$ are useful when comparing classifiers based on statistical algorithms, neural nets, expert systems, etc. to decide how well they do as compared to the classifier obtained with nearly zero effort (viz., the classifier guessing solely on the basis of a priori distribution of the classes). With L, $SE(L)$, $\kappa_{b,w}$, and $SE(\kappa_{b,w})$ it is possible to get a more complete evaluation of a classifier than when using L alone. The next two chapters introduce the real-world dataset that will be used to demonstrate the use of these criteria and techniques for estimating the value of these (or any other) criteria and their standard errors.

Chapter 3

Expert and classifier evaluation of POSCH atherosclerosis data

Much of the theory in this thesis will be illustrated using the serial coronary angiogram evaluation application from the Program On the Surgical Control of the Hyperlipidemias (POSCH). An angiogram is a X-ray cine film of the heart taken after injecting radiographic dye into the coronary arteries using a catheter. Since the radiographic dye is opaque to X-rays, it is possible to visualize the flow of blood through the coronary arteries by using an angiogram and thus determine the diameter of the lumen (bore) of the artery. This application may be referred to as the AEPOSCH application and the dataset obtained from this application as the AEPOSCH dataset. This chapter will describe the POSCH data and the procedure used by POSCH experts to evaluate the angiograms. The resulting data are computerized. This chapter also includes a description of the classifiers used to evaluate the computerized data resulting from the angiograms.

POSCH was a large clinical trial that followed the progress of 838 patients for an average of nine years and up to 15 years. The study ended in 1990. The patients selected for the study had one heart attack before entry into the study. Half the patients were randomly assigned to undergo a surgical operation to reduce their lipid levels such as cholesterol. The other half was a control group. POSCH developed an expert system, ESCA, and algorithmic classifiers to evaluate a part of its data.

The AEPOSCH application consists of evaluating two angiograms of the same patient taken some years apart. Angiograms were taken at baseline when a patient entered the study and at three, five, and seven or ten years into the study. POSCH created a panel of experts to evaluate the angiograms. Two member subpanels were drawn from this panel to evaluate the angiograms. Pairs of angiograms, labeled Film A and Film B, were assigned to the subpanel for reading and evaluation. When the angiograms were assigned to the subpanels for evaluation, it was not revealed whether the patient belonged to the intervention group or the control group. Further, the angiograms were randomized with respect to the time order that they were taken in. Therefore Film A may have been taken before or after Film B with one exception. This exception was when a patient has had a coronary bypass or angioplasty operation. It was not possible to disguise the time order in this case. Therefore, the angiogram taken before the operation was always given as Film A. All of the evaluated cases in the AEPOSCH dataset used in this thesis are baseline and followup cases. At baseline, when patients were admitted into the study they did not have a bypass or angioplasty

operation.

Because of the randomizing procedures above, the identity of the patient and the time order of the angiograms were not revealed to the subpanel. The angiograms are simply labeled Film A and Film B. Quality-control information for measuring reproducibility is gathered by having another subpanel evaluate some of the pairs of angiograms a second time. The two subpanels did not always agree with each other but their evaluations were quite consistent.

The two experts reviewed Film A independently and recorded their findings of the stenosis (narrowing or contraction of the arterial segments) of the arteries on an Independent Film A form containing a diagram of the patient's arteries [Figure 3.1]. After the independent evaluations were done, the two experts compared their findings artery by artery and arrived at a consensus with respect to the location and size of each stenosis. The experts coded the stenosis in an arterial segment as "999" if the stenosis could not be visualized because of a total obstruction in an upstream arterial segment or because of other technical reasons.

The consensus regarding the stenosis is recorded by a technician on a Consensus A form [Figure 3.2]. The Consensus A form was then copied on an Independent Film B form, a copy of which was then given to the experts. The experts independently reviewed Film B and recorded their findings of the change in stenosis along with an overall global assessment of change from Film A to Film B on the independent Film B form [Figure 3.3]. A consensus on the change in stenosis in each artery was then arrived at and recorded by the technician on the Consensus B form [Figure 3.4]. The experts then arrived at a consensus on the overall change and this consensus was recorded by the technician on the Consensus B form [Figure 3.4]. The experts' overall global assessment of change was recorded using an eight-point scale from much worse to much better [Table 3.1], encoded as (-3, -2, -1, -0, +0, +1, +2, +3) or (1, 2, 3, 4, 5, 6, 7, 8). The information on the Consensus A and Consensus B forms was entered on an M-14 form [Figure 3.5] and added to the POSCH database at a later time at the coordinating center. Besides this information, the M-14 form contains other information such as the condition of the film and the visibility of the stenoses. This additional information was not used by either the expert system or the algorithmic classifiers. Before supplying the information on the stenosis to the expert system and the classifiers the values of the stenoses are transformed so that if either the Film A or the Film B stenosis of a particular arterial segment is encoded as a "999," then both the Film A and the Film B stenoses are treated as 0.

As can be seen from the above description, the experts read Film A first, and used their observations of stenosis of Film A while making observations on Film B. Because of this fact they can be biased while making observations on Film B. In Section 7.2, a bias is detected towards having more assessments in the categories -1 to -3 than in the categories +1 to +3. The categories -0 and +0 are not considered because there is so little difference between them. Dr. John Matts[1] believes that the bias might be because in the event of a coronary bypass or an angioplasty operation, Film B would always be the film showing the operation. Film B would therefore be later in time and usually show increased atherosclerotic disease. Another possible factor, in Dr. Matts' opinion, is a strong subconscious pessimistic bias

[1] John Matts, PhD, was assistant director of the POSCH coordinating center and the chief statistician of POSCH.

Meaning	Coding Scheme 1	Coding Scheme 2
Much worse	-3	1
Worse	-2	2
Slightly Worse	-1	3
Imperceptibly Worse	-0	4
Imperceptibly Better	+0	5
Slightly Better	+1	6
Better	+2	7
Much Better	+3	8

Table 3.1: Scale for experts' opinion of overall change in ESCA.

on the part of the experts, because in clinical practice, doctors see more patients whose atherosclerosis increases with time rather than decreases. Therefore the experts maybe inclined to be pessimistic regarding the progression/retardation of atherosclerotic disease in the arteries. Even though the time order is randomized (except for films where coronary bypass or angioplasty operations have taken place) and the doctors are aware of this, there may have been a subconscious bias to view Film B as the one taken later in time. In this case, the doctors' clinical experience may have biased their observations.

3.1 The expert system, ESCA

The expert system, ESCA (Evaluator of Serial Coronary Angiograms), does the task of arriving at an overall assessment of change based on the percentage of stenoses found in the arterial segments and recorded on the M14 form.

Some arterial segments listed on the form have distal (succeeding or downstream) segments [Table 3.2]. If there is a complete obstruction of flow of blood (a 100% stenosis) in any arterial segment, the flow in the distal segments usually cannot be visualized unless there is collateral filling[2]. In this case the flow is coded as "999." If an artery has a complete obstruction of flow of blood (a 100% stenosis) in the distal arterial segment then there is usually a contraction of the lumen in it. Because of this contraction, the experts cannot decide whether a change in stenosis is because of the contraction of the artery due to the blockage in the distal segment, or a change in the deposition of the atherosclerotic material in the artery itself. For this reason, the experts tend to give less weight to changes in segments whose distal segments show a complete obstruction in either Film A or Film B, which is reflected in the expert system's rules. In the expert system, when we observe that the distal segment of an artery has a complete blockage in either Film A or Film B, we reduce the importance given to the change in the preceding (upstream) segment. Table 3.2 gives the names of arterial segments and their distal arterial segments.

ESCA arrives at an assessment of change by using the following procedure.

[2] There are times when vessels around the stenosis spontaneously expand to provide blood flow to the muscles. These vessels are known as collateral arteries.

Figure 3.1: Independent Film A Form.

Figure 3.2: Consensus Film A Form.

Figure 3.3: Independent Film B Form.

Figure 3.4: Consensus Film B Form.

Figure 3.5: M14 Form.

Arterial segment	Distal arterial segment
Right-ostium	none
Proximal-right-coronary	mid-right-coronary
mid-right-coronary	distal-right-coronary
distal-right-coronary	none
proximal-posterior-descending	distal-posterior-descending
distal-posterior-descending	none
left-ostium	none
left-main-coronary	none
proximal-lad	mid-lad
mid-lad	distal-lad
distal-lad	none
proximal-left-circumflex	mid-left-circumflex
mid-left-circumflex	distal-left-circumflex
distal-left-circumflex	none
first-diagonal-lad	second-diagonal-lad
second-diagonal-lad	third-diagonal-lad
third-diagonal-lad	none
first-obtuse-marginal	second-obtuse-marginal
second-obtuse-marginal	third-obtuse-marginal
third-obtuse-marginal	none
other-segment-a	none
other-segment-b	none
other-segment-c	none

Table 3.2: Arterial segments with their distal arterial segments. lad denotes left-anterior-descending.

lower bound of change	upper bound of change	points
-100	-65	3
-65	-35	2
-35	-20	1
-20	20	0
20	35	-1
35	65	-2
65	100	-3

Table 3.3: Mapping of change in stenosis to individual artery points in ESCA. The table is scanned from top to bottom and points allocated according to the first satisfied range.

1. For each arterial segment (as) compute,
   change[as] = stenosis in Film B[as] - stenosis in Film A [as]
   The names of the arterial segments are given in the first column of Table 3.2. "as" ranges over all these names.

2. Adjusted change[as] = change[as] if there is no 100% blockage in the distal (succeeding or downstream) arterial segment in either Film A or Film B. Table 3.2 gives the names of arterial segments with distal segments.
   Adjusted change[as] = change[as]/ 2, otherwise.

3. Adjusted Percent stenosis (APB) in Film B [as] = Percent stenosis in Film A[as] + Adjusted Change [as].

4. Individual Artery Points [as] (IAP[as]) are allocated as in Table 3.3.

5. Total Points = sum of individual artery points for all arterial segments.

6. Tentative assessment is decided based on the total points as in Table 3.4.

7. If the tentative assessment is "unassigned" then the sum of adjusted change in all arteries is computed. In this sum only those arteries are considered where the change has not been from a 100% blockage. Changes to or from zero are weighted by a factor of 1.5. If this sum is greater than 0 then the final assessment is "zero-minus." If this sum is less than 0 then the final assessment is "zero-plus."

   If the sum above is 0 then the sum of adjusted change in all arteries is computed again. In this sum even those arteries are considered where the change has not been from a 100% blockage. Changes to or from zero are weighted by a factor of 1.5. If this sum is greater than 0 then the final assessment is "zero-minus." If this sum is less than 0 then the final assessment is "zero-plus."

   If both the sums above are 0 then the final assessment is "zero-no-change."

8. If the tentative assessment is +3 then the number of changes greater than or equal to 50% (NPH), the number of changes greater than or equal to 0 and less than 50%

lower bound	upper bound	tentative assessment
-69	-6	-3
-5	-4	-2
-3	-2	-1
-1	1	unassigned
2	3	1
4	5	2
6	69	3

Table 3.4: Mapping of total-points to tentative-assessment in ESCA.
In practise, for the cases in the B3, B3-QA, BR, and BR-QA datasets described in Chapter 7, $-21 <$ total-points < 18.

(NPL), the number of changes less than 0 and greater than -50% (NNL), and the number of changes less than or equal to -50% (NNH) is counted.

If (((NPH > 2) or (NPL > 6)) and (NNH = 0) and (NNL < 3)) then
final assessment = +3
else
final assessment = + 2

9. If the tentative assessment is -3 then the number of changes less than or equal to -50% (NNH), the number of changes less than or equal to 0 and greater than -50% (NNL), the number of changes greater than 0 and less than 50% (NPL), and the number of changes greater than or equal to -50% (NPH) is counted.

   If (((NNH > 2) or (NNL > 6)) and (NPH = 0) and (NPL < 3)) then
   final assessment = -3
   else
   final assessment = -2

10. If the tentative assessment is -2, -1, 1, or 2, then the final assessment is the same as the tentative assessment.

The development process of ESCA took place as follows:

1. A set of 8 cases from the baseline three-year study, one from each category was presented to one of the POSCH panel members, Dr. David Hunter. Dr. David Hunter evaluated these cases, stating aloud his reasoning strategy. At the end of the knowledge engineering session all of the rules above (except 7, 8, and 9) were developed. The rules were tested manually and appeared to work. Dr. John Matts, the chief statistician at POSCH monitored and checked the work.

2. A prototype expert system was built and 200 cases evaluated using the expert system. The evaluation of this expert system was subsequently done and presented to the domain expert Dr. Hunter who suggested rules 7, 8, and 9 be added.

3. The additional rules were incorporated into the prototype system and the system tested again on the same 200 cases. The results were reviewed again by the domain expert. The results were judged to be satisfactory and the results reported in [48].

The identity of the cases used in the development process above is not known. In the results presented in Chapter 7 practically all these 208 cases were inseparably mixed in the 630 cases used to evaluate ESCA for the baseline three-year cases. It can be argued that in some sense ESCA has been "trained" on the 208 cases used for testing the system in the development process, as the expert has reviewed the results of ESCA on these cases. For these reasons, the results presented in Chapter 7 cannot be strictly considered as an indication of ESCA's performance should it encounter an unseen case that has one visit from baseline and another from follow-up.

3.2 Multiple linear regression

The expert's assessment on a scale from 1 to 8 (using Coding Scheme 2 in Table 3.1) is regressed against the variables $s_i, i = 1, \ldots, 46$. The s_i are the measured stenoses in Film A and Film B of artery i in Table 3.5. The coefficients of the multiple linear regression equation [15] fitted by least squares, are used in the prediction equation,

$$P = a_0 + \sum_{i=1}^{46} a_i s_i$$

where

a_0 is the intercept,

a_i is the i'th regression coefficient,

s_i is the stenosis value recorded for arterial segment i as in Table 3.5.

P is rounded to give an integer value PI. The assessment A is computed as follows:

$$A = 1, \text{ if } PI \leq 1$$
$$A = 8, \text{ if } PI \geq 8$$
$$A = PI, \text{ otherwise}$$

The coefficients of the multiple linear regression equation fitted by least squares when training over the entire B3 dataset are given in Table 7.1.

3.3 Back-propagation

Back-propagation [62] is the model used to set up a *neural network* using the POSCH data. A neural network is typically composed of simple units connected to each other with each link between units assigned a connection "strength." The values of the connection strengths constitute the information in a neural network. A typical unit with connections between it and other units is shown in Figure 3.6.

Figure 3.6: A simple unit in back-propagation.

In back-propagation a network of units is set up. We are interested in back-propagation!layered feed-forward networks. In such networks, units are arranged in *layers* with input units in the bottom layer and output units in the top layer. There can be many layers of *hidden units* between the bottom layer and the top layer. Every unit must send its output to higher layers than its own, if any, and receive its input from lower layers than its own, if any. If the input consists of a vector of a length m and the output consists of a vector of length n, then there are m units in the bottom layer and n units in the top layers. Given an input vector, the input units are set to the value of the input vector, the values of the output units are computed by an upward pass which computes the output levels of each layer in turn using the already computed output levels in the earlier level. The discrepancy between the output values and the output vector is fed downwards into the network and the connection strengths adjusted. This process is continued iteratively over all the patterns in the training set and is the training process for back-propagation.

A network that has only an input layer and an output layer has no *hidden units.* A hidden unit is a unit in an intermediate layer, i.e., a layer whose units are not connected directly to the input or output patterns. A network without any hidden units cannot do the mapping from the input to the output patterns when the similarity structure of the input and output patterns is complex. A classic example of this case is the exclusive-or function which cannot be learned by a two-layer network but can be learned by a network having hidden units.

A small network with $m = 5$ and $n = 2$ is shown in Figure 3.7. In the figure, the output of each unit in a layer is connected to the input of every unit in the succeeding layer. This is the setup in our experiments. In general, this need not be the case. Let

> v_{ji} be the weight on the connection from the output of the i'th unit to the input of the j'th unit,
>
> o_i be the output of the i'th unit assumed between 0 and 1,
>
> θ_j be the threshold of the j'th unit.

Then for the j'th unit, o_j is computed as:

$$o_j = \frac{1}{1 + exp(-\sum_i v_{ji} o_i + \theta_j)} \tag{3.1}$$

These units can theoretically encode a pattern ranging in values from 0 to 1. In Equation 3.1, o_j can never reach the values of 0 and 1 in practice since that woud require the exponent in the denominator to be $+\infty$ or -∞, respectively. If we restrict our outputs to be between 0 and 1, then this requires that either our weights or our thresholds are $+\infty$ or $-\infty$.

For the purpose of training a network, an error signal, δ_j, has to be computed for each unit u_j in the output and hidden layers and propagated downwards to units in the lower layers. If unit u_j is an output unit, its error signal δ_j is computed as

$$\delta_j = (t_j - o_j) o_j (1 - o_j)$$

where t_j is the target or desired output value. If u_j is a hidden unit, δ_j is computed as

$$\delta_j = o_j (1 - o_j) \sum_k \delta_k v_{kj}$$

Figure 3.7: A back-propagation network with one hidden layer and two units in the output layer.

where the sum is over all units directly above of unit u_j. The weights are adjusted iteratively as follows. On the n'th iteration, i.e., the n'th time, the change in weight $\Delta v_{ji}^{(n+1)} = v_{ji}^{n+1} - v_{ji}^{n}$, is computed as

$$\Delta v_{ji}^{(n+1)} = \eta \delta_j o_i + \alpha \Delta v_{ji}^{(n)}$$

where

η is a constant determining how much the current error signal changes the weights. η is known in the literature as the learning rate [62],

α is a constant which determines the effect of past weight changes on the current weight change. The effect of α is to increase the rate of convergence and contribute to stability.

The initialization of the network takes place by initializing random weights to the connections in the network. In our experiments, the weights were randomly chosen between -7 and 0.999999. The choice of these weights ensures that most units are given a negative input when the network is initialized and presented with the first few patterns. The exact numbers -7 and 0.999999 were chosen arbitrarily. Other values will probably work too. Since the classification is described by a single number we have only one output unit. For each pattern the predictor variables constitute the input values to the net and the correct classification constitutes the value presented to the output unit. The goal is that for every input vector the network must learn to predict the value presented at the output unit. We have 46 input units because Since we have 46 predictor variables. The experts' classification is presented to the output unit. The network is trained for k iterations as follows.
Do k times (or until convergence criteria met before k iterations are done)
begin
For each pattern γ in the training set:

1. Establish the input portion of the pattern at the input layer. The output portion of the pattern constitutes the target for the output layer.

2. For layers $l = 2, \ldots, o$ where o is the output layer, compute the output of units in layer l, after the output of units in the lower layers have been computed.

3. Compute the discrepancy $t_i - o_i$ between the output values of the units and the target values of the units in the output layer.

4. Propagate the error down through all the units from the output layer, by computing δ_j for each unit.

5. Adjust the weight of each connection in the network.

end

In our experiments a network with 46 input units and one output unit was used. 600 iterations and $\eta = 0.3$ and $\alpha = 0.9$ were used. Input unit i is set to one-hundredth the value of the stenosis recorded for s_i in Table 3.5. The output unit is set to one-tenth[4] the expert's assessment using coding scheme 2 in Table 3.1.

[4] Setting the output to one-eigth the expert's assessment might fully utilize the expert's assessment scale, but one-tenth was better to visually assess the results when setting up the experiments

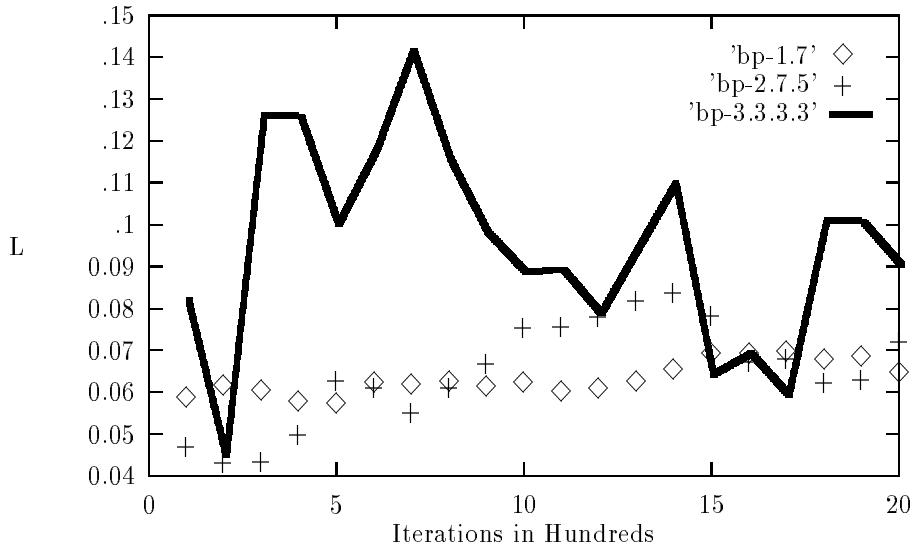

Figure 3.8: $\hat{L}$ on test cases vs. number of iterations for some promising configurations for back-propagation networks using 1, 2, and 3 hidden layers.
bp-2.7.5 refers to a back-propagation network with two hidden layers, seven units in the first hidden layer, and five units in the second hidden layer.

Three promising configurations obtained in [42] were tested for 2000 iterations and the configuration with one hidden layer having seven units was selected. The second configuration had two hidden layers with seven units in the first layer and five units in the second layer. The third configuration had three hidden layers with three units in each layer. The configuration selected usually had the lowest value of loss. The results of the testing are shown in Figure 3.8.

If P is the output of the network, $PI = \text{round}(P \times 10)$ is computed. The assessment A is computed as follows:
$A = 1$, if $PI \leq 1$
$A = 8$, if $PI \geq 8$
$A = PI$, otherwise

3.4 C4

C4 [60] is a C language implementation of ID3 [61]. Given a small training set of objects whose class is known, ID3 uses induction to derive a classification rule that can determine the class of any object from its values of the attributes. ID3 represents the classification rule in the form of a decision tree. Table 3.6 shows a small training set.

Each object's value of each attribute is shown, together with the class of the object (class "P" or "N"). A decision tree that correctly classifies each object in the training set is given in Figure 3.9. Leaves of a decision tree are class names. Other nodes represent attribute-based tests with a branch for each possible outcome. In order to classify an object, we start at the root of the tree, evaluate the test, and take the branch appropriate to the

outcome. The process continues until a leaf is encountered, at which time the object is asserted to belong to the class named by the leaf.

In C4 and ID3 a subset of the training set called the *window* is chosen at random and a decision tree formed from it using an information based criterion. This tree correctly classifies all objects in the window if there are no cases with identical data values and different assessments. All other objects in the training set are then classified using the tree. If the tree gives the correct answer for all these objects then it is correct for the entire training set and the process terminates. If not, a selection of the incorrectly classified objects is added to the window and the process continues. An option is present to disable windowing, i.e., to use all objects in the training set for training the classifier at the first attempt to form a decision tree.

To form a decision tree for an arbitrary collection C of objects, the following procedure is used. It is assumed here that no cases exist with the same data and different classes. If C is empty or contains objects of only one class, the simplest decision tree is just a tree labeled with the class. Otherwise, let T be any test on an object with possible outcomes $O_1, O_2, \ldots, O_w$. Each object in C will give one of these outcomes for T, so T produces a partition $\{C_1, C_2, \ldots, C_w\}$ of C with C_i containing those objects having outcome O_i. This is represented graphically by the tree form in Figure 3.9. If each subset C_i in this figure could be replaced by a decision tree for C_i, the result would be a decision tree for all of C. As long as two or more $C_i's$ are non-empty, each C_i is smaller than C. In the worst case, this divide-and-conquer strategy will yield single-object subsets that satisfy the one-class requirement for a leaf. Thus, provided that a test can always be found that gives a non-trivial partition of any set of objects, this procedure will always produce a decision tree that correctly classifies each object in C.

ID3 (and C4) uses an information-based method. Let C contain p objects of class P and n of class N. The assumptions are:

1. Any correct decision tree for C will classify objects in the same proportion as their representation in C. A randomly selected object from C will be determined to belong to class P with probability $p/(p+n)$ and to class N with probability $n/(p+n)$.

   If there are $k, (2 \leq k)$ classes, say $M_1, M_2, \ldots, M_k$ then an arbitrary object will be determined to belong to class M_l with probability $Count(M_l)/\sum_{i=1}^{k} Count(M_i)$ where $Count(M_i)$ is the number of objects in class $M_i, 1 \leq i \leq k$.

2. A decision tree returns a class when it is used to classify an object. A decision tree can thus be regarded as a source of a message "P" or "N," with the information needed to generate this message (for the two-class case) given by

   $$I(p,n) = -\frac{p}{p+n} log_2 \frac{p}{p+n} - \frac{n}{p+n} log_2 \frac{n}{p+n}$$

   In general, for the k-class case, $2 \leq k$, the expected information needed to generate this message is given by

   $$I(Count(M_1), Count(M_2), \ldots, Count(M_k)) = -\sum_{i=1}^{k} (\frac{Count(M_i)}{Sum} log_k \frac{M_i}{Sum})$$

   where $Sum = \sum_{j=1}^{k} M_j$

Figure 3.9: Example decision tree in C4.

Let attribute A having values $\{A_1, A_2, \ldots, A_v\}$ be used for the root of the decision tree. It will partition C into $\{C_1, C_2, \ldots, C_v\}$ where C_i contains those objects in C that have value A_i of A. Let C_i contain p_i objects of class P and n_i of class N. In general, for the k class case C_i will contain $m_{i,j}$ objects of class $j, 1 \leq j \leq k$. The information required for the tree with A as root is then obtained as the weighted average

$$I(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

for the two-class case, and

$$I(A) = \sum_{i=1}^{v} \frac{\sum_{j=1}^{k} m_{i,j}}{\sum_{j=1}^{k} Count_j} I(m_{i,1}, m_{i,2}, \ldots, m_{i,k})$$

where the weight for the i'th branch is the proportion of the objects in C that belong to C_i. The information gained by branching on A is therefore

$$\text{gain}(A) = I(p, n) - I(A)$$

for the two-class case, and

$$\text{gain}(A) = I(m_1, m_2, \ldots, m_k) - I(A)$$

for the k-class case.

C4 examines all candidate attributes and chooses A to maximize gain(A), forms the tree as above, and then uses the same process recursively to form decision trees for the residual subsets $C_1, C_2, \ldots, C_v$.

In our experiments, C4 is given the stenosis as continuous values from zero to a hundred and the expert assessment as a discrete eight-valued set using coding scheme 2 in Table 3.1. C4's assessments are used directly. A single decision tree is produced in C4.

3.5 Summary

This chapter describes the POSCH atherosclerosis data used in the AEPOSCH application. L is a well-established criterion in the pattern-recognition and statistical literature but $\kappa_{b,w}$ is not. Four classifiers, viz., an expert system ESCA, multiple linear regression, back-propagation, and C4 (a C-language implementation of ID3) are described here. The details of how these classifiers are used in this thesis are given. Each classifier is based on a distinct paradigm that is different from the others and is widely used in academic and real-world settings. Use of real-world classifiers and real-world data enables one to collect data having probability distributions that are more representative of a real-world problem, than if a simplistic idealized model is used. A simplistic, idealized model of data is used in the next chapter to gain some insight into techniques introduced to compute the value of a criterion over a dataset along with the standard error.

i	arterial segment (in film)	i	arterial segment (in film)
1	right-ostium (a)	2	right-ostium (b)
3	proximal-right-coronary (a)	4	proximal-right-coronary (b)
5	mid-right-coronary (a)	6	mid-right-coronary (b)
7	distal-right-coronary (a)	8	distal-right-coronary (b)
9	proximal-posterior-descending (a)	10	proximal-posterior-descending (b)
11	distal-posterior-descending (a)	12	distal-posterior-descending (b)
13	left-ostium (a)	14	left-ostium (b)
15	left-main-coronary (a)	16	left-main-coronary (b)
17	proximal-lad[3] (a)	18	proximal-lad (b)
19	mid-lad (a)	20	mid-lad (b)
21	distal-lad (a)	22	distal-lad (b)
23	proximal-left-circumflex (a)	24	proximal-left-circumflex (b)
25	mid-left-circumflex (a)	26	mid-left-circumflex (b)
27	distal-left-circumflex (a)	28	distal-left-circumflex (b)
29	first-diagonal-lad (a)	30	first-diagonal-lad (b)
31	second-diagonal-lad (a)	32	second-diagonal-lad (b)
33	third-diagonal-lad (a)	34	third-diagonal-lad (b)
35	first-obtuse-marginal (a)	36	first-obtuse-marginal (b)
37	second-obtuse-marginal (a)	38	second-obtuse-marginal (b)
39	third-obtuse-marginal (a)	40	third-obtuse-marginal (b)
41	other-segment-a (a)	42	other-segment-a (b)
43	other-segment-b (a)	44	other-segment-b (b)
45	other-segment-c (a)	46	other-segment-c (b)

Table 3.5: Mapping of variable numbers to arterial segments in multiple linear regression, back-propagation, and C4.

Number.	Attributes			windy	Class
	outlook	temperature	humidity		
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	overcast	hot	high	false	P
4	rain	mild	high	false	P
5	rain	cool	normal	false	P
6	rain	cool	normal	true	N
7	overcast	cool	normal	true	P
8	sunny	mild	high	false	N
9	sunny	cool	normal	false	P
10	rain	mild	normal	false	P
11	sunny	mild	normal	true	P
12	overcast	mild	high	true	P
13	overcast	hot	normal	false	P
14	rain	mild	high	true	N

Table 3.6: An example training set for C4.

Chapter 4

Estimating the value of a criterion and its standard error

This chapter discusses various methods of evaluating performance criteria for a classifier based on multinomial results, the bootstrap [18, 17, 19], and the grouped jackknife [53]. It is proved that there is a deficiency in the bootstrap procedure described in [17, 19] for estimating L for classifiers. To maximize the utilization of the available cases, a method for splitting and recombining the available cases of a sample is provided. This method is an extension of the cross-validation procedure [46, 75, 29, 79]. A simulation is given to demonstrate the plausibility of this method.

A point estimate of the criterion θ is $\hat{\theta}$. An estimate of $\sigma_{\hat{\theta}}$ from a sample is $\hat{\sigma}_{\hat{\theta}}$. $\hat{\sigma}_{\hat{\theta}}$ may be written as $SE(\theta)$ [45, 14]. Many point estimates are approximately normally distributed. In such a case, knowing $\hat{\theta}$ and $\hat{\sigma}_{\hat{\theta}}$ we can compute confidence intervals for θ and test the null hypothesis $H_0 : \theta \leq \theta_{\min}$ versus the alternative hypothesis $H_a : \theta > \theta_{\min}$ to determine the acceptability of a classifier, where $\theta_{\min}$ is the minimum value of θ that is considered acceptable. This chapter discusses methods of computing $\hat{\theta}$ and $\hat{\sigma}_{\hat{\theta}}$ using the split recombination conjecture coupled with multinomial results, the grouped jackknife, and the bootstrap [17, 19]. Four new methods for estimating $\hat{\theta}$ and $SE(\theta)$ are introduced in this section. They are the method of repeated random halving (RRH), the split multinomial method (SM), the split grouped jackknife (SGJK), and the split bootstrap (SB) technique.

4.1 Evaluation considering the influence of the number of cases

An estimate $\hat{\theta}$ of a parameter or criterion θ, where θ can be $L, \kappa_{b,w}$, or any other quantity describing the performance of a classifier is always obtained using a sample set of finite size N. The technique by which the sample is obtained is critical [7]. We will assume that the sample at hand is a random sample from a population.

If N is large we have more confidence that the estimate $\hat{\theta}$ is close to the true (population) value of θ than if N is small. This statement is given a more precise meaning by using the concept of a confidence interval. When the statistic is an average computed from a random sample from a population with standard deviation σ, the Central Limit Theorem [14] implies that, for "large N," the $100(1-\alpha)\%$ confidence interval (C.I.) of a point estimate $\hat{\theta}$ is

approximately given by

$$(\hat{\theta} - z_{\alpha/2}\frac{\sigma}{\sqrt{N}}, \hat{\theta} + z_{\alpha/2}\frac{\sigma}{\sqrt{N}})$$

where the normal probability point z_p satisfies $P(z > z_p) = p$. A $100(1-\alpha)\%$ C.I. signifies that in $100(1-\alpha)\%$ of experiments the random $100(1-\alpha)$ confidence interval about $\hat{\theta}$ will contain the true value of θ. As N increases, the width of this interval decreases. In most cases, σ must be estimated by the sample standard deviation $\hat{\sigma}_{\hat{\theta}}$. What constitutes "large N" depends on the situation. Often $N > 25$ is sufficiently large.

More generally, even when $\hat{\theta}$ is not a mean it is often the case that $\hat{\theta}$ is asymptotically normal[1] with mean θ and standard error $\sigma_{\hat{\theta}}$ which decreases as N increases, typically at the same rate as $1/\sqrt{N}$. In such a case, a $100(1-\alpha)\%$ confidence interval is given by

$$(\hat{\theta} - z_{\alpha/2}\hat{\sigma}_{\hat{\theta}}, \hat{\theta} + z_{\alpha/2}\hat{\sigma}_{\hat{\theta}})$$

Thus a fundamental problem for a statistical evaluation of classifier performance will often be estimating the standard error of an estimated criterion.

If a minimum value $\theta_{\min}$ is specified for the performance of a classifier we can determine to what extent a classifier satisfies the requirement by using hypothesis testing and p-values. To carry out an hypothesis test we must first define a null hypothesis which is the opposite of what we want to prove. Since we will usually want to prove that our classifier's performance is greater than the minimum required, we establish the null hypothesis, denoted H_0, to be

$$H_0 : \theta \leq \theta_{\min}$$

The alternative hypothesis we wish to establish is opposite to the null hypothesis. In this case it is

$$H_a : \theta > \theta_{\min}$$

A test statistic has then to be chosen. In our case, for large samples the test statistic is

$$z = \frac{\hat{\theta} - \theta_{\min}}{\hat{\sigma}_{\hat{\theta}}}.$$

To show that a performance criterion of one classifier (θ_a) is higher than that of another classifier (θ_b) we set up hypotheses as follows

$$H_0 : \theta_a \leq \theta_b$$

$$H_a : \theta_a > \theta_b$$

---

[1] $\hat{\theta}_N$ is asymptotically normal (AN) [37] if there exist constants a_N and b_N depending on sample size N such that

$$\frac{\hat{\theta}_N - a_N}{b_N} \sim N(0,1) \text{ as } N \to \infty$$

For the multi-variate case, $\hat{\theta}$ is AN if there exists a constant vector a_N and a constant matrix b_N depending on sample size N such that

$$b_N^{-1}(\hat{\theta} - a_N) \sim N(0, I) \text{ as } N \to \infty$$

where $N(0, I)$ is the multi-variate normal distribution with mean 0 and co-variance matrix I. In layman's terms, asymptotically normal means that as the size of the sample increases, the distribution is closer to normal.

and a test statistic is[2]

$$z = \frac{\hat{\theta}_a - \hat{\theta}_b}{\hat{\sigma}[\hat{\theta}_a - \hat{\theta}_b]}$$

When $\hat{\theta}_a$ and $\hat{\theta}_b$ are independent

$$SE[\theta_a - \theta_b] = \hat{\sigma}[\hat{\theta}_a - \hat{\theta}_b] = \sqrt{\hat{\sigma}[\hat{\theta}_a]^2 + \hat{\sigma}[\hat{\theta}_b]^2}$$

A sufficiently large value of z would lead us to discredit H_0 in favor of H_a.

The values of θ and $\sigma_{\hat{\theta}}$ depend on the population from which data is selected. Thus a more complete notation would be $\theta(F)$ and $\sigma(F)$ where F is the probability distribution characterizing the population.

4.2 Theoretically derived estimates for a function of random variables

To compute the standard error of a function of random variables we can often make use of the following corollary (Page 124 of [65]).

Suppose that $\boldsymbol{X}_n = (X_{n1}, \ldots, X_{nk})$ *is* $AN(\boldsymbol{\mu}, n^{-1}\boldsymbol{\Sigma})$, *with* $\boldsymbol{\Sigma} = [\sigma_{ij}]$ *a* $k \times k$ *covariance matrix. Let* $g(\boldsymbol{x})$ *be a real-valued function having a nonzero differential at* $\boldsymbol{x} = \boldsymbol{\mu}$. *Then*

$$g(\boldsymbol{X}_n) \textit{is } AN(g(\boldsymbol{\mu}), \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{k}\sigma_{ij}\frac{\partial g}{\partial x_i}\mid_{\boldsymbol{x}=\boldsymbol{\mu}} \cdot \frac{\partial g}{\partial x_j}\mid_{\boldsymbol{x}=\boldsymbol{\mu}})$$

Therefore, the approximate standard deviation of $g(\boldsymbol{\mu})$ *is given by*

$$\sqrt{\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{k}\sigma_{ij}\frac{\partial g}{\partial x_i}\mid_{\boldsymbol{x}=\boldsymbol{\mu}} \cdot \frac{\partial g}{\partial x_j}\mid_{\boldsymbol{x}=\boldsymbol{\mu}}}$$

Remarks:

1. A sufficient condition for g to have a nonzero differential at $\boldsymbol{\mu}$ is that the first partial derivatives $\partial g/\partial x_i, 1 \leq i \leq k$, be continuous at $\boldsymbol{\mu}$ and not all zero at $\boldsymbol{\mu}$.

2. To obtain the asymptotic normality of $g(X_{n1}, \ldots, X_{nk})$ the asymptotic *joint normality* of $X_{n1}, \ldots, X_{nk}$ is needed.

Under fairly general conditions the sample mean, the sample variance, and the sample median are asymptotically normal. The maximum and the minimum are not.

[2] For typographical reasons we denote $\hat{\theta_a - \theta_b}$ as $\hat{\theta}_a - \hat{\theta}_b$

4.3 The resubstitution method, the holdout method, and cross-validation

Let $\boldsymbol{\mathcal{X}} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ be the set of N pattern samples available. Each pattern sample $\boldsymbol{X}_i = (\boldsymbol{T}_i, \boldsymbol{Y}_i)$ where $\boldsymbol{T}_i$ is the training or measurement information and $\boldsymbol{Y}_i$ is the class of the i'th sample. Let

$N_{\boldsymbol{\mathcal{X}}_1,\boldsymbol{\mathcal{X}}_2}$ = number of errors of a classifier trained on $\boldsymbol{\mathcal{X}}_1$ when applied to $\boldsymbol{\mathcal{X}}_2$.

$N_{\boldsymbol{\mathcal{X}}_2}$ = number of samples in $\boldsymbol{\mathcal{X}}_2$.

$$\hat{P}_{\boldsymbol{\mathcal{X}}_1,\boldsymbol{\mathcal{X}}_2} = \frac{N_{\boldsymbol{\mathcal{X}}_1,\boldsymbol{\mathcal{X}}_2}}{N_{\boldsymbol{\mathcal{X}}_2}}$$

$\hat{P}_{\boldsymbol{\mathcal{X}}_1,\boldsymbol{\mathcal{X}}_2}$ is the error rate for a classifier trained on $\boldsymbol{\mathcal{X}}_1$ and tested on $\boldsymbol{\mathcal{X}}_2$.

More generally, $\hat{\theta}_{\boldsymbol{\mathcal{X}}_1,\boldsymbol{\mathcal{X}}_2}$ is the value estimated for θ over $\boldsymbol{\mathcal{X}}_2$ for a classifier trained on $\boldsymbol{\mathcal{X}}_1$.

The method used for a long time for estimating classification error rates was the resubstitution method. In the resubstitution method, the entire set of available samples is first used for training (or design) of the classifier and the performance of the classifier on unseen cases is predicted to be that achieved on this training set. That is, error probability is estimated as $\hat{P}_{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{X}}}$. This procedure results in a optimistic estimate of the performance known as the *resubstitution error rate* or *apparent error rate* [44, 79, 34, 27, 28].

The holdout method can be described as follows [79].

1. Partition $\boldsymbol{\mathcal{X}}$ into two mutually exclusive sets $\boldsymbol{\mathcal{X}}_\alpha$ and $\boldsymbol{\mathcal{X}}_\beta$ such that $\boldsymbol{\mathcal{X}}_\alpha \cup \boldsymbol{\mathcal{X}}_\beta = \boldsymbol{\mathcal{X}}$.
2. Train the classifier on $\boldsymbol{\mathcal{X}}_\alpha$.
3. Test the classifier on $\boldsymbol{\mathcal{X}}_\beta$. $\hat{P}_{\boldsymbol{\mathcal{X}}_\alpha,\boldsymbol{\mathcal{X}}_\beta}$ is the Holdout error rate. In general, $\hat{\theta}_{\boldsymbol{\mathcal{X}}_\alpha,\boldsymbol{\mathcal{X}}_\beta}$ is the Holdout estimate of θ.

Traditionally half of the available samples have been used for training and the other half used for testing. Fukunaga [27, 28] has shown that this is a good strategy in the absence of any other information because the bias of the estimates depends only on the size of the training set, and the variance of the estimates depend mainly on the size of the test set. The result of the holdout method has a pessimistic bias because all the samples are not used for training as they would be when the classifier is applied in practise. Duda and Hart [16] describe a variant of the Holdout method that is known as the *rotation* or *data shuffling* method. It can be described as [79]:

1. Partition $\boldsymbol{\mathcal{X}}$ into K randomly chosen pairs $\{\boldsymbol{\mathcal{X}}^i_\alpha, \boldsymbol{\mathcal{X}}^i_\beta\}$ of sets of equal size with $\boldsymbol{\mathcal{X}}^i_\alpha \cup \boldsymbol{\mathcal{X}}^i_\beta = \boldsymbol{\mathcal{X}}, i = 1, \ldots, K$ such that for $i = 1, \ldots, K$, $\boldsymbol{\mathcal{X}}^i_\alpha$ and $\boldsymbol{\mathcal{X}}^i_\beta$ are mutually exclusive.
2. For $i = 1, \ldots, K$, train the classifier on $\boldsymbol{\mathcal{X}}^i_\alpha$ and test it on $\boldsymbol{\mathcal{X}}^i_\beta$. The resulting proportion of errors is $\hat{P}_{\boldsymbol{\mathcal{X}}^i_\alpha,\boldsymbol{\mathcal{X}}^i_\beta}$

3. The rotation error rate is given by

$$\frac{1}{K}\sum_{i=1}^{K}\hat{P}_{\boldsymbol{\mathcal{X}}^i_\alpha,\boldsymbol{\mathcal{X}}^i_\beta}$$

More generally, the rotation estimate of a criterion θ is given by

$$\hat{\theta} = \frac{1}{K}\sum_{i=1}^{K}\hat{\theta}_{\boldsymbol{\mathcal{X}}^i_\alpha,\boldsymbol{\mathcal{X}}^i_\beta}$$

Although this procedure uses the data more thoroughly than the Holdout method, it still uses only half the data for training each time. The final result is still overly pessimistic.

The holdout, or the H method, is a special case of cross-validation. Among the earliest articles on cross-validation is that by Lachenbruch and Mickey [46]. M. Stone [75] and S. Geisser [29] discuss cross-validation in a wider context. Cross-validation uses the data more efficiently and gives estimates of the performance that are less biased compared to other methods. One form of cross-validation is also known in the literature as leave-1-out, or the U method [79, 46, 6, 1, 73]. Leave-k-out is another form of cross-validation. Cross-validation can be described as follows:

1. Partition the set $\boldsymbol{\mathcal{X}}$ into non-overlapping sets $\boldsymbol{\mathcal{X}}^1_\alpha, \ldots, \boldsymbol{\mathcal{X}}^g_\alpha$, $i = 1, \ldots, g$, and let $\boldsymbol{\mathcal{X}}^i_\beta = \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^i_\alpha$. For simplicity let us assume that N is a multiple of K. $\boldsymbol{\mathcal{X}}^1_\alpha, \ldots, \boldsymbol{\mathcal{X}}^g_\alpha$ are of equal size ($k = \frac{N}{g}$) and are mutually disjoint.

2. Train the classifier on $\boldsymbol{\mathcal{X}}^i_\alpha$.

3. Test the classifier on $\boldsymbol{\mathcal{X}}^i_\beta$.

4. Compute the error rate $\hat{P}^i_e = \hat{P}_{\boldsymbol{\mathcal{X}}^i_\alpha,\boldsymbol{\mathcal{X}}^i_\beta}$.

5. Do steps 2, 3, and 4 for $i = 1, \ldots, g$ to obtain values for $P^i_e, i = 1, \ldots, g$.

6. The cross-validation estimate of error, $\hat{P}^{CV}_e$ is given by

$$\hat{P}^{CV}_e = \frac{1}{N}\sum_{i=1}^{g}\hat{P}^i_e$$

The cross-validation estimate of a general criteria θ cannot be computed by this procedure and needs the SR conjecture (Section 4.6.3) introduced in this thesis.

B. Efron [18] shows that for small samples of size less than 20 using a Gaussian classifier, the leave-1-out method has a small bias in the error rate but a large variance in the estimates. The bootstrap technique is one technique that has a smaller variance than the leave-1-out method [18, 19, 17]. Other techniques have also been developed [17] that have better properties than the bootstrap. The search for a better technique continues [82, 73] as difficulties with the older techniques are found [6, 1].

4.4 Obtaining performance estimates using the grouped jackknife

Consider a sample S of n cases. In the grouped jackknife we split the sample into g groups each of size h, $n = gh$, where n is the number of cases. For the remaining discussion, we will assume that if n, the number of cases in S, is not a multiple of g then some of the cases are discarded. For $h = 1$, the grouped jackknife reduces to the jackknife.

The grouped jackknife [53] allows us to estimate the mean, the bias, and standard error of a function of independent and identically distributed random variables or vectors. Let $Y_1, \ldots, Y_n$ be a sample of independent and identically distributed random variables. Let $\hat{\theta}$ be an estimator of a parameter θ based on the sample of size n. Let $\hat{\theta}_{-i}$ be the corresponding estimator based on the sample set where the i'th group has been deleted. The size of this sample set is $(g-1)h = n - h$. Define

$$\tilde{\theta}_i = g\hat{\theta} - (g-1)\hat{\theta}_{-i}, i = 1, \ldots, g$$

The estimator

$$\tilde{\theta} = \frac{1}{g}\sum_{i=1}^{g}\tilde{\theta}_i \tag{4.1}$$

has the property that it eliminates the order $\frac{1}{n}$ term from a bias of the form

$$E(\hat{\theta}) = \theta + a_1/n + O(1/n^2)$$

so that $E[\tilde{\theta}_i] = \theta + a_2/n^2 + O(1/n^3)$. An estimated standard error is given by

$$\hat{\sigma}_{\hat{\theta}} = \left\{ \frac{1}{g(g-1)} \sum_{i=1}^{g} (\tilde{\theta}_i - \tilde{\theta})^2 \right\}^{\frac{1}{2}} \tag{4.2}$$

If we set $h = 1$, the grouped jackknife reduces to the (ungrouped) jackknife [17].

4.5 The Bootstrap algorithm to estimate the characteristics of the sampling distribution of a statistic

The bootstrap is a computer-based method that substitutes Monte Carlo simulation involving considerable amounts of computation in place of theoretical analysis. This Monte Carlo simulation involves repeatedly drawing *bootstrap sample sets* with re-sampling from the given sample set. It is assumed in the bootstrap that when a bootstrap replication sample is drawn from the entire available sample, then the relation of the bootstrap replication sample to the entire available sample is analogous to the relation of the entire available sample to the population. Since we can expect a large sample to be similar to the population from which it is drawn, the properties of estimators based on a bootstrap replication sample will be similar to the properties of an estimator based on a sample from the population. On this assumption the Monte Carlo simulation is based.

Suppose that our data consists of a random sample from an unknown probability distribution F on the real line,

$$X_1, X_2, \ldots, X_n \sim F$$

Let $\hat{F}$ indicate the empirical probability distribution

$$\hat{F}: \text{ probability } \frac{1}{n} \text{ on } x_1, x_2, \ldots, x_n.$$

We expect that $\hat{F} \cong F$. To estimate the statistical standard error, $\sigma(\theta(F))$ of an arbitrary criterion θ, where θ can be any criterion such as the median, is not an easy task in general because $\sigma(\theta(F))$ cannot be expressed as a simple function of the sampling distribution F. Using the computer we can always numerically evaluate the bootstrap estimate $\hat{\sigma} = \sigma(\hat{F})$, without knowing a simple expression for σ. The evaluation of $\hat{\sigma}$ is a straightforward Monte Carlo exercise as below.

Let us have a sample $\boldsymbol{\mathcal{X}}$ of n cases $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ drawn from a population Ω. Each case $\boldsymbol{x}_i$ is a vector of numbers $(x_{i1}, x_{i2}, \ldots, x_{ik}), i = 1, \ldots, n$. It is possible that one of the x_{ij} for each sample will be a classifier evaluation. This evaluation could be a result of classifying the sample based on information represented partially or wholly in the remaining numbers. In the present case we are considering that these samples are evaluated by a classifier such as a human being or an expert system that does not require training. The case where the classifier requires training is discussed later (Sections 4.5.2 and 4.6).

The bootstrap may be used to estimate $\sigma_{\hat{\theta}}$ and the bias $E[\hat{\theta}] - \theta$, where $\hat{\theta} = \hat{\theta}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \hat{\theta}(S)$ is an estimator based on sample S. The bootstrap consists of the following steps [17, 19]. Let $\sigma(F)$ be the standard deviation of $\hat{\theta}(\boldsymbol{\mathcal{X}})$ where $\boldsymbol{\mathcal{X}}$ is a sample from the distribution F, the probability distribution describing the statistic.

1. Define $\hat{F} = \hat{F}_n$ to be the empirical distribution of the sample, that is,

   $$\hat{F}: \text{ probability } \frac{1}{n} \text{ on } \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n.$$

   $\hat{F}$ is the nonparametric Maximum Likelihood Estimate (MLE) of F.

2. Draw a bootstrap sample set $\boldsymbol{\mathcal{X}}^*$, that is, a random sample from $\hat{F}$.

   $$\boldsymbol{\mathcal{X}}^* = \{\boldsymbol{X}_1^*, \ldots, \boldsymbol{X}_n^*\} \sim_{iid} \hat{F}$$

   where *iid* stands for independent and identically distributed. This amounts to taking a random sample with replacement of size n from the finite population $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. For reference below define the probability distribution

   $$F^*: \text{ probability } \frac{1}{n} \text{ at } \boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_n^*$$

3. Compute $\hat{\theta}^* = \theta(F^*)$, that is, compute $\hat{\theta}$ based on $\boldsymbol{\mathcal{X}}^*$ in the same way $\hat{\theta}$ was computed from $\boldsymbol{\mathcal{X}}$.

4. Independently repeat steps 2 and 3 a large number B of times, obtaining bootstrap replications,

   $$\hat{\theta}^{*1}, \hat{\theta}^{*2}, \ldots, \hat{\theta}^{*B}$$

   These are a random sample of $\hat{\theta}$ when $\boldsymbol{\mathcal{X}}^*$ has distribution $\hat{F}$.

5. Define

$$\hat{\theta}^{*\cdot} = \sum_{b=1}^{B} \hat{\theta}^{*b}/B \tag{4.3}$$

Compute the bootstrap estimate of $\sigma_{\hat{\theta}}$ as

$$\hat{\sigma}_B = \hat{\sigma} = \left\{ \frac{1}{B-1} \sum_{b=1}^{B} [\hat{\theta}^{*b} - \hat{\theta}^{*\cdot}]^2 \right\}^{\frac{1}{2}} \rightarrow \sigma(\hat{F}), \text{as } B \rightarrow \infty \tag{4.4}$$

where $\sigma(\hat{F})$, is the standard deviation of $\hat{\theta}$ when sampling from $\boldsymbol{\mathcal{X}}$.

6. Compute the bootstrap estimate of bias as

$$\widehat{bias} = \hat{\theta}^{*\cdot} - \hat{\theta}$$

A bias-corrected estimate of bias of θ is then

$$\hat{\theta}_{BS} = \hat{\theta} - \widehat{bias} = 2\hat{\theta} - \hat{\theta}^{*\cdot}$$

Of course, if it is known that $\hat{\theta}$ is unbiased, there is no need to estimate bias. For instance, L is unbiased, while $\kappa_{b,w}$ is biased. Even if $\hat{\theta}$ is biased, the additional variability introduced by the estimate of bias will increase the mean squared error (MSE),

$$MSE = E[(\hat{\theta} - \theta)^2]$$

and it may be preferable to omit the bias correction.

4.5.1 How large should B be?

As $B \rightarrow \infty, \hat{\sigma}_B$ estimates $\sigma(\hat{F})$ perfectly but $\sigma(\hat{F})$ depends on the particular set of data drawn from the population. Hence there is not much point in using a very large value of B. The coefficient of variation (CV) of a random variable T is defined when T can assume positive values only [63]. It is given by

$$CV(T) = \frac{\sigma_T}{E[T]}$$

Since $\hat{F}$ is a random estimate of F, $\hat{\sigma} = \hat{\sigma}(\hat{F})$ will have a non-zero coefficient of variation for estimating the true standard error $\sigma = \sigma(F)$. Moreover, $\hat{\sigma}_B$ will have a still larger CV because of the randomness added by the Monte Carlo bootstrap sampling [20]. The following approximation is given in [20],

$$CV(\hat{\sigma}_B) \doteq \left\{ CV(\hat{\sigma})^2 + \frac{E[\hat{\delta}] + 2}{4B} \right\}^{1/2}, \tag{4.5}$$

where $\hat{\delta}$ is the estimated kurtosis of the bootstrap distribution of $\hat{\theta}^*$, given the data $\boldsymbol{y} = (x_1, x_2, \ldots, x_n)$. $E[\hat{\delta}]$ is the expected value of the kurtosis averaged over $\boldsymbol{y}$. The kurtosis is defined by [63],

$$\delta = \frac{E[(\hat{\theta} - E[\hat{\theta}])^4]}{V[\hat{\theta}]^2} - 3$$

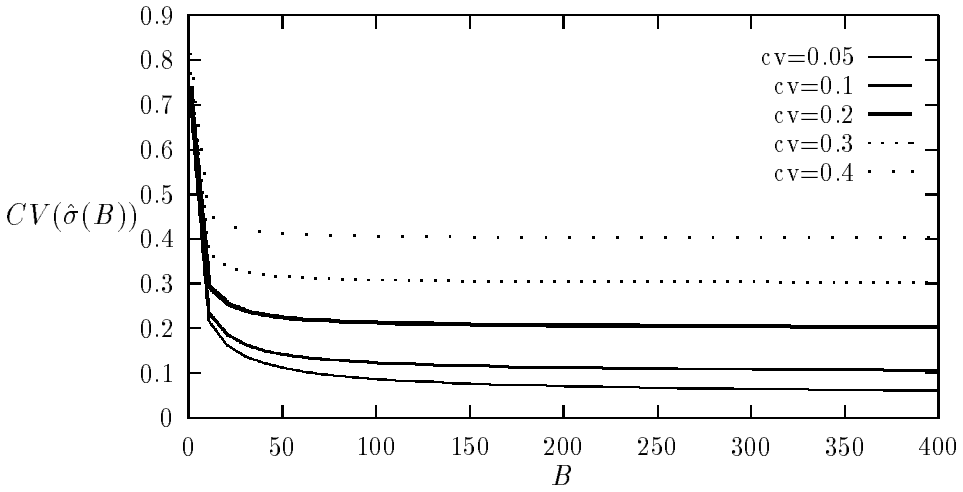

Figure 4.1: $CV(\hat{\sigma}_B)$ vs. B for kurtosis = 0.
In the figure, cv gives the value of $CV(\hat{\sigma})$. For instance, "cv=0.4" means that $CV(\hat{\sigma}) = 0.4$.

where $V[\hat{\theta}]$ is the variance of $\hat{\theta}$. If we have B bootstrap replications, the kurtosis can be estimated by

$$\hat{\delta} = \frac{n \sum_{i=1}^{B} (\hat{\theta}_i^* - \hat{\theta}^{*\cdot})^4}{\sum_{i=1}^{B} \left[(\hat{\theta}_i^* - \hat{\theta}^{*\cdot})^2 \right]^2} - 3$$

In Equation 4.5, if the ratio of the second term to $CV(\hat{\sigma})^2$ is small for a particular B, there is little reason to use a larger B.

In the bootstrap, F^* parallels $\hat{F}$. Since we usually expect (for large samples at least) that $\hat{F} \cong F$, we expect that the properties of an estimator $\theta(F^*)$ such as bias and standard error when doing bootstrap replications from $\hat{F}$ are similar to the properties of $\theta(\hat{F})$ when doing repeated random sampling from the population. $\hat{\theta} = \theta(\hat{F})$ bears the same relationship to $\hat{F}$ as $\theta = \theta(F)$ does to F, and the sampling distribution of $\hat{\theta} = \theta(\hat{F})$ is paralleled by the sampling distribution of $\hat{\theta}^* = \theta(F^*)$. This is useful in the bootstrap because $\hat{F}$ is similar to F, and F^* is similar to $\hat{F}$. $\theta(F)$ parallels $\theta(\hat{F})$, $\sigma_{\hat{\theta}}(F)$ parallels $\sigma_{\hat{\theta}}(\hat{F})$, and $E_F[\hat{\theta}]$ parallels $E_{\hat{F}}[\theta(F^*)]$.

Plots of $CV(\hat{\sigma}_B)$ vs. B for different values of $CV(\hat{\sigma})$ and kurtosis are given in Fig. 4.1 to Fig. 4.4. From the plots we can see that, for values of $CV(\hat{\sigma}) > .10$, and assuming $E[\hat{\delta}] = 0$, there is little improvement past $B = 100$. In fact, B as small as 25 gives reasonable results. For typical situations, $CV(\hat{\sigma})$ lies between .10 and .30. For example, if $\hat{\theta} = \bar{x}, n = 20, x_i \sim_{iid} N(0,1)$, then $CV(\hat{\sigma}) \doteq .16$ [20].

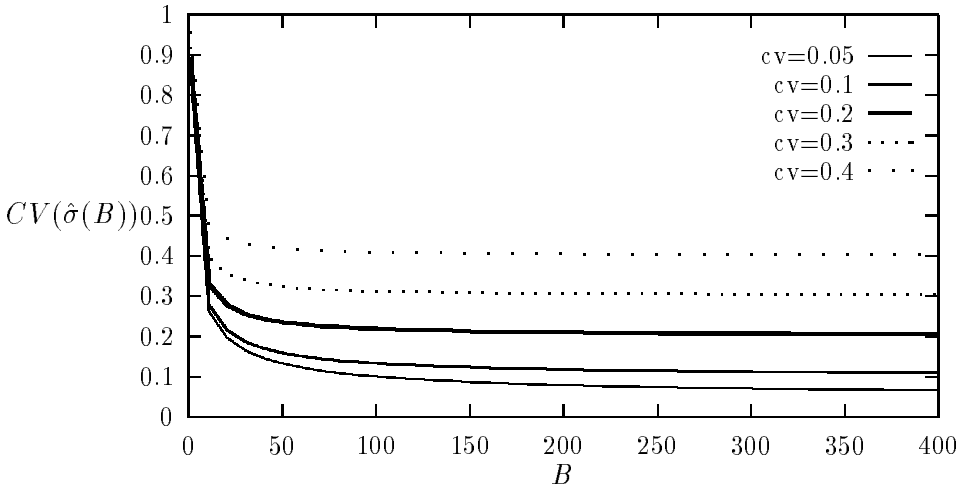

Figure 4.2: $CV(\hat{\sigma}_B)$ vs. B for kurtosis = 1.
In the figure, cv gives the value of $CV(\hat{\sigma})$. For instance, "cv=0.4" means that $CV(\hat{\sigma}) = 0.4$.

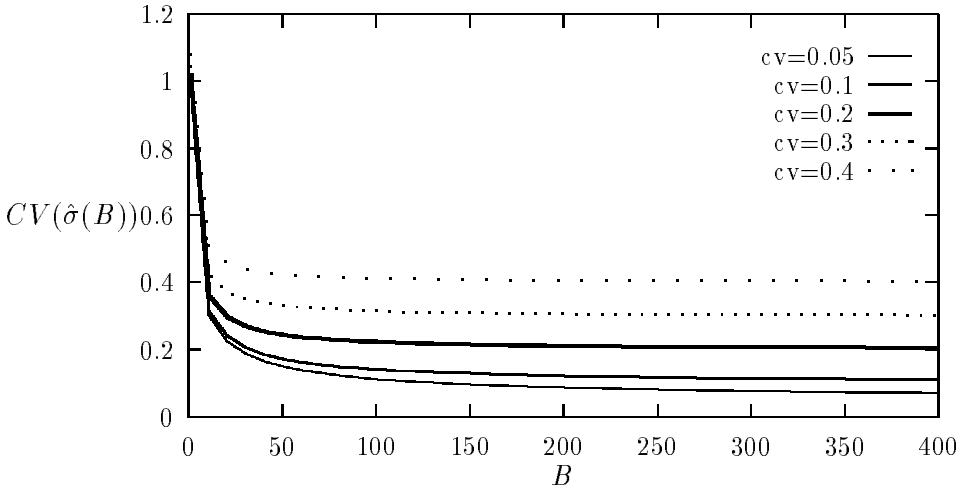

Figure 4.3: $CV(\hat{\sigma}_B)$ vs. B for kurtosis = 2.
In the figure, cv gives the value of $CV(\hat{\sigma})$. For instance, "cv=0.4" means that $CV(\hat{\sigma}) = 0.4$.

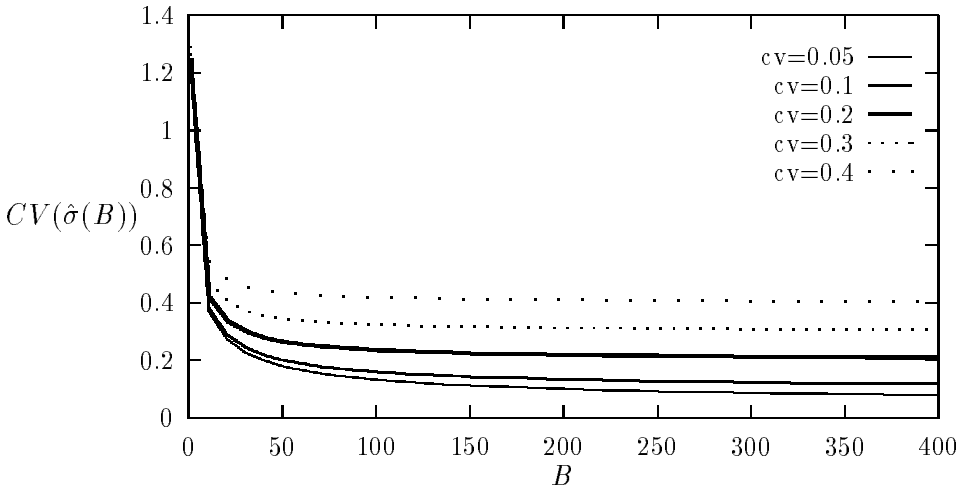

Figure 4.4: $CV(\hat{\sigma}_B)$ vs. B for kurtosis = 4.
In the figure, cv gives the value of $CV(\hat{\sigma})$. For instance, "cv=0.4" means that $CV(\hat{\sigma}) = 0.4$.

4.5.2 Inaccuracy in computing Expected Excess Error by the bootstrap

If we have a trainable classifier, then the data in the problem consists of pairs, that is,

$$\boldsymbol{x}_1 = (\boldsymbol{t}_1, y_1), \boldsymbol{x}_2 = (\boldsymbol{t}_2, y_2), \ldots, \boldsymbol{x}_n = (\boldsymbol{t}_n, y_n)$$

where $\boldsymbol{t}_i$ is a p-dimensional vector of predictors and y_i is a real-valued variable. We assume that the data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are observations on n independent and identically distributed (iid) random variables $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ with $\boldsymbol{X}_i = (\boldsymbol{T}_i, Y_i)$ defined on $C = R^{p+1}$ with common unknown distribution F, that is,

$$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \sim_{iid} F \tag{4.6}$$

The goal is to be able to predict Y from $\boldsymbol{T}$ when $\boldsymbol{X}$ has not been observed before (during training). We assume here for simplicity that the Y_i can take on only two possible values, say 0 or 1. $\boldsymbol{\mathcal{X}} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$ are collectively called *the training set.* A formula (or algorithm) η is available for constructing *a prediction rule* $\eta_{\boldsymbol{\mathcal{X}}}(.)$ from the training set $\boldsymbol{\mathcal{X}}$. The notation $\eta_{\boldsymbol{\mathcal{X}}}(.)$ indicates that the prediction rule depends on both the training algorithm η and the training set $\boldsymbol{\mathcal{X}}$. Given a new randomly selected vector $\boldsymbol{X}$ where $\boldsymbol{X} = (\boldsymbol{T}, Y)$, $\eta_{\boldsymbol{\mathcal{X}}}(.)$ can be applied to $\boldsymbol{T}$ to produce the value $\eta_{\boldsymbol{\mathcal{X}}}(\boldsymbol{T})$. The value $\eta_{\boldsymbol{\mathcal{X}}}(\boldsymbol{T})$ is intended to predict the response Y corresponding to the predictor vector $\boldsymbol{T}$. The *true error rate* $\mathrm{Err}(\boldsymbol{\mathcal{X}}, F)$ is the expected value

$$\mathrm{Err}(\boldsymbol{\mathcal{X}}, F) \equiv E_F Q[Y, \eta_{\boldsymbol{\mathcal{X}}}(\boldsymbol{T})] \tag{4.7}$$

the expectation being taken over $(\boldsymbol{T}, Y) \sim F$, with $\boldsymbol{\mathcal{X}}$ fixed at its observed value. $Q[y_1, y_2]$ is the error indicator

$$Q[y_1, y_2] \quad = \quad 0 \text{ if } y_1 = y_2$$

$$= \quad 1 \text{ if } y_1 \neq y_2$$

The apparent error rate $err(\boldsymbol{\mathcal{X}})$ is the sample proportion of errors

$$err(\boldsymbol{\mathcal{X}}) = \text{Err}(\boldsymbol{\mathcal{X}}, \hat{F}) = \frac{1}{n}\sum_{i=1}^{n} Q[y_i, \eta_{\boldsymbol{\mathcal{X}}}(\boldsymbol{T}_i)] \tag{4.8}$$

err is usually an underestimate of Err because $\eta_{\boldsymbol{\mathcal{X}}}(.)$ has been optimized on the observed sample $\boldsymbol{\mathcal{X}}$. Let $op(\boldsymbol{\mathcal{X}}, F)$, "op" being short for optimism, be defined as

$$op(\boldsymbol{\mathcal{X}}, F) \equiv \text{Err}(\boldsymbol{\mathcal{X}}, F) - \text{Err}(\boldsymbol{\mathcal{X}}, \hat{F})$$

As $\boldsymbol{\mathcal{X}}$ varies in repeated sampling, $op(\boldsymbol{\mathcal{X}}, F)$ is a random variable. Let ω be the expectation of $op(S, F)$ as $\boldsymbol{\mathcal{X}}$ is sampled from F^n. ($\boldsymbol{\mathcal{X}}$ is sampled from F^n because $\boldsymbol{\mathcal{X}}$ consists of n independent random vectors).

$$\omega(F) \equiv E_{F^n}\ \text{op}(\boldsymbol{\mathcal{X}}, F) = E_{F^n}\{\text{Err}(\boldsymbol{\mathcal{X}}, F) - \text{Err}(\boldsymbol{\mathcal{X}}, \hat{F})\} \tag{4.9}$$

Let $\boldsymbol{\mathcal{X}}^*$ indicate a random sample of size n from $\hat{F}$,

$$\boldsymbol{\mathcal{X}}_1^*, \ldots, \boldsymbol{\mathcal{X}}_n^* \sim_{iid} \hat{F} \tag{4.10}$$

and E_* indicate expectation with respect to the random mechanism 4.10, $\hat{F}$ fixed at its observed value. Then, if $P_i^* \equiv$ proportion of case $\boldsymbol{x}_i$ in the bootstrap sample $\boldsymbol{\mathcal{X}}^*$, the bootstrap estimate of $\omega = E(op(\boldsymbol{\mathcal{X}}, F))$ is given by

$$\begin{aligned}
\hat{\omega}^{(BOOT)} &= \omega(\hat{F}) \\
&= E_* op(\boldsymbol{\mathcal{X}}^*, \hat{F}) \\
&= E_* \sum_{i=1}^{n} (\frac{1}{n} - P_i^*) Q[y_i, \eta_{\boldsymbol{\mathcal{X}}^*}(\boldsymbol{T}_i)] \\
&= E_*\{(\text{error of } \eta_{\boldsymbol{\mathcal{X}}^*}(\boldsymbol{T}_i) \text{ in set of all cases } \boldsymbol{\mathcal{X}})\} \\
&\quad -(\text{error in } \eta_{\boldsymbol{\mathcal{X}}^*}(\boldsymbol{T}) \text{ in } \boldsymbol{\mathcal{X}}^*)
\end{aligned}$$

One way in which the bootstrap replication differs from the repeated sampling case, is that the test sample will almost certainly have cases in common with the training sample. In the bootstrap the fraction of cases in the finite population that occur in any bootstrap replication sample is the same as the fraction of unique cases in a bootstrap sample. This is because the size of the bootstrap sample (n) is the same as the size of the finite population. We can compute the expected fraction of unique cases in a bootstrap sample as follows: n cases are randomly drawn with replacement for the bootstrap sample from the sample set of size n. Let

> u_k be the number of distinct cases in each set of size k drawn sequentially from a set of size n, $1 \leq k \leq n$,
>
> $\delta_k = 1$ if there is a new case in the set of size k not seen in the set of size $k-1, k \geq 1$, (occurs conditionally with probability $\frac{n-u_{k-1}}{n}$),
> $\delta_k = 0$, otherwise.

Define $u_1 = 1$, $u_0 = 0$. Therefore, $\delta_1 = 1$.
We have, since $u_k = u_{k-1} + \delta_k$ where

$$P(\delta_k = 1|u_{k-1}) = (n - u_{k-1})/n = 1 - P(\delta_k = 0|u_{k-1})$$

Therefore, $E[u_k|u_{k-1}] = u_{k-1} + \frac{n-u_{k-1}}{n}$, and

$$\begin{aligned} u_1 &= 1 \\ E[u_k] &= E_{u_{k-1}}[E[u_k|u_{k-1}]] \\ &= E[u_{k-1}] + \frac{n - E[u_{k-1}]}{n} \\ &= E[u_{k-1}](1 - \frac{1}{n}) + 1 \end{aligned}$$

Setting $\alpha = 1 - \frac{1}{n}$, we have

$$E[u_k] = \alpha E[u_{k-1}] + 1$$

Expanding this set of equations, we have

$$\begin{aligned} E[u_1] &= 1 \\ E[u_2] &= \alpha E[u_1] + 1 \\ E[u_3] &= \alpha E[u_2] + 1 \\ &= \alpha(\alpha + 1) + 1 \\ &= \alpha^2 + \alpha + 1 \\ E[u_n] &= \alpha^{n-1} + \alpha^{n-2} + \ldots + \alpha + 1 \end{aligned}$$

Since $\alpha < 1$, we have

$$\begin{aligned} E[u_n] &= \frac{1 - \alpha^n}{1 - \alpha} \\ &= n[1 - (1 - \frac{1}{n})^n] \qquad (4.11) \\ \lim_{n\to\infty} E[u_n] &= n(1 - e^{-1}) \\ &\approx 0.632 * n \qquad (4.12) \end{aligned}$$

In the bootstrap, we expect about .632 of all cases in the finite population to be present in the bootstrap sample drawn for each replication. The proportion is .651 for sets of size 10, .638 for sets of size 30, and .634 for sets of size 100. This overlap violates the bootstrap assumption that the relation of the bootstrap replication sample to the entire available sample is analogous to the relation of the entire available sample to the population. On this assumption the Monte Carlo simulation of the bootstrap is based. This overlap can cause the bootstrap evaluation of $E_* \text{ op}(\boldsymbol{x}^*, \hat{F})$ to be in error as the following calculations show.

Consider a situation with two classes and a classifier that classifies perfectly cases that it is trained on. Assume that it classifies randomly all cases that it has not previously seen (this is not likely for a real classifier because we expect some degree of continuity and not

random selection). [3]
For the training set, where we assume the classifier classifies perfectly,

$$E[Q(y,\eta)] = 0$$

For the test set, where we assume the classifier classifies randomly,

$$E[Q(y,\eta)] = 0.5$$

Therefore, the correct value of E op$(\boldsymbol{x}, F)$ is 0.5 - 0 = 0.5
The bootstrap equation 4.9 as given in [17, 18, 19] incorrectly gives this value as

$$(0.632 * 0) + (0.368 * 0.5) - 0 = .184$$

The bootstrap value of E_* op$(\boldsymbol{x}^*, \hat{F})$ is too low if the classifier has this behavior or a test error significantly greater than the training error.

The above example represents an extreme case, but even in practise, large differences can occur. One way for such differences to occur is when over-training has taken place. Over-training means to grant sensitivity to extraneous features of the particular items in the sample being classified. This can lead to an over-specification of the classification space into different regions [Figure 4.5].

Tables 7.30 and 7.31 give the values of $\hat{L}$ and $\hat{\kappa}_{b,w}$ for the trainable classifiers back-propagation, multiple linear regression, and C4 for the training set and the test set, respectively. We notice that there is a large difference in $\hat{L}$ and $\hat{\kappa}_{b,w}$ between the training set and the test set in the case of C4. In fact, the negative values of $\hat{\kappa}_{b,w}$ for C4 indicate that it is classifying worse than a classifier that guesses based on the a priori class distribution.

It is assumed in the bootstrap that when a bootstrap replication sample is drawn from the entire available sample, then the relation of the bootstrap replication sample to the entire available sample is analogous to the relation of the entire available sample to the population. It appears unreasonable that a real-world classifier has no *continuity properties,* that it has no concept of similarity. For most classifiers, as the size of the sample set increases, the estimates obtained from the training sample approach those of the test sample. However, there can be classifiers that can be based on one principle, for instance a discrimination tree of integer-valued attributes, while the data requires that the classifier does not discriminate between attributes but does a classification based on the sum of the attributes. We have encountered such a situation in our research. The C4 classifier, an implementation of a popular classifier, ID3 [61], uses a discrimination net in its working. Classifiers that can do a sum of attributes work well with our dataset. Even if such an incompatibility between a classifier and the data is not present in the application, it is difficult to judge the impact of the overlap between the bootstrap replication sample and the entire sample when computing the expected excess error. For this reason, a method different from the bootstrap needs to be used.

The procedure to compute the expected excess error of a classifier by the bootstrap [17, 18, 19, 34] depends on computing E_* op$(\boldsymbol{x}^*, \hat{F})$ by this method. This method does not take

[3] It is not possible to have even an artificial classifier that randomly classifies all cases that it has not previously seen, unless we assume that no unobserved cases are identical with training cases. This is because it has to classify a case encountered during testing in the same way as it did when the case was used during training.

into account the overlap between the bootstrap replication sample and the entire sample. The bootstrap procedure therefore cannot be used to compute the expected excess error of a classifier and hence cannot be used to estimate the true error rate of a classifier.

4.6 Estimating performance of a trainable classifier

The grouped jackknife (Section 4.4) can be used to estimate characteristics for classifiers that do not need to be trained. The literature on cross-validation (also known as leave-1-out) does not provide a method for estimating an arbitrary performance criterion for a classifier [79, 75, 34]. The bootstrap literature provides a procedure to estimate the expected excess error of a trainable classifier [17, 18, 19] but we showed above that this procedure has deficiencies. This procedure is essential for estimating an arbitrary characteristic of a trainable classifier using the bootstrap. We introduce three procedures to compute the mean and the standard error of an estimated performance criterion for a trainable classifier and show how to use them. The first procedure computes the mean and the standard error of the estimate when in each replication half the samples are randomly selected and used for training and the other half used for estimating. We call this procedure the repeated random halving (RRH) technique. The RRH technique is valid except that it drastically cuts the number of cases used for training and testing the classifiers. The RRH technique clearly is an almost unbiased estimator of the expected value of a criterion for a classifier trained on $\frac{n}{2}$ cases and evaluated on $\frac{n}{2}$ additional cases. We have not proved that the estimated standard error is unbiased but the close resemblance of the RRH technique to the bootstrap makes us confident that this is so. Fukunaga [27, 28] states that when both training and test sets are finite:

1. The bias of the classification error comes entirely from the finite training set.

2. The standard error comes predominantly from the finite test set.

He continues on to state that for this reason, in the absence of any a priori information, splitting the available sample set into two halves, one for training and one for testing, is an appropriate strategy. The RRH method is an extension of the holdout and rotation methods [79, 16]. The RRH method involves splitting the data into two mutually exclusive sets, training using one and testing using the other. It suffers from the disadvantage of the holdout and rotation methods that it is an inefficient use of the data – as one would like to use all the available data to train and test the classifier.

The second and third procedures try to reuse the cases for training and testing so that the number of cases available for training and testing is not reduced as drastically as in the RRH technique. The second and third procedures share a common stage where we first split the sample $\boldsymbol{\mathcal{X}}$ into g sets $\boldsymbol{\mathcal{X}}_i, i = 1, \ldots, g$ of size h each. Each set $\boldsymbol{\mathcal{X}}_i$ is used for evaluating the correctness of the classification of the classifier trained on the remaining $(g-1) \times h$ cases. This is the splitting stage in the technique.

Pooling all the evaluations (and the given assessments) together yields n pairs $< y_i, \hat{y}_i >$ where y_i is the correct classification and $\hat{y}_i$ is the attempted classification using a classifier trained on a set of $n - h$ cases not including $\boldsymbol{\mathcal{X}}_i$. If h is not too large, it is plausible that these n pairs have similar properties as they would if the $\hat{y}_i$ were derived from a sample of n cases independent of all the cases in the training sample.

One can estimate the expected value of a criterion and its standard error by treating the n pairs as independent (which they are not) and computing the estimate of the standard error of an estimator. If we use the grouped jackknife to estimate the characteristics of the pooled set we obtain the split grouped jackknife technique (SGJK). If we use the bootstrap to estimate the characteristics of the pooled set we obtain the split bootstrap (SB) technique.

The splitting procedure virtually doubles the size of the available sample set as almost all the samples can be used for training and testing.

4.6.1 Using repeated random halving to estimate the mean and the standard error (RRH).

The RRH procedure given below is an extension of the rotation method proposed in [16]. The rotation method is described in Section 4.3. From the given sample set we randomly draw *without replacement* half of the samples for a training set. The remaining half constitutes a test set. This procedure can be used repeatedly to estimate the value of a criterion for a particular sample set. This is a reasonable strategy in the absence of other information because the (unavailability of) training samples alone account for the degradation in a classifier's performance from a best possible classifier, while the (number of) test samples dominate the variance of the error estimate [28]. We call this method repeated random halving (RRH).

Given a set of observations $\boldsymbol{\mathcal{X}} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ and a criterion θ to be estimated, the RRH techique proceeds as follows.

1. Let B be the number of replications. A "sufficient" number for most applications should be $B = 50$. For $i = 1, \ldots, B$, the following steps are taken.
   (a) $n/2$ or $(n+1)/2$ observations are drawn randomly for the training set used in the $i'th$ replication, $\boldsymbol{\mathcal{X}}_{\alpha,i}$, depending on whether n is even or odd. The remaining samples constitute the estimation (or test) set for the $i'th$ replication, $\boldsymbol{\mathcal{X}}_{\beta,i}$.
   (b) The classifier is trained on the training set $\boldsymbol{\mathcal{X}}_{\alpha,i}$, the cases in the test set $\boldsymbol{\mathcal{X}}_{\beta,i}$ are evaluated using this classifier. The performance statistic is evaluated on the test set $\boldsymbol{\mathcal{X}}_{\beta,i}$ and is given by $\hat{\theta}_i = \hat{\theta}_{\boldsymbol{\mathcal{X}}_{\alpha,i}, \boldsymbol{\mathcal{X}}_{\beta,i}}$
2. The estimate of θ is
$$\hat{\theta} = \frac{1}{B} \sum_{i=1}^{B} \hat{\theta}_i \tag{4.13}$$
3. The estimate of standard error of $\hat{\theta}$ is
$$\hat{\sigma}_{\hat{\theta}} = \{ \frac{1}{B-1} \sum_{i=1}^{B} (\hat{\theta}_i - \hat{\theta})^2 \}^{1/2} \tag{4.14}$$

The RRH method estimates $\hat{\theta}$ and $SE(\theta)$ based on half the sample size. This estimate of $\hat{\theta}$ and $SE(\theta)$ will differ from the estimate obtained by using the entire sample. However, for a real-world application where we are given classifiers $C_1, C_2, \ldots, C_d$ and wish to order them based on $\hat{\theta}$, the RRH method can give good results. To obtain better results, a technique that reuses more of the entire sample for both training and testing is needed. Such techniques are presented in the next section but their correctness is unproven although a simulation indicating their plausibility is given.

4.6.2 Splitting and recombining the cases

The grouped jackknife as outlined in Section 4.4 cannot be used to estimate the performance of a trainable classifier. To use the SGJK or the SB technique it is necessary to follow the split recombination (SR) procedure outlined below.

Assume that we have a sample set S of n cases. Let each case $\boldsymbol{x}$ consist of a feature vector $\boldsymbol{t}$ and an evaluation (or classification) y. Therefore the set S can be written as

$$S = \{< \boldsymbol{t}_1, y_1 >, < \boldsymbol{t}_2, y_2 >, \ldots, < \boldsymbol{t}_n, y_n >\}$$

Divide the set of cases S into g sets of size h each, i.e., $S_1, S_2, \ldots, S_g$. (We assume that n is a multiple of h, otherwise some cases are discarded and not considered further.)
Number all the cases in set $S_k, 1 \leq k \leq g$. Let

$S(i) = i$'th case in set $S, i = 1, \ldots, n$,

$S_k(j) = j$'th case in set $S_k, j = 1, \ldots, h; k = 1, \ldots, g$

$$\gamma(k, j) = ((k-1) \times h) + j$$

$\gamma(k, j)$ is a numbering for the cases in S such that $S_k(j) = S(\gamma(k, j))$

$o = \lfloor \frac{i}{g} \rfloor$, where $\lfloor r \rfloor$ is the largest integer smaller than or equal to r. The i'th case of set S corresponds to the $(i - (o \times g))$'th case of set S_o.

With this numbering it is clear that S is divided into g sets $S_1, S_2, \ldots, S_g$ of size h and a one-to-one correspondence among the cases in S and $S_1, S_2, \ldots, S_g$ is established. For each set S_k define the set M_k. M_k is the set of all cases in S that are not present in S_k. $M_k = S - S_k$.
Let $M_k(j)$ denote the $j'th$ case in set M_k. We index the elements of M_k such that

$$\begin{aligned} M_k(j) &= S(j), \text{ if } (j \leq ((k-1) \times h)) \\ &= S(j+h), \text{ otherwise} \end{aligned}$$

After the above partitioning is done, for all $k = 1, \ldots, g$ train the given classifier on the cases in set M_k to obtain the classifier η_k. Evaluate the cases in set S_k using the classifier η_k. Let the evaluation of case $S_k(j)$ be $y'_k(j)$. Form the set S',

$$S' = \{< y_k(j), y'_k(j) > \forall k = 1, \ldots, g, j = 1, \ldots, h\}$$

Let, $S'(i)$ be the pair $< y(i), y'(i) >$ in S', where $y(i)$ is the given classification of case $S(i) =< \boldsymbol{t}(i), y(i) >$ and $y'(i)$ is the evaluation of $\boldsymbol{t}(i)$ by a classifier trained on the cases in set $M_{o'}$, where $o' = \lceil \frac{i}{h} \rceil$

4.6.3 Evaluating statistics over a training set using the split recombination procedure

If we cannot get more data, an unrealizable ideal is to train the classifier on one set $\boldsymbol{\mathcal{X}}_1 = \{\boldsymbol{x}_i =< \boldsymbol{t}_i, y_i >, i = 1, \ldots, n\}$ and evaluate the classifier on another set $\boldsymbol{\mathcal{X}}_2 = \{\boldsymbol{x}_i =<$

$\boldsymbol{t}_i, y_i >, i = 1, \ldots, n\}$. Let y'_i be the assessment of the case $x_i, i = 1, \ldots, n$. In this case, all the estimates depend on the pairs $< y_i, y'_i >, i = 1, \ldots, n$.

Consider the following conjecture which extends the concept of cross-validation (Section 4.3).

Split recombination (SR) Conjecture: *The statistical properties of S' as obtained using the split-recombination procedure are approximately the same as the properties of the set of assessment pairs $< y_i, y'_i >$ obtained by training the classifier on $n - h$ cases drawn independently of S, and tested on $S = \{\boldsymbol{x}_i =< \boldsymbol{t}_i, y_i >, i = 1, \ldots, n\}$.*

It is plausible that when n is large and h is a small fraction of n that this conjecture is true. If the SR conjecture is correct then we can estimate the values of criteria for a trainable classifier by estimating their values from S'. The validity of this procedure of splitting and recombining the cases rests on the assumption that the SR conjecture is correct.

To compute an estimator such as $\hat{\kappa}_{b,w}$ or $\hat{L}$ we compute it over the set S' using any available technique. $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ can be estimated using formulae based on the multinomial distribution since the estimators we are using are functions of the proportions in each cell. In fact, we can summarize the information in S' by a contingency table. The grouped jackknife can approximate the values produced by the multinomial results and saves computation as compared with the bootstrap, although this saving is not significant today with the easy availability of high-speed digital computers. The values estimated using the bootstrap will approach the value computed using the formulae for multinomials as the number of bootstrap replications increases. $SE(\kappa_{b,w})$ has to be estimated from S' using either the grouped jackknife or the bootstrap because a formula for $SE(\kappa_{b,w})$ is not available.

$\hat{\kappa}$ and $\hat{L}$ depend on S.

4.6.4 The split multinomial method (SM)

When we wish to evaluate a criterion for a trainable classifier over a set S and use results based on multinomial theory [Equations 2.2, 2.4] to evaluate a criterion over S' as computed above, we call the procedure the split multinomial (SM) technique. The split multinomial technique is preferable to the split grouped jackknife (SGJK) technique (Section 4.6.5) and the split bootstrap (SB) technique (Section 4.6.6) if the multinomial results are available. The value and standard error of a criterion estimated using the SB technique approach those of the multinomial results over S' as the number of bootstrap replications increases ($B \rightarrow \infty$). If the criterion is linear (such as L but not $\kappa_{b,w}$) then the results of the jackknife (which is the grouped jackknife with $h = 1$) are essentially identical to multinomial results. The results of the SM method should be similar to the results of the multi-diff method (Section 4.7.1) if the SR conjecture is correct.

4.6.5 The split grouped jackknife method (SGJK)

When we wish to evaluate a criterion for a trainable classifier over a set S and when we use the grouped jackknife to evaluate a criterion over S' as computed above, we call the procedure the split grouped jackknife (SGJK) technique. After using the split recombination procedure, S' has to be divided into sets for the grouped jackknife to be applied. S' can be divided into a different number of sets (than S was divided into) using a leave-out factor of h_s (instead of h as in S). If $h = h_s$ we can simply specify h. The order of cases in S' can

be randomized (so that the order is different from the cases in S). Randomization of the order of cases in S' makes little difference to the estimates (Chapter 7). The results of the SGJK method should be similar to the results of the gjk-diff method (Section 4.7.1) if the SR conjecture is correct.

4.6.6 The split bootstrap method (SB)

When we wish to evaluate a criterion for a trainable classifier over a set S and when we use the bootstrap to evaluate a criterion over S' as in Section 4.6.2, we call the procedure the split bootstrap (SB) technique. The results of the SB method should be similar to the results of the bst-diff method (Section 4.7.1) if the SR conjecture is correct.

The name *split bootstrap* is derived from the fact that the split recombination procedure is applied followed by estimation using the bootstrap technique. The entire procedure does not constitute the bootstrap technique because the principle underlying the bootstrap technique is that "the relationship between a random sample drawn from the given sample is the same as the relationship between the sample (drawn randomly from the population) and the population, and we can estimate properties from the sample and extrapolate to the population." This principle has not been used for the entire SB procedure.

4.7 Simulation to show the plausibility of the SM, SGJK, and SB methods

The following simulation is done to determine the plausibility of the SM, SGJK, and SB methods. We take a very simple model consisting of a mixture of two univariate normal random variables with the same variance. Since the SR conjecture relates the statistical properties of a classifier trained and tested on S with those obtained by training the classifier on $n-h$ cases drawn independently of S, and tested on $S = \{x_i =< t_i, y_i >, i = 1, \ldots, n\}$, we need to compare with a different dataset of $n-h$ cases drawn from the same population as S. This is impossible to do with a real-world dataset but it can be done with a simulated model. We introduce the *diff methods* and *diff datasets* as below for comparison purposes.

4.7.1 The *diff dataset* and the *diff methods*

The *diff dataset*, S_{diff}, is a random sample of $n-h$ cases drawn from the same population as S independently of S. Once the diff dataset is defined we can define the *diff methods* as those methods trained on S_{diff} and tested on S. We define the multi-diff, gjk-diff, and bst-diff methods below. These methods can collectively be termed as the diff methods. The SM, SGJK, and SB methods can collectively be termed as the *split methods.* If the SR conjecture is correct, each diff method should have comparable results to the corresponding split method that is based on the SR conjecture. Simulation using the S_{diff} datasets and the diff-methods provide us with results that we know to be correct for the S' dataset. These results can be compared with those from simulation using the SM, SGJK, and SB methods to check the plausibility of the SR conjecture. All diff-methods consists of training a method over the diff dataset and applying it to S. They differ in the technique used to estimate criteria and their standard errors. To recall, S' is the set of pairs $< y, y' > \forall \boldsymbol{x} = (\boldsymbol{t}, y) \in S$ where y' is the classification of $\boldsymbol{t}$ by a method using the SR conjecture. S' can be evaluated

using the multinomial results for the SM method, or the grouped jackknife for the SGJK method, or the bootstrap for the SB method. Similarly, S'_{diff} can be defined as the set of pairs $< y, y' > \forall \boldsymbol{x} = (\boldsymbol{t}, y) \in S$ where y' is the classification of $\boldsymbol{t}$ by a method trained on the diff dataset.

For the multi-diff method $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ are evaluated using the results from the multinomial distribution over S'_{diff}. The results of the multi-diff method can be compared with those of the SM method. $SE(\kappa_{b,w})$ cannot be computed using the multi-diff or the SM methods because the results from the multinomial distribution for $SE(\kappa_{b,w})$ are not known. For the gjk-diff method $\hat{L}$, $SE(L)$, $\hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$ are evaluated using the grouped jackknife over S'_{diff}. The results of the gjk-diff method can be compared with those of the SGJK method. For the bst-diff method $\hat{L}$, $SE(L)$, $\hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$ are evaluated using the bootstrap over S'_{diff}. The results of the bst-diff method can be compared with those of the SB method.

4.7.2 Procedure used in the simulation

Independently obtain S, a sample set of size n, and the diff dataset, S_{diff}, a sample set of size $n - h$, as follows

1.
$$y_i = Bernoulli(p)$$
where $Bernoulli(p)$ is a Bernoulli Random Variable with parameter p

2. **if** $(y_i = 0)$ **then**
$$x_i = N(\mu_0, 1)$$
else
$$x_i = N(\mu_1, 1)$$
endif

$N(\mu, 1)$ is a Normal Random Variable with mean μ, and standard deviation 1. Define class $\omega_j = \{(x, y)|y = j\}, j = 0, 1$. For all the experiments $p = 0.6$ and $\mu_0 = 0$. $\mu_1 = 1, 2.5,$ or 4 is chosen for the experiments below. y is the variable to be predicted and x is the independent variable. Form the sample sets $S = \{< x_i, y_i >, i = 1, \ldots, n\}$. and $S_{diff} = \{< x_i, y_i >, i = 1, \ldots, n - h\}$. The loss function used is

$$L_{0|0} = L_{1|1} = 0$$

$$L_{0|1} = L_{1|0} = 1$$

Train the classifier by computing the sample means $\overline{X}_0$ and $\overline{X}_1$ as below [Equations 4.15, 4.16]. If $|X_i - \overline{X}_0| < |X_i - \overline{X}_1|$ classify the case as belonging to class ω_0, otherwise as belonging to class ω_1. On the basis of feature value x_i the classifier will thus classify case i to the class ω_0 or ω_1 whose estimated mean is nearest to x_i.
For $r = 1, \ldots, B$ replications, do as follows:

1. Obtain independent sample sets S_r of size n and $S_{diff,r}$ of size $n - h$. To recall, $S_{diff,r}$ is the set of pairs $< y, y' > \forall \boldsymbol{x} = (\boldsymbol{t}, y) \in S_r$ where y' is the classification of $\boldsymbol{t}$ by any method.

2. Compute the sample means

$$\overline{X}_{0,r} = \frac{\sum_{<x_i,y_i>\in S_{diff,r}}(1-y_i)x_i}{\sum_{<x_i,y_i>\in S_{diff,r}}(1-y_i)} \tag{4.15}$$

$$\overline{X}_{1,r} = \frac{\sum_{<x_i,y_i>\in S_{diff,r}} y_i x_i}{\sum_{<x_i,y_i>\in S_{diff,r}} y_i} \tag{4.16}$$

from the training set S_{diff}. Note that each summation consists of $n-h$ terms.

3. For each case $< x_i, y_i > \in S_r$, define

$$\begin{aligned} \hat{y}_i &= 0 \text{ if } |x_i - \overline{X}_{0,r}| < |x_i - \overline{X}_{1,r}| \\ &= 1, \text{ otherwise.} \end{aligned}$$

That is, we assign x_i to the population with nearer $\overline{X}_r$.

For notational convenience we will denote the set of cases with the evaluated assessments (of the sample set S_r) as $S'_{diff,r}$.

4. Compute $\hat{L}$, $\hat{\kappa}_{b,w}$, $SE(L)$, and $SE(\kappa_{b,w})$ over the cases in S_r by using the classifier trained over $S_{diff,r}$ using the multinomial results, the grouped jackknife technique, and the bootstrap technique. These values can be termed as $\hat{L}(M,r)$, $\hat{\kappa}_{b,w}(M,r)$, $SE(L,M,r)$, and $SE(\kappa_{b,w},M,r)$ where M is the multi-diff, gjk-diff, or the bst-diff technique respectively. The multinomial results are not known for computing $SE(\kappa_{b,w})$ therefore $SE(\kappa_{b,w})$ can be computed using the gjk-diff and the bst-diff techniques only. These estimates will be compared with the estimates obtained over S_r alone using the SM, SGJK, and the SB techniques respectively. To apply the grouped jackknife and the SGJK technique, the cases in S'_r can be split into a different number of sets than the cases in S_r.

5. Compute the values $\hat{L}(M,r)$, $SE(L,M,r)$, $\hat{\kappa}_{b,w}(M,r)$, and $SE(\kappa_{b,w},M,r)$ over the set S'_r using the methods M = SM, M = SGJK, and M = SB . The multinomial results are not known for computing $SE(\kappa_{b,w})$ therefore $SE(\kappa_{b,w})$ can be computed using the SGJK and the SB techniques only.

Compare the statistics (using stem and leaf diagrams, scatterplots, t-values, and coverage of confidence intervals) obtained by the SM, SGJK, and the SB methods with the statistics obtained using the diff-methods multi-diff, gjk-diff, and bst-diff respectively. The values estimated for statistics using the diff-methods are the values that would be obtained when a classifier was trained on $n-h$ independent samples and tested on the given sample of size n. These are the same values that we would hope that the SM, SGJK and SB methods deliver. Let

$$\hat{\theta}_{diff} = \frac{1}{R}\sum_{r=1}^{R}\hat{\theta}(M-diff,r)$$

$$SE(\theta_{diff}) = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}(\hat{\theta}(M-diff,r)-\hat{\theta}_{diff})^2}$$

$\hat{\theta}_{diff}$ and $SE(\theta_{diff})$ are known to be correct estimators of θ and $\sigma_{\hat{\theta}_{M-diff}}$ when the classifier is trained with $n-h$ cases drawn from a population and tested on n cases that are independently drawn from the same population. The results of comparing $SE(\theta, M)$ obtained during the R replications with the correct values for θ follow.

The combinations of $R, n, h = (n/g), p$ chosen are shown in Table 4.1. The number of bootstrap replications, $B = 200$ for determining the coverage of confidence intervals although for the other tests $B = 1000$ may also be used.

Simulation No.	B	R	n	$h = (n/g)$	p
1	200	200	50	1	0.6
2	200	200	50	5	0.6
3	200	200	630	1	0.6
4	200	200	630	21	0.6

Table 4.1: Table showing simulation configurations for testing the SGJK and the SB methods.

4.7.3 Determining the number of times the true value falls in the estimated confidence interval about the estimated value

One way to determine whether the estimated standard error is accurate is to determine whether the probability that the $100(1-\alpha)\%$ confidence limits contain the true value is $1-\alpha$. We set $p = 0.6$. If X_0 and X_1 are the means of the classes ω_0 and ω_1, and $N(0,1)$ and $N(0,\mu)$, are the populations from which the data is drawn then it is easy to show that:

$$P(\hat{y}=0|y=0) = P(x < (\overline{X}_0 + \overline{X}_1)/2),\ x \sim N(0,1)$$

$$P(\hat{y}=0|y=1) = P(x < (\overline{X}_0 + \overline{X}_1)/2),\ x \sim N(\mu,1)$$

$$P(\hat{y}=1|y=0) = P(x > (\overline{X}_0 + \overline{X}_1)/2),\ x \sim N(0,1)$$

$$P(\hat{y}=1|y=1) = P(x > (\overline{X}_0 + \overline{X}_1)/2),\ x \sim N(\mu,1)$$

These values can be easily computed from known algorithms. $L, \sigma(L), \kappa_{b,w}$ can be computed from the above values as follows. Define,

$$L_{0|0} = L_{1|1} = 0$$

$$L_{0|1} = L_{1|0} = 1$$

Therefore,

$$L = L_1 + L_2 \tag{4.17}$$

where

$$L_1 = p \times P(\hat{y}=0|y=1)$$

$$L_2 = (1-p)P(\hat{y}=1|y=0)$$

$$SE(L) = \sqrt{\frac{1}{n}(L_1 + L_2 - L_1 L_2)} \tag{4.18}$$

where n is the number of cases. Define

$$W = 1 - L$$

$$W_0 = 1 - p$$

$$W_1 = p$$

$$W_b = \max(W_0, W_1)$$

Then

$$\kappa = \frac{W - W_b}{1 - W_b} \tag{4.19}$$

The following steps are done in the experiment:

1. For sample sets of size $n \in \{50, 630\}$ do as follows for $r = 1, \ldots, 200$ replications:
   (a) Compute $\overline{X}_0$ and $\overline{X}_1$ from the sample set. Determine the true values of L and $\kappa_{b,w}$ for the sample using Equations 4.17 and 4.19 respectively.
   (b) Determine $\hat{L}_r$, $\hat{\kappa}_{b,w,r}$, and $SE(L_r)$ for an independent sample of size n using the following techniques: multinomial results (multi-diff and SM), the bootstrap (bst-diff and SB), and the grouped jackknife (gjk-diff and SGJK). $SE(\kappa_{b,w,r})$ is computed using the the bootstrap (bst-diff and SB) and the grouped jackknife (gjk-diff and SGJK). For $n = 50$, $h = 1$ and $h = 5$ is used. For $n = 630$, $h = 1$ and $h = 21$ is used. When $h = 1$ is used, the grouped jackknife reduces to the jackknife.
   (c) For each method determine whether θ is in the 90% and 95% confidence interval centered at $\hat{\theta}$. In our experiments, the criterion θ is either L or $\kappa_{b,w}$. A value θ is in the k% C.I. if
   $$|(\hat{\theta} - \theta)|/SE(\theta) \leq \tau$$
   For $k = 90, \tau = 1.645$. For $k = 95, \tau = 1.96$.
2. Determine the fraction of times L_r $(\kappa_{b,w,r})$ is in the 90% and 95% confidence interval of L_r $(\kappa_{b,w,r})$ about $\hat{L}_r$ $(\hat{\kappa}_{b,w,r})$. Do not consider those cases where the classifier classifies all the test cases perfectly when determining these fractions.

The results of this experiment with $\mu = 1$, $n = 630$, and $h = 21$ are given in Table 4.2. The results of this experiment with $\mu = 1$, $n = 50$, and $h = 1$ are given in Table 4.3. The results show that none of these coverages [Tables 4.2, 4.3] are significantly different from nominal, since $2 \times$ SE(binomial) = .03 for $p = .95$ and $2 \times$ SE(binomial) = .04 for $p = .90$. The multi-diff, bst-diff, and gjk-diff techniques also give the same order of coverage as the SM, SB and SGJK techniques. Therefore the SR conjecture is plausible.

The results of this experiment with $\mu = 2.5$ and $n = 630$ or $n = 50$ are given in Tables 4.4 and 4.5 respectively. The results show that these coverages [Tables 4.2, 4.3] are slightly different from nominal, since $2 \times$ SE(binomial) = .03 for $p = .95$ and $2 \times$ SE(binomial) = .04 for $p = .90$ but the multi-diff, bst-diff, and gjk-diff techniques also give the same order of coverage as the SM, SB and SGJK techniques. Therefore the SR conjecture is plausible.

Technique	$L(90\%)$	$L(95\%)$	$\kappa_{b,w}(90\%)$	$\kappa_{b,w}(95\%)$
multi-diff	.885	.945	X	X
SM	.890	.960	X	X
bst-diff	.885	.945	.850	.935
SB	.890	.960	.855	.935
gjk-diff	.885	.940	.845	.925
SGJK	.865	.945	.850	.925

Table 4.2: Fraction of times the true value lies within the within the 90% and 95% C.I estimated by different methods. $\mu_1 = 1, n = 630, h = 1$

Technique	$L(90\%)$	$L(95\%)$	$\kappa_{b,w}(90\%)$	$\kappa_{b,w}(95\%)$
multi-diff	0.875	0.925	X	X
SM	.88	.915	X	X
bst-diff	.88	.930	.915	.965
SB	.87	.915	.915	.97
bst-diff	.875	.930	.915	.955
SB	.885	.920	.890	.945

Table 4.3: Fraction of times the true value lies within the 90% and 95% C.I estimated by different methods. $\mu_1 = 1, n = 50, h = 1$

Technique	all-same	$L(90\%)$	$L(95\%)$	$\kappa_{b,w}(90\%)$	$\kappa_{b,w}(95\%)$
multi-diff	0	.88	.945	X	X
SM	0	.89	.935	X	X
bst-diff	0	.855	.95	.855	.915
SB	0	.89	.925	.88	.925
gjk-diff	0	.88	.935	.865	.915
SGJK	0	.88	.94	.875	.925

Table 4.4: Fraction of times the true value lies within the within the 90% and 95% C.I estimated by different methods. $\mu_1 = 2.5, n = 630, h = 21$

Technique	all-same	$L(90\%)$	$L(95\%)$	$\kappa_{b,w}(90\%)$	$\kappa_{b,w}(95\%)$
multi-diff	2	.878	.898	X	X
SM	0	.86	.895	X	X
bst-diff	2	.878	.904	.883	.919
SB	0	.84	.900	.86	.900
gjk-diff	2	.883	.904	.858	.883
SGJK	0	.86	.895	.855	.885

Table 4.5: Fraction of times the true value lies within the within the 90% and 95% C.I estimated by different methods. $\mu_1 = 2.5, n = 50, h = 1$

Technique	all-same	$L(90\%)$	$L(95\%)$	$\kappa_{b,w}(90\%)$	$\kappa_{b,w}(95\%)$
multi-diff	0	.90	.95	X	X
SM	0	.90	.945	X	X
bst-diff	0	.90	.94	.90	.95
SB	0	.905	.95	.910	.94
gjk-diff	0	.91	.935	.90	.925
SGJK	0	.895	.935	.910	.93

Table 4.6: Fraction of times the true value lies within the within the 90% and 95% C.I estimated by different methods. $\mu_1 = 4.0, n = 630, h = 21$

The results of this experiment with $\mu = 4.0$ and $n = 630$ or $n = 50$ are given in Tables 4.6 and 4.7 respectively. In Table 4.6 the normal approximation is invalid because there are many replications in which the classifier classifies all cases perfectly. The results show the SM technique gives larger coverage than indicated by the normal approximation of $\hat{L}$. The SGJK technique and the SB technique give larger coverage than indicated by the normal approximation of $\hat{L}$ and $\hat{\kappa}_{b,w}$. The bst-diff and gjk-diff techniques also give larger coverage that are similar to the SB and SGJK techniques. Therefore the SR conjecture is plausible.

4.7.4 p-values for the simulation results

We compute the p values and the Bonferroni[4] p-values using a multiplier of 3 ($3 \times 2/2 = 3$) for $SE(\kappa_{b,w})$ estimated by the SGJK and the SB techniques. Since $\hat{L}$, $\hat{\kappa}_{b,w}$, and $SE(L)$ are estimated using multinomial results we do not compute p-values for them. The goal of the p values is to determine whether the gjk-diff technique, the SGJK technique, the bst-diff technique, and the SB technique are estimating the same quantity. We do this by showing that the bst-diff and SB techniques, the gjk-diff and bst-diff techniques, and the

[4] The Bonferroni p-value is computed by multiplying the p-value by the number of hypotheses being tested to obtain an overall p-value. The Bonferroni technique is one way to account for the effect of multiple comparisons [54]

Technique	all-same	$L(90\%)$	$L(95\%)$	$\kappa_{b,w}(90\%)$	$\kappa_{b,w}(95\%)$
multi-diff	64	.992	.992	X	X
SM	62	.992	1.0	X	X
bst-diff	64	.992	1.0	1.0	1.0
SB	62	.992	1.0	.992	1.0
gjk-diff	64	.992	.992	.992	1.0
SGJK	62	.992	1.0	.992	1.0

Table 4.7: Fraction of times the true value lies within the within the 90% and 95% C.I estimated by different methods. $\mu_1 = 4.0, n = 50, h = 1$

Quantity studied	t-value	p-value	Bonferroni p-value
SE($\kappa_{b,w}$, bst-diff) - SE($\kappa_{b,w}$, SB)	-0.095394	0.9241	> 0.5
SE($\kappa_{b,w}$, gjk-diff) - SE($\kappa_{b,w}$, SGJK)	-0.190730	0.8489	> 0.5

Table 4.8: Bonferroni values for differences of $SE(\kappa_{b,w})$ estimated using different methods. $R = 200, \mu_1 = 1, n = 50, h = 1$
199 degrees of freedom.

gjk-diff and SGJK estimate the same quantities for $SE(\kappa_{b,w})$, respectively. For any two methods M_1 and M_2 we can test the hypothesis that the estimated standard errors are the same by computing $\hat{\sigma}_1 = \hat{\sigma}_{\theta(M_1)}$, $\hat{\sigma}_2 = \hat{\sigma}\theta(M_2)$, $\hat{\sigma}_{diff} = \hat{\sigma}_1 \widehat{-} \hat{\sigma}_2$, and $SE(\hat{\sigma}_1 - \hat{\sigma}_2)$. Then $z = \hat{\sigma}_{diff}/SE(\hat{\sigma}_1 - \hat{\sigma}_2)$. By comparing the z-score with the standard normal distribution we can determine whether the methods estimate different values for $SE(\theta)$.

The results are shown in Tables 4.8 to 4.11. The SB and bst-diff techniques, and the SGJK and gjk-diff techniques yield estimated standard errors that are mathematically the same except for a constant multiplier near 1[5]. For this reason the t-values are not shown and a Bonferroni multiplier of two is chosen.

The t-tests indicate that the SR conjecture is plausible, since the t-tests do not detect any significant difference between the values computed by the diff methods trained on the DIFF dataset and the SM, SB, and SGJK methods based on the SR conjecture.

[5] The bst-diff technique approaches the multinomial results as $B \to \infty$. In the multinomial results a divisor of n is used, while in the grouped jackknife a divisor of $n-1$ is used. Therefore the standard errors obtained by the grouped jackknife are consistently higher than those obtained by the multinomial results (and the bootstrap) by a factor of $\sqrt{n/(n-1)}$.

Quantity studied	t-value	p-value	Bonferroni p-value
SE($\kappa_{b,w}$, bst-diff) - SE($\kappa_{b,w}$, SB)	1.436152	0.1525	0.4575
SE($\kappa_{b,w}$, gjk-diff) - SE($\kappa_{b,w}$, SGJK)	1.063519	0.2888	> 0.5

Table 4.9: Bonferroni values for differences of $SE(\kappa_{b,w})$ estimated using different methods.
$R = 200, \mu_1 = 4, n = 50, h = 5$
199 degrees of freedom.

Quantity studied	t-value	p-value	Bonferroni p-value
SE($\kappa_{b,w}$, bst-diff) - SE($\kappa_{b,w}$, SB)	0.422959	0.6728	> 0.5
SE($\kappa_{b,w}$, gjk-diff) - SE($\kappa_{b,w}$, bst-diff)	-1.687765	0.09302	0.2791
SE($\kappa_{b,w}$, gjk-diff) - SE($\kappa_{b,w}$, SGJK)	-0.079394	0.9368	> 0.5

Table 4.10: Bonferroni values for differences of $SE(\kappa_{b,w})$ estimated using different methods.
$R = 200, n = 630, \mu_1 = 1, h = 1$
199 degrees of freedom.

Quantity studied	t-value	p-value	Bonferroni p-value
SE($\kappa_{b,w}$, bst-diff) - SE($\kappa_{b,w}$, SB)	.090776	0.9278	> 0.5
SE($\kappa_{b,w}$, gjk-diff) - SE($\kappa_{b,w}$, bst-diff)	-1.168701	0.2439	> 0.5
SE($\kappa_{b,w}$, gjk-diff) - SE($\kappa_{b,w}$, SGJK)	.297007	0.7668	> 0.5

Table 4.11: Bonferroni values for differences of $SE(\kappa_{b,w})$ estimated using different methods.
$R = 200, \mu_1 = 1, n = 630, h = 21, B = 200.$
199 degrees of freedom.

Figure 4.5: Effect of over-training on classification space.

	Probability
$P(X < 0.8)$	1.9e-5
$P(X < 0.9)$	.0229
$P(X > 1.1)$	.0229
$P(X > 1.2)$	4.7e-5
$P(X > 1.3)$	4.0e-9
$P(X > 1.5)$	~ 0
$P(X > 2.0)$	~ 0

Table 4.12: Probabilities for $X = \chi_F/\sqrt{F}$, $F = 199$.

4.7.5 Stem and leaf diagrams for the simulation.

The tables in the previous section are one way to summarize the results of the simulation. Another way is to obtain stem and leaf plots of $\hat{L}, SE(L), \hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$. Since 200 replications are used in the simulation, 199 degrees of freedom are present. To the extent that the estimated standard errors behave like estimated standard deviations based on a normal random sample of size 200 using $\sigma\sqrt{\chi^2(199)/199}$, the interval $(.9018\sigma, 1.098\sigma)$ should contain approximately 95% of the estimated standard errors. For the χ distribution the lower and upper quartiles in the stem and leaf plots correspond to $(.965\sigma, 1.032\sigma)$. The median of the χ distribution for 199 d.f. corresponds to $.99832\sigma$. In the stem and leaf plots of $SE(L, M, r)/\sigma_{SE(L_{diff})}$ and $SE(\kappa_{b,w}, M, r)/SE(\kappa_{b,w,diff})$ it is helpful to know that for the χ distribution, .82381 of the data points lie between the points $SE/\sigma = 0.8$ and $SE/\sigma = 1.1$. Selected probabilities for $X = \chi/\sqrt{199}$ are in Table 4.12.

The stem and leaf plots can be compared with the shape of the χ distribution. We see in the stem and leaf plots that the median standard errors estimated are within the above intervals. This shows that the methods are giving the correct values of $\hat{L}, SE(L), \hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$. The shapes of the stem and leaf diagrams for $\hat{L}$ and $\hat{\kappa}_{b,w}$ are like the normal distribution. The variability of $SE(L)$ is similar to that of the χ distribution. The variability of $SE(\kappa_{b,w})$ is much greater than that of the χ distribution both for the diff methods and for the split methods. Since the shapes of the distribution for the diff methods and the split methods is the same, the fact that $SE(\kappa_{b,w})$ is more variable than expected from the χ distribution does not speak against the SR conjecture.

L has a lower limit of 0.0 and an upper limit of 1.0. However, since our classifiers are designed to give a low value of L, L appears positively skewed. $\kappa_{b,w}$ has an upper limit of 1.0 and no lower limit. Therefore plots of $\hat{\kappa}_{b,w}$ appear negatively skewed. Besides the median value falling within the limits of the χ distribution, we note that the standard errors of $\hat{\kappa}_{b,w}$ are extremely variable. We divide the quantity being studied obtained during each replication by the quantity obtained across replications and plot this value in the stem and leaf diagrams. This lets us know how variable the quantity under study is during replications.

Looking at the stem and leaf plots [Figures 4.6 to 4.29] in greater detail we see that:

1. Stem and leaf plots of $\hat{L}(M, r)/\hat{L}_{diff}$ where $M = SM$ or $M = multi - diff$ are bell-shaped as in the normal distribution for Figures 4.6, 4.7, 4.8, and 4.9. For these figures

the normal approximation is probably appropriate. The plots for $\hat{L}(M,r)/\hat{L}_{diff}$ where $M = SM$ or $M = multi - diff$ are not bell-shaped as in the normal distribution for Figures 4.10 and 4.11. This is probably because the normal approximation to the binomial breaks down as $\mu_1 = 4$ resulting in almost perfect classification. Plots of $\hat{L}$ whether obtained by the SM method or the multi-diff method for $h = 1$ or $h = 5$ are similar in shape for all the above plots [Figures 4.6, 4.7, 4.8, 4.9, 4.10, and 4.11]. The median value of $\hat{L}(M,r)/\hat{L}_{diff}$ where $M = SM$ or $M = multi - diff$ in the plots where the normal approximation is valid [Figures 4.6, 4.7, 4.8, and 4.9] is close to 1. This indicates that the SM and multi-diff methods are estimating the same value of $\hat{L}$ in each replication that is being estimated over the DIFF datasets across the 200 replications. The plots of the SM method and the multi-diff methods are similar in shape and have similar medians. Therefore the SR conjecture is plausible.

2. Plots of $SE(L,M,r)/SE(L_{diff})$ where $M = SM$ or $M = multi - diff$ obtained are not like the χ distribution in shape [Figures 4.12, 4.13, 4.14, 4.15, 4.16, and 4.17]. The variability is according to χ distribution but the skewness is in the opposite direction for the figures where the normal approximation holds ($\mu_1 \neq 4$) [Figures 4.12, 4.13, 4.14, and 4.15]. As noted before, this is because L has a minimum value of 0, a maximum value of 1, and our classifiers are attempting to have L values close to 0. The median value of $SE(L,M,r)/SE(L_{diff})$ where $M = SM$ or $M = multi - diff$ in these plots is close to 1 and the plots are similar in shape. This indicates that the SM and multi-diff methods are estimating the same value of $SE(L)$ in each replication that is being estimated over the DIFF datasets across the 200 replications. The plots of the SM method and the multi-diff methods are similar in shape and have similar medians. Therefore the SR conjecture is plausible.

3. Many of the plots show *clumping* of values along the stems. We can examine this phenomenon further. Figures 4.10 and 4.11 of $\hat{L}$ by the multi-diff and SM techniques and $SE(L)$ in Figures 4.16 and 4.17 show considerable clumping of values along the stems. Looking at Figures 4.32 and 4.33 which are plots of $\hat{L}(SM\ ,r)$ vs. $\hat{L}(multi-diff\ ,r)$, we notice that there are only six [five] possible values for $\hat{L}(SM\ ,r)$ [$\hat{L}(multi-diff\ ,r)$.] This is reasonable, because $\hat{L}(SM\ ,r)$ and $\hat{L}(multi-diff\ ,r)$ depend only on the number of incorrect classifications (since $L_{01} = L_{10} = 1$). In some of the other figures (e.g., Figure 4.12) this clumping may be observed to a lesser extent. This is because $\hat{L}$, $SE(L)$, $\hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$ depend only on the number of incorrect classifications and the number of incorrect classifications is finite. Even after clumping, the median value of $SE(L)/SE(L_{diff})$ in these plots are close to 1. This indicates that the SM and multi-diff methods are estimating the same value of $SE(L)$ in each replication that is being estimated over the DIFF datasets across the 200 replications. The plots of the SM method and the multi-diff methods are similar in shape and have similar medians. Therefore the SR conjecture is plausible.

4. Plots of $\hat{\kappa}_{b,w}(M,r)/\hat{\kappa}_{b,w,diff}$ where $M = SM$ or $M = multi-diff$ [Figures 4.18, 4.19, 4.20] are similar in shape. They are negatively skewed and the median values for these plots are less than the value computed by the multi-diff method for the experiment across all replications, whether the median is computed over the SPLIT dataset or the DIFF dataset. The plots of the SM method and the multi-diff methods are similar in

shape and have similar medians. Therefore the SR conjecture is plausible.

5. Plots of $SE(\kappa_{b,w}, M, r)/SE(\kappa_{b,w,diff})$ where $M = SM$ or $M = multi - diff$ are given in Figures 4.22 to 4.29. The plots are positively skewed and there are values of $SE(\kappa_{b,w})$ reported that are as high as $2 \times SE(\kappa_{b,w,diff})$. For a χ plot with 199 d.f. the probability of such a high standard error is close to 0 [Table 4.12]. The median values for these plots of $SE(\kappa_{b,w})/SE(\kappa_{b,w,diff})$ is close to 1. Since the shapes of the DIFF stem and leaf plots and the SPLIT stem and leaf plots is the same, the SR conjecture is plausible.

From the above observations/plots it is clear that the shape and median of the distributions obtained by the SM method while computing $\hat{L}, SE(L)$, and $\hat{\kappa}_{b,w}$ over the SPLIT dataset is similar to that obtainable by using the multinomial results over the diff dataset. Therefore the SR conjecture as plausible. SGJK and SB can be used to compute $SE(\kappa_{b,w})$ but with a high degree of variability over the SPLIT dataset, and their shapes will be similar to that obtained over the DIFF dataset. The high degree of variability is partly because it requires many samples to estimate the standard error. Estimating $SE(\kappa_{b,w})$ across 200 replications makes available $200 \times n$ data points for estimating standard error while the various replications of the diff-methods multi-diff, gjk-diff, and bst-diff and the SM, SB, and SGJK methods being tested had access to only n data points.

```
n = 200  Min=0.500313  QL=0.875547  M=1.00063  QU=1.1257  Max=1.62602
    5     5|00666
   16     6|22222888888
   29     7|5555555555555
   62     8|11111111111117777777777777777777
   91     9|33333333333333333333333333333
 ( 49)   10|00000000000000000000000000666666666666666666666666
   60    11|22222222222222222888888888888888
   27    12|555555555
   18    13|111111777777
    6    14|33
    4    15|000
    1    16|2

          1|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.6: Stem and leaf plot of $\hat{L}(multi-diff\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff} = 0.3198$. n=200, N=50, h=1.

```
n = 200  Min=0.500313  QL=0.875547  M=1.00063  QU=1.1257  Max=1.68856
    5     5|00666
   15     6|2288888888
   29     7|55555555555555
   70     8|11111111111111111117777777777777777777777
   97     9|33333333333333333333333333
 ( 48)   10|00000000000000000000000000000006666666666666666
   55    11|22222222222228888888888888888
   26    12|55555555
   18    13|11111177
   10    14|3333
    6    15|00066
    1    16|8

          1|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.7: Stem and leaf plot of $\hat{L}(SM\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff} = 0.3198$. n=200, N=50, h=1.

```
n = 200  Min=0.512492  QL=0.832799  M=0.960922  QU=1.15311  Max=1.47341
    5     5|11777
    9     6|4444
   38     7|00000000000066666666666666666
   77     8|333333333333333333999999999999999999999
 ( 26)    9|66666666666666666666666666
   97    10|22222222222222222222888888888888888888888888
   54    11|555555555555555555
   36    12|1111111111111888888888
   14    13|444444
    8    14|00000007

          1|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.8: Stem and leaf plot of $\hat{L}(multi-diff\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff}$ = .3122. n=200, N=50, h=5.

```
n = 200  Min=0.320307  QL=0.832799  M=1.02498  QU=1.15311  Max=1.53748
    1     3|2
    2     4|4
    4     5|77
    8     6|4444
   44     7|00000000000000006666666666666666666666
   74     8|33333333333333999999999999999
   95     9|666666666666666666666
 ( 47)   10|22222222222222222222222288888888888888888888888888
   58    11|55555555555555555555555
   35    12|11111111111118888888
   15    13|444444444
    6    14|0000
    2    15|33

          1|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.9: Stem and leaf plot of $\hat{L}(SM\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff}$ = .3122. n=200, N=50, h=5.

```
n = 200  Min=0  QL=0  M=0.877193  QU=1.75439  Max=3.50877
   57   +0*|000000000000000000000000000000000000000000000000000000
 ( 74)  +0.|88888888888888888888888888888888888888888888888888888888*
   69    1*|
   69    1.|77777777777777777777777777777777777777777777
   21    2*|
   21    2.|6666666666666
    8    3*|
    8    3.|55555555

         1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.10: Stem and leaf plot of $\hat{L}(multi-diff\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff} = 0.0228$. n=200, N=50, h=1, $\mu_1 = 4$.

```
n = 200  Min=0  QL=0  M=0.877193  QU=1.75439  Max=4.38596
   59   +0*|000000000000000000000000000000000000000000000000000000
 ( 71)  +0.|88888888888888888888888888888888888888888888888888888888*
   70    1*|
   70    1.|77777777777777777777777777777777777777777777
   26    2*|
   26    2.|666666666666666666
    8    3*|
    8    3.|555555
    2    4*|33

         1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.11: Stem and leaf plot of $\hat{L}(SM\ ,r)/\hat{L}_{diff}$. $\hat{L}_{diff} = 0.0228$. n=200, N=50, h=1, $\mu_1 = 4$.

```
n = 200  Min=0.784303  QL=0.96057  M=0.997965  QU=1.02689  Max=1.06882
    2    7.|88
    2    8*|
    5    8t|222
   10    8f|55555
   10    8s|
   16    8.|888888
   29    9*|1111111111111
   42    9t|3333333333333
   42    9f|
   62    9s|66666666666666666666
 ( 55)   9.|8888888888888888888888888888999999999999999999999999999
   83   10*|11111111111111111111
   60   10t|22222222222222222333333333333333333
   27   10f|444444444555555
   12   10s|666666666666

         1*|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.12: Stem and leaf plot of $SE(L, multi-diff\ , r)/SE(L_{diff})$. $SE(L_{diff}) = .0661$. n=200, N=50, h=1.

```
n = 200  Min=0.784303  QL=0.96057  M=0.997965  QU=1.02689  Max=1.06968
    2    7.|88
    2    8*|
    5    8t|222
    7    8f|55
    7    8s|
   15    8.|88888888
   29    9*|11111111111111
   48    9t|3333333333333333333
   48    9f|
   70    9s|666666666666666666666
 ( 58)   9.|8888888888888888888888888899999999999999999999999999999999
   72   10*|11111111111111111
   55   10t|22222222222222333333333333333
   26   10f|44444444555555
   12   10s|666666666666

         1*|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.13: Stem and leaf plot of $SE(L, SM\ , r)/SE(L_{diff})$. $SE(L_{diff}) = .0661$. n=200, N=50, h=1.

```
n = 200  Min=0.801522  QL=0.958994  M=1.00189  QU=1.04943  Max=1.08966
    2    8*|00
    5    8t|333
    5    8f|
    9    8s|7777
    9    8.|
   21    9*|000000000000
   38    9t|33333333333333333
   56    9f|555555555555555555
   56    9s|
   77    9.|888888888888888888
 ( 46)  10*|00000000000000000000000001111111111111111111
   77   10t|3333333333333333333333
   54   10f|444444444444444444
   36   10s|666666666666677777777777777777
    8   10.|88888888

          1*|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.14: Stem and leaf plot of $SE(L, multi-diff\ , r)/SE(L_{diff})$. $SE(L_{diff}) = .0646$. n=200, N=50, h=5.

```
n = 200  Min=0.655892  QL=0.958994  M=1.01987  QU=1.04943  Max=1.09229
    1    6.|5
    1    7*|
    2    7.|5
    4    8*|33
    8    8.|7777
   44    9*|00000000000000003333333333333333333
   74    9.|55555555555555888888888888888
 ( 91)  10*|000000000000000000001111111111111111111111113333333333333333*
   35   10.|6666666666666777777777777777888899

          1*|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.15: Stem and leaf plot of $SE(L, SM\ , r)/SE(L_{diff})$. $SE(L_{diff}) = .0646$. n=200, N=50, h=5.

```
n = 200  Min=0  QL=0  M=0.950019  QU=1.32976  Max=2.03574
   59   +0*|00000000000000000000000000000000000000000000000000000000000000
   59   +0t|
   59   +0f|
   59   +0s|
 ( 71)  +0.|99999999999999999999999999999999999999999999999999999999999999*
   70    1*|
   70    1t|33333333333333333333333333333333333333333333333
   26    1f|
   26    1s|666666666666666666
    8    1.|888888
    2    2*|00

          1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.16: Stem and leaf plot of $SE(L, multi-diff\ , r)/SE(L_{diff})$. $SE(L_{diff}) = .0208$. n=200, N=50, h=1. $\mu_1 = 4.0$.

```
n = 200  Min=0  QL=0  M=0.950019  QU=1.32976  Max=1.84097
   57    +0|00000000000000000000000000000000000000000000000000000000000000
   57     1|
   57     2|
   57     3|
   57     4|
   57     5|
   57     6|
   57     7|
   57     8|
 ( 74)    9|55555555555555555555555555555555555555555555555555555555555555*
   69    10|
   69    11|
   69    12|
   69    13|22222222222222222222222222222222222222222222222222
   21    14|
   21    15|
   21    16|1111111111111
    8    17|
    8    18|44444444

           1|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.17: Stem and leaf plot of $SE(L, SM\ , r)/SE(L_{diff})$. $SE(L_{diff}) = .0208$. n=200, N=50, h=1. $\mu_1 = 4.0$.

```
n = 200  Min=-3.26859  QL=0.301056  M=1.19794  QU=1.90667  Max=3.64002
    1   -3*|2
    3   -2.|66
    6   -2*|332
   12   -1.|999977
   16   -1*|3221
   22   -0.|888765
   37   -0*|333332222200000
   59   +0*|000002222222333333334
   92   +0.|55555555566666666777788888999999
 ( 29)   1*|000111112222222222333333444444
   79    1.|555555666666777777788888888999999999
   41    2*|00011112222222333444444
   18    2.|5566777788
    8    3*|023444
    2    3.|56

         1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.18: Stem and leaf plot of $\hat{\kappa}_{b,w}(multi-diff\ ,r)/\hat{\kappa}_{b,w,diff}$. $\hat{\kappa}_{b,w,diff} = .17523$. n=200, N=50, h=1.

```
n = 200  Min=-3.81335  QL=0.286001  M=1.20421  QU=2.00201  Max=3.73045
    1   -3.|8
    1   -3*|
    2   -2.|5
    6   -2*|3220
    9   -1.|966
   14   -1*|44433
   28   -0.|99888776666555
   41   -0*|4333333200000
   59   +0*|000002222222333333
   87   +0.|555566666677888888899999999
 ( 31)   1*|000011111122222233333333344444
   82    1.|55566667777777778888889999999
   52    2*|00000011111112222222333333444
   22    2.|5555566677777888
    6    3*|0123
    2    3.|57

         1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.19: Stem and leaf plot of $\hat{\kappa}_{b,w}(SM\ ,r)/\hat{\kappa}_{b,w,diff}$. $\hat{\kappa}_{b,w,diff} = .17523$. n=200, N=50, h=1.

```
n = 200  Min=-5.18792  QL=0.368559  M=1.20141  QU=1.86419  Max=3.72177
    1   -5*|1
    1   -4.|
    1   -4*|
    3   -3.|95
    4   -3*|0
    7   -2.|666
    9   -2*|10
   10   -1.|5
   16   -1*|444211
   19   -0.|976
   35   -0*|3333333322000000
   54   +0*|0000000223333333344
   86   +0.|555566666666777888888899999999999
 ( 40)   1*|00000000111112222222222222233334444444444
   74    1.|55555555566666777777778889999
   46    2*|000000111112222222344444
   22    2.|566667777788999
    7    3*|0114
    3    3.|567

          1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.20: Stem and leaf plot of $\hat{\kappa}_{b,w}(multi-diff\ )/\hat{\kappa}_{b,w,diff}$. $\hat{\kappa}_{b,w,diff}$ = .1752 n=200, N=50, h=5.

```
n = 200  Min=-5.18792  QL=0.543496  M=1.1889  QU=1.82615  Max=4.34797
    1    -5|1
    3    -4|82
    4    -3|9
   11    -2|8662110
   19    -1|99731100
   36    -0|77664433333320000
   87    +0|000002233345555555666666667777777888888888899999999
 ( 71)    1|00000011111111222222222233444445555555556666667777777777788888*
   42     2|000001112233334444444556677788888999
    6     3|01269
    1     4|3

          1|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.21: Stem and leaf plot of $\hat{\kappa}_{b,w}(SM\ ,r)/\hat{\kappa}_{b,w,diff}$. $\hat{\kappa}_{b,w,diff}$ = .1752 n=200, N=50, h=5.

```
n = 200  Min=0.568628  QL=0.795693  M=0.990719  QU=1.17995  Max=2.9175
    6   +0f|555555
   51   +0s|666666666666666667777777777777777777777777777
 ( 53)  +0.|88888888888888888889999999999999999999999999999
   96    1*|000000000000000000000000000000011111111111111111
   44    1t|2222222222223333333
   25    1f|44455555
   17    1s|66677
   12    1.|8899
    8    2*|11
    6    2t|2233
    2    2f|4
    1    2s|
    1    2.|9

         1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.22: Stem and leaf plot of $SE(\kappa_{b,w}, bst - diff\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff}) =$ .2291. n=200, N=50, h=1, B=200.

```
n = 200  Min=0.521324  QL=0.781249  M=0.998014  QU=1.2076  Max=2.37813
    1     5|2
   19     6|011222356888889999
   55     7|000011222344444455556667777778888899
   81     8|00001112444444556777777999
  100     9|0022233334567788899
  100    10|000111222333444445555566777889
   69    11|011222344567899
   54    12|000002244556899
   39    13|011223556777
   27    14|0568
   23    15|011268
   17    16|117
   14    17|07
   12    18|229
    9    19|5
    8    20|9
    7    21|2
    6    22|034
    3    23|067

          1|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.23: Stem and leaf plot of $SE(\kappa_{b,w}, SB\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff}) = .2291$. n=200, N=50, h=1, B=200.

```
n = 200  Min=0.509025  QL=0.790337  M=0.964898  QU=1.09611  Max=2.43091
   11     5|01124578888
   32     6|000011123344445666678
   55     7|11223444456666788899999
   83     8|011111222222244456667789999
 ( 35)    9|000122233334455666677777888888999999
   82    10|00122222233344556677788888888899
   50    11|001222233555677889
   32    12|126678899
   23    13|0334666
   16    14|23
   14    15|167
   11    16|79
    9    17|2258
    5    18|5
    4    19|5
    3    20|6
    2    21|4
    1    22|
    1    23|
    1    24|3

          1|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.24: Stem and leaf plot of $SE(\kappa_{b,w}, gjk - diff\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff}) =$ .2291. n=200, N=50, h=1.

```
n = 200  Min=0.518724  QL=0.798912  M=0.929118  QU=1.12494  Max=2.02703
   10     5|1477778889
   25     6|001111233466679
   50     7|011111233444444566688999
   84     8|00111111122233344455566667788888999
 ( 32)    9|000000111122222222344555677778888
   84    10|12222222223333344556777788899999
   52    11|022344555556788
   37    12|01124457788899
   23    13|000566
   17    14|119
   14    15|126
   11    16|0
   10    17|0366
    6    18|4579
    2    19|6
    1    20|2

          1|1  represents 0.11  Leaf digit unit = 0.01
```

Figure 4.25: Stem and leaf plot of $SE(\kappa_{b,w}, SGJK\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff}) =$ .2291. n=200, N=50, h=1.

```
n = 200  Min=0.483642  QL=0.763  M=0.951585  QU=1.16203  Max=3.86332
    5   +0f|45555
   66   +0s|666666666666666666677777777777777777777777777777777777777777
 ( 53)  +0.|88888888888888888888899999999999999999999999999999
   81    1*|00000000000000000000011111111111111111
   43    1t|222222222222222223333
   22    1f|4445555
   15    1s|667
   12    1.|8889
    8    2*|1
    7    2t|23
    5    2f|445
    2    2s|7
    1    2.|
    1    3*|
    1    3t|
    1    3f|
    1    3s|
    1    3.|8

         1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.26: Stem and leaf plot of $SE(\kappa_{b,w}, bst - diff\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff}) =$ .2439. n=200, N=50, h=5, B=200.

```
n = 200  Min=0.469072  QL=0.768103  M=0.918342  QU=1.12518  Max=4.19184
    5   +0f|45555
   65   +0s|66666666666666666777777777777777777777777777777777777777777
 ( 57)  +0.|88888888888888888888888888999999999999999999999999999999
   78    1*|000000000000000000000011111111111111111
   38    1t|2222333333
   28    1f|44444555555
   17    1s|66666
   12    1.|88999
    7    2*|01
    5    2t|33
    3    2f|
    3    2s|66
    1    2.|
    1    3*|
    1    3t|
    1    3f|
    1    3s|
    1    3.|
    1    4*|1

         1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.27: Stem and leaf plot of $SE(\kappa_{b,w}, SB\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff}) = .2439$. n=200, N=50, h=5, B=200.

```
n = 200  Min=0.459264  QL=0.742993  M=0.882711  QU=1.06718  Max=2.65502
   14   +0f|44455555555555
   69   +0s|66666666666666666666777777777777777777777777777777777
 ( 69)  +0.|88888888888888888888888888888888899999999999999999999999*
   62    1*|00000000000000001111111111111111
   29    1t|222222233333
   17    1f|44445
   12    1s|667777
    6    1.|8899
    2    2*|0
    1    2t|
    1    2f|
    1    2s|6

         1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.28: Stem and leaf plot of $SE(\kappa_{b,w}, gjk - diff\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff}) = .2439$. n=200, N=50, h=50.

```
n = 200  Min=0.433956  QL=0.751922  M=0.908239  QU=1.06948  Max=2.53969
   16   +0f|4444555555555555
   68   +0s|6666666666666666666777777777777777777777777777777777
 ( 65)  +0.|888888888888888888888888888899999999999999999999999999999*
   67    1*|00000000000000000000000011111111111111
   28    1t|22233333333
   17    1f|444455
   11    1s|667
    8    1.|8889
    4    2*|11
    2    2t|3
    1    2f|5

         1*|1  represents 1.1  Leaf digit unit = 0.1
```

Figure 4.29: Stem and leaf plot of $SE(\kappa_{b,w}, SGJK\ , r)/SE(\kappa_{b,w,diff})$. $SE(\kappa_{b,w,diff}) = .2439$. n=200, N=50, h=5.

4.7.6 Scatterplots to show the results of the simulations

Scatterplots are another way to summarize the data in the simulation. Only some of the many scatterplots examined are shown here. The symbols in the scatterplot go as *,2,3,4, ..., 9, A, B, ..., Z, a, b, ..., z, #, and then cycle. * is used to denote 1. The scatterplots plot the results of SM vs. multi-diff for $\hat{L}$, $\hat{\kappa}_{b,w}$, and $SE(L)$. The multinomial techniques are preferred over the bootstrap and the grouped jackknife if the multinomial formulae are available. $SE(\kappa_{b,w})$ cannot be evaluated using the multinomial techniques, because the multinomial results have not been derived. $SE(\kappa_{b,w})$ is computed using the SGJK, gjk-diff, SB, and bst-diff techniques. The results obtained using the SB technique are plotted against those of the bst-diff technique. The results obtained using the SGJK technique are plotted against those of the gjk-diff and the bst-diff technique. Since the estimates of $SE(\kappa_{b,w})$ obtained using the grouped jackknife are not guaranteed to converge to the multinomial results, the results of the gjk-diff technique are plotted against those of the bst-diff technique, so we can compare the results of the techniques.

All the scatterplots show a strong linear relationship and the correlation coefficient r is usually greater than 0.8. Examining the scatterplots enables us to judge the effect of increasing h while splitting and recombining the cases, increasing h_1 for the grouped jackknife in the $SGJK$ method, and increasing the number of bootstrap replications (B).

The correlation coefficient is high (approximately 0.8) for all plots except in Figures 4.41 where there is one outlier point. Re-doing that plot with a different random seed [Figure 4.42] removes that outlier. That outlier is not surprising considering the number of plots drawn and comparisons made (including those not shown in the thesis).

We use scatterplots to compare the standard errors obtained by the SM, SGJK, and SB methods with the standard errors obtained using the multi-diff, the bst-diff, and the gjk-diff methods. Scatterplots are drawn and given in Figures 4.30 to 4.58.

4.7.6.1 Scatterplots of $\hat{L}$

Figure 4.30 gives the scatterplot of $\hat{L}(SM\ , r)$ versus $\hat{L}(multi-diff\ , r)$. A reasonably linear plot with a high correlation coefficient ($r = 0.836$) is obtained, showing that the SR conjecture is plausible. h, the number of cases left out, is 1. The intercept (b0) is close to 0 and the slope (b1) is close to 1.0, as required, since the two methods should report equal values of $\hat{L}$.

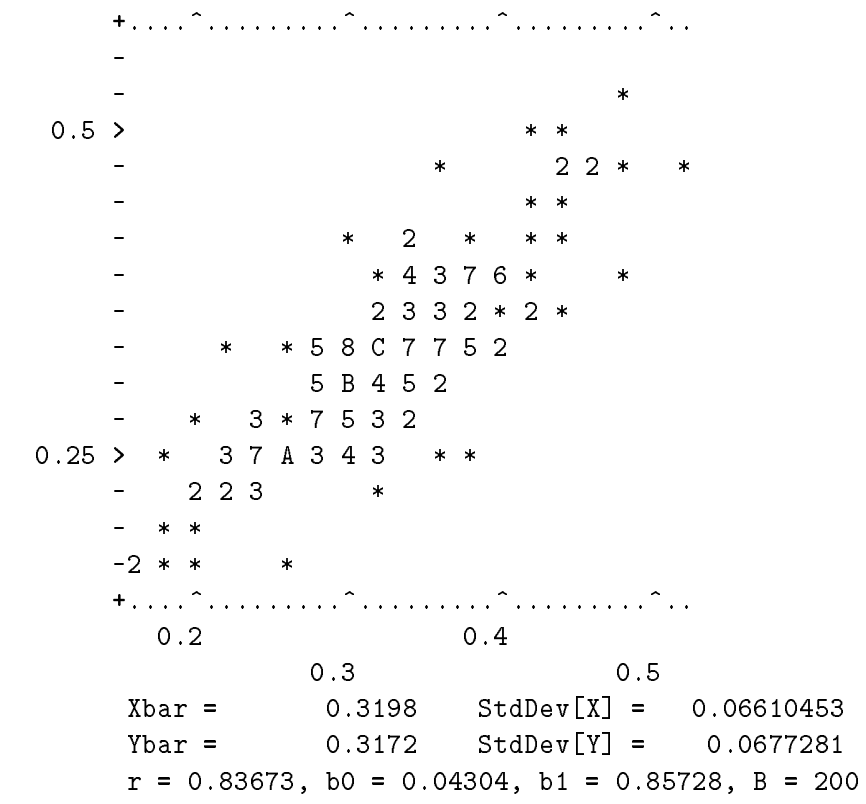

Figure 4.30: Scatterplot of $\hat{L}(multi-split)$ vs. $\hat{L}(multi-diff)$. n = 50, h = 1, $\mu_1 = 1$.

Figure 4.31 gives the scatterplot of $\hat{L}(multi-split)$ versus $\hat{L}(multi-diff)$. A reasonably linear plot with a high correlation coefficient ($r = 0.819$) is obtained, showing that the SR conjecture is plausible. h, the number of cases left out, is 5. The correlation coefficient is less than that obtained when $h = 1$, showing that it is preferable to have as small a value of h as computationally feasible. The intercept (b0) is close to 0 and the slope (b1) is close to 1.0, as required since the two methods should report equal standard values of L.

Similar results are obtained for the scatterplots where $\mu_1 = 2.5$. The plots show a strong linear relationship. For $h = 1, r = 0.82626$. For $h = 5, r = 0.81402$.

For the scatterplots where $\mu_1 = 4$ we have for $h = 1, r = 0.76505$ and for $h = 5, r = 0.80791$.

Figures 4.32 and 4.33 are plots of $\hat{L}(multi-split)$ vs. $\hat{L}(multi-diff)$. We notice that there are only six (five) possible values for $\hat{L}(multi-split)$ and $\hat{L}(multi-diff)$ leading to considerable "clumping" in the figures. This is reasonable because $\hat{L}(multi-split)$ and $\hat{L}(multi-diff)$ depend on only the number of incorrect classifications (since $L_{01} = L_{10} = 1$). In some of the other figures this clumping may be observed to a lesser extent. This is because $\hat{L}$, $SE(L)$, $\hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$ depend on only the number of incorrect classifications, and the number of incorrect classifications is finite.

Similar results are obtained when the sample size is increased from 50 to 630 [Figures 4.34 and 4.35]. The clumping in the scatterplots for $\mu_1 = 4$ is reduced considerably although

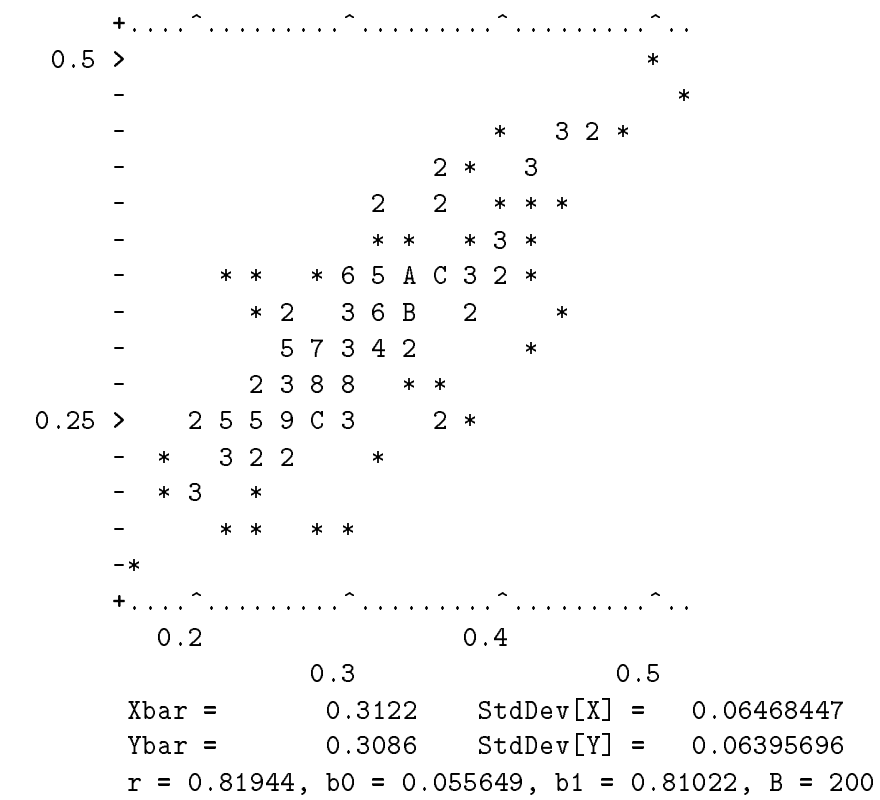

Figure 4.31: Scatterplot of $\hat{L}(multi-split)$ vs. $\hat{L}(multi-diff)$. n = 50, h = 5, $\mu_1 = 1$.

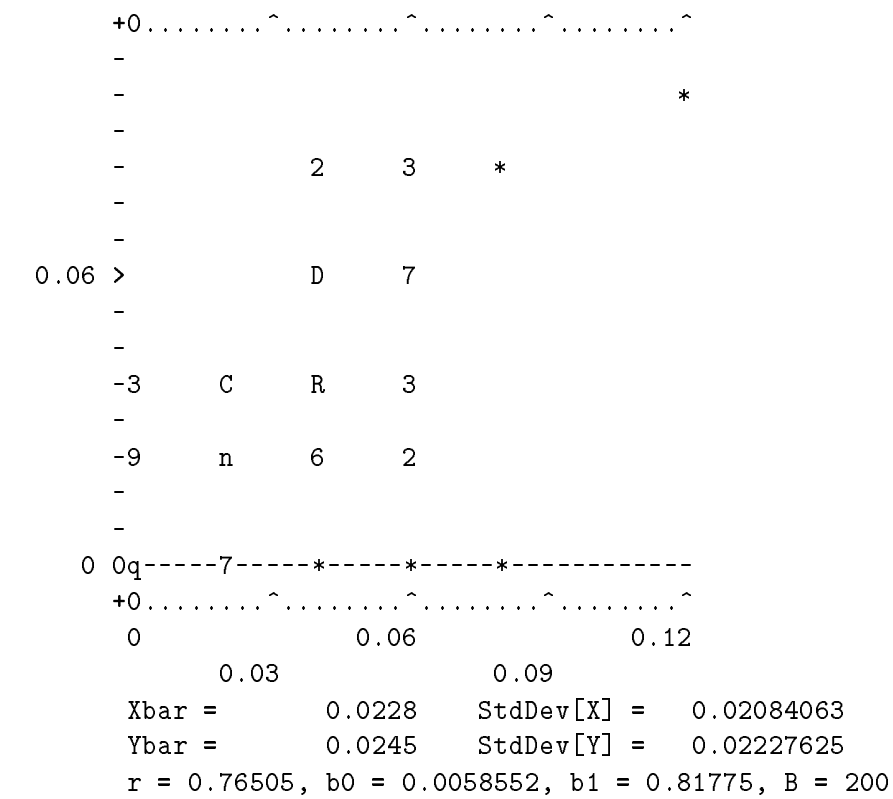

Figure 4.32: Scatterplot of $\hat{L}(multi - split)$ vs. $\hat{L}(multi - diff)$. n = 50, h = 1, $\mu_1 = 4$.

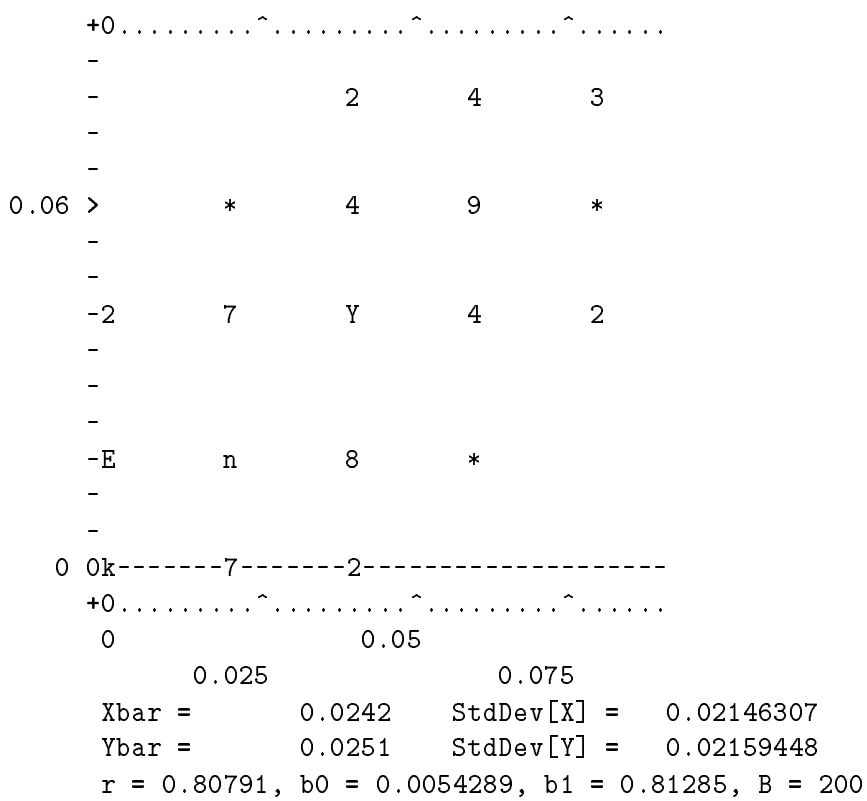

Figure 4.33: Scatterplot of $\hat{L}(multi - split)$ vs. $\hat{L}(multi - diff)$. n = 50, h = 5, $\mu_1 = 4$.

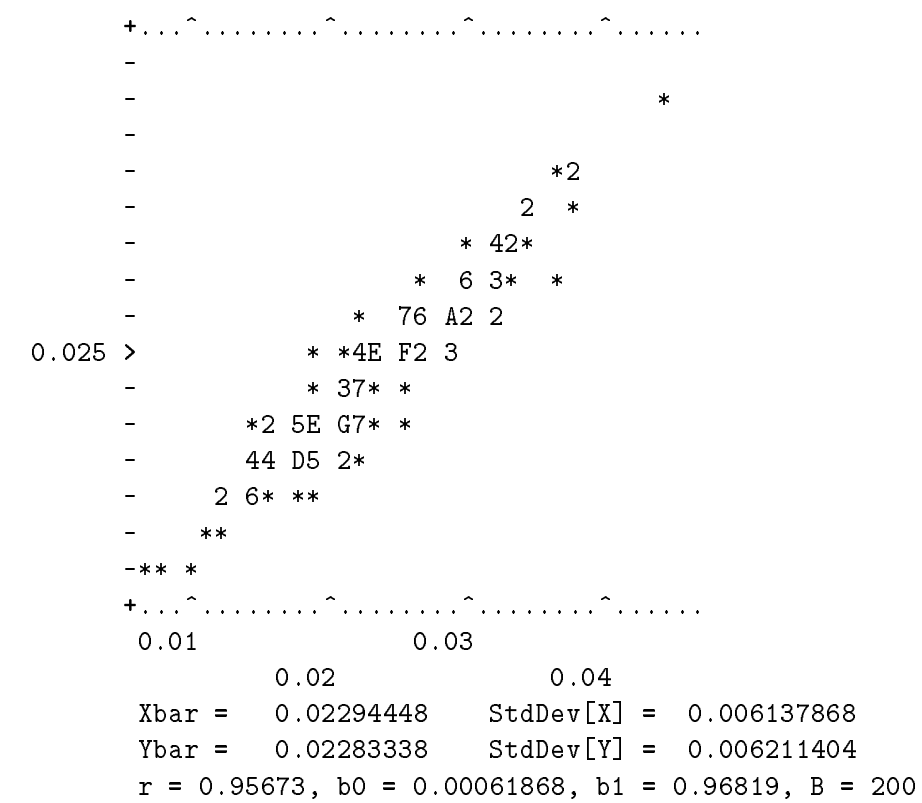

Figure 4.34: Scatterplot of $\hat{L}(multi - split)$ vs. $\hat{L}(multi - diff)$. n = 630 , h = 1, $\mu_1 = 4$.

there are vacant vertical columns, indicating some clumping.

All the scatterplots above show that the SR conjecture is plausible and we can estimate $\hat{L}$ using the split recombination procedure. Since the results of computing $\hat{L}, SE(L)$, and $\hat{\kappa}_{b,w}$ using the grouped jackknife and bootstrap techniques only approximates the results of the multinomial results, the results of the gjk-diff, SGJK, bst-diff, and SB techniques for $\hat{L}, SE(L)$, and $\hat{\kappa}_{b,w}$ are not shown. In any case, since we are neglecting the bias correction for $\hat{L}$ and $\hat{\kappa}_{b,w}$ in the grouped jackknife and the bootstrap, the grouped jackknife and the bootstrap give values that are identical with the multinomial technique for $\hat{L}$ and $\hat{\kappa}_{b,w}$.

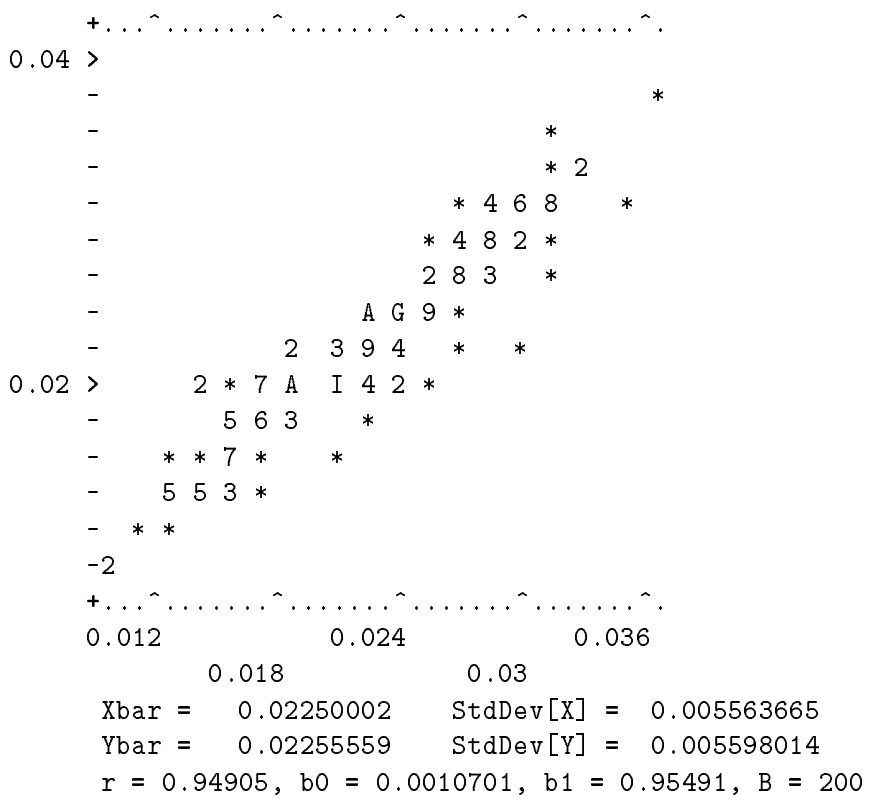

Figure 4.35: Scatterplot of $\hat{L}(multi-split)$ vs. $\hat{L}(multi-diff)$. n = 630 , h = 21, $\mu_1 = 4$.

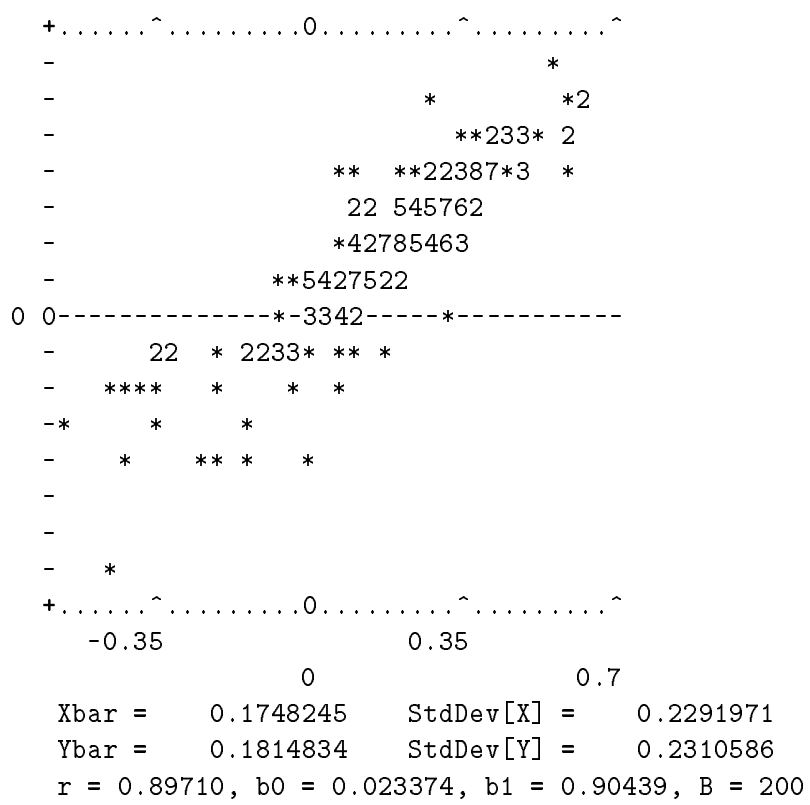

Figure 4.36: Scatterplot of $\hat{\kappa}_{b,w}(multi-split)$ vs. $\hat{\kappa}_{b,w}(multi-diff)$. n = 50, h = 1, $\mu_1 = 1$.

4.7.6.2 Scatterplots of $\hat{\kappa}_{b,w}$

Figure 4.36 gives the scatterplot of $\hat{\kappa}_{b,w}(multi - split)$ versus $\hat{\kappa}_{b,w}(multi - diff)$. A reasonably linear plot with a high correlation coefficient ($r = 0.897$) is obtained, showing that the SR conjecture is plausible. h, the number of cases left out, is 1.

Figure 4.37 gives the scatterplot of $\hat{\kappa}_{b,w}(multi - split)$ versus $\hat{\kappa}_{b,w}(multi - diff)$ for $h = 5$. The plot shows a reasonably strong linear relationship. The correlation coefficient ($r = 0.876$) is lower than that for $h = 1$ [Figure 4.36], showing that it is preferable to use as small an h as computationally feasible.

Similar results are obtained for $\mu_1 = 2.5$. $r = 0.84267$ for $h = 1$, $r = 0.84$ for $h = 5$. The shape of the plots is similar to that for $\mu_1 = 1$ in Figures 4.36 and 4.37.

Similar results are obtained for $\mu_1 = 4.0$ [Figures 4.38 and 4.39]. $r = .78163$ for $h = 1$, $r = 0.80913$ for $h = 5$.

In Figures 4.38 and 4.39 we notice many occurences of $\hat{\kappa}_{b,w} = 1$ (the maximum). This is because with $\mu_1 = 4$, there are many perfectly classified samples during the replications of the experiment, as the two classes are spaced far apart ($\mu_0 = 0, \mu_1 = 4$).

From the above figures, after observing the high values of the correlation coefficient r, and the fact that the intercept (b0) is close to zero, and the slope (b1) is close to 1, we conclude that the SR conjecture can be used to estimate $\hat{\kappa}_{b,w}$.

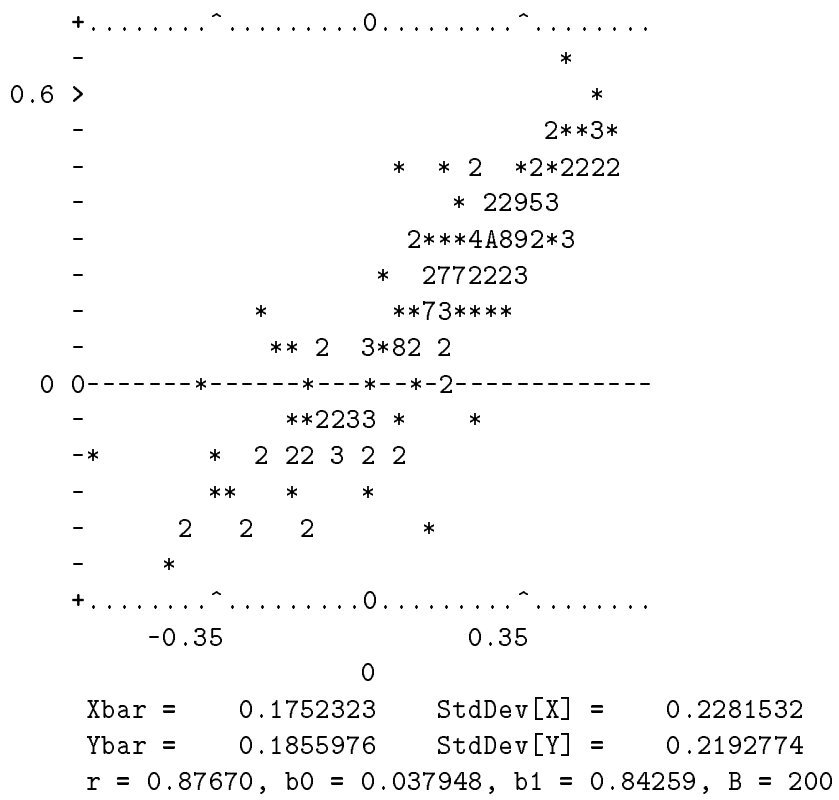

Figure 4.37: Scatterplot of $\hat{\kappa}_{b,w}(multi-split)$ vs. $\hat{\kappa}_{b,w}(multi-diff)$. n = 50, h = 5, $\mu_1 = 1$.

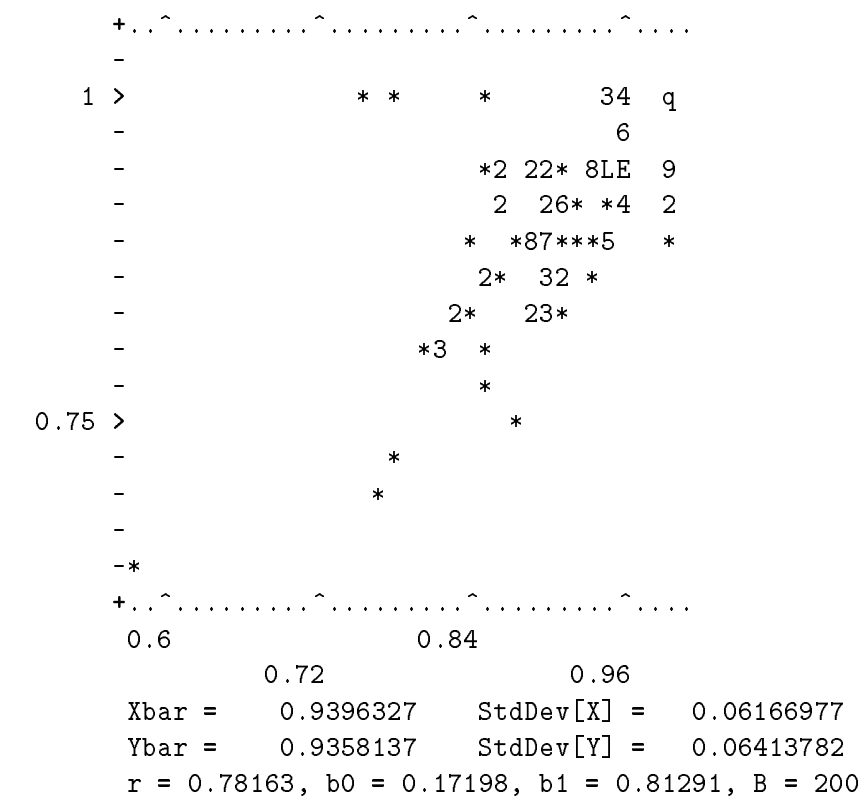

Figure 4.38: Scatterplot of $\hat{\kappa}_{b,w}(multi-split)$ vs. $\hat{\kappa}_{b,w}(multi-diff)$. n = 50, h = 1, $\mu_1 = 4$.

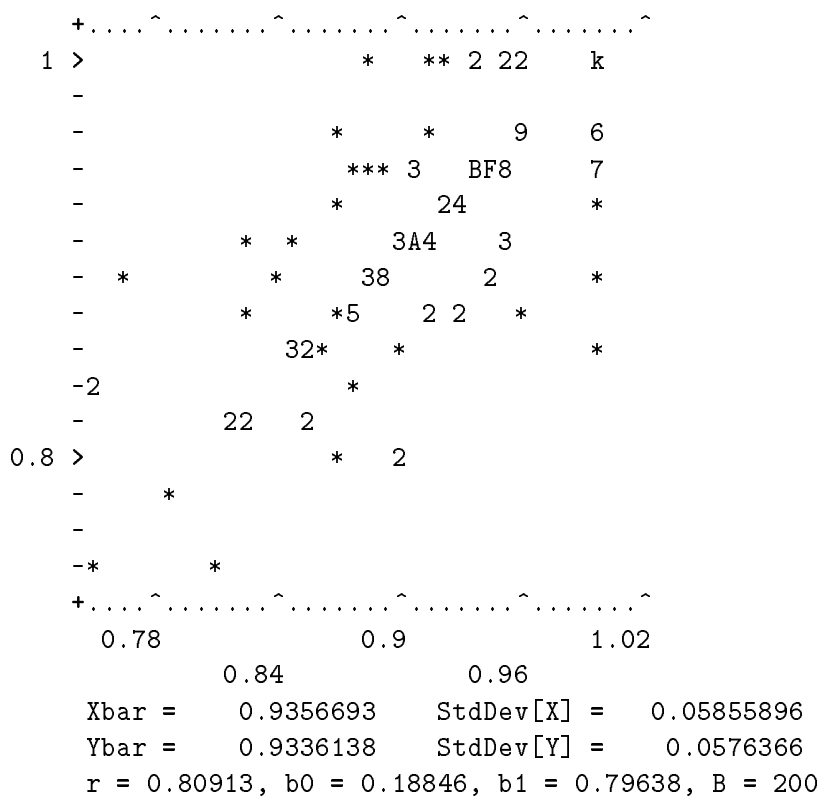

Figure 4.39: Scatterplot of $\hat{\kappa}_{b,w}(multi-split)$ vs. $\hat{\kappa}_{b,w}(multi-diff)$. n = 50, h = 5, $\mu_1 = 4$.

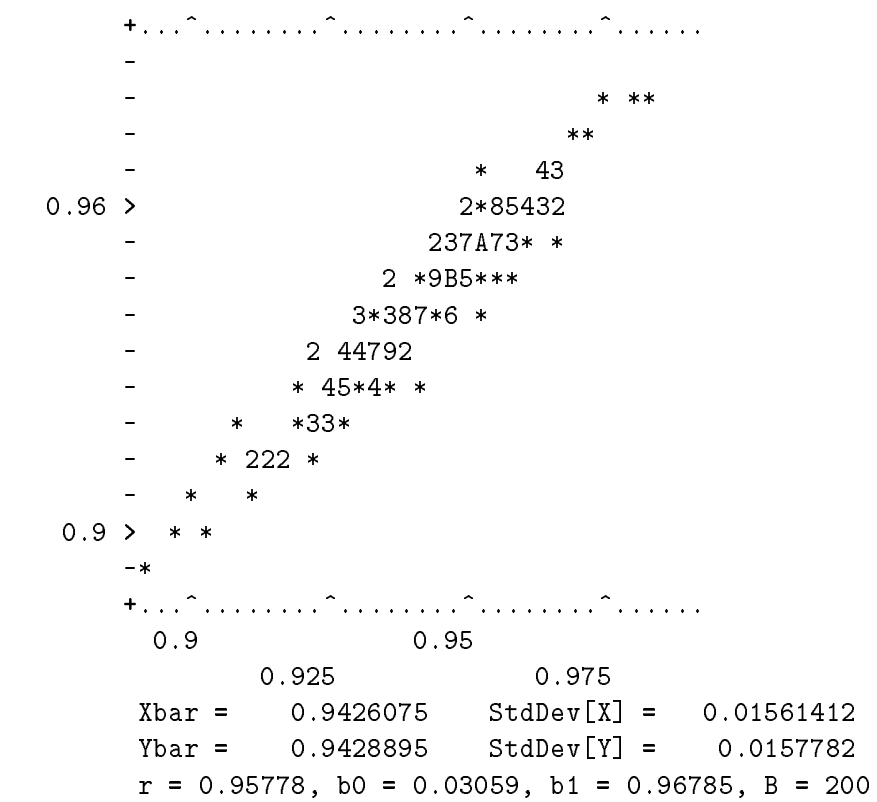

Figure 4.40: Scatterplot of $\hat{\kappa}_{b,w}(multi-split)$ vs. $\hat{\kappa}_{b,w}(multi-diff)$. n = 630 , h = 1, $\mu_1 = 4$.

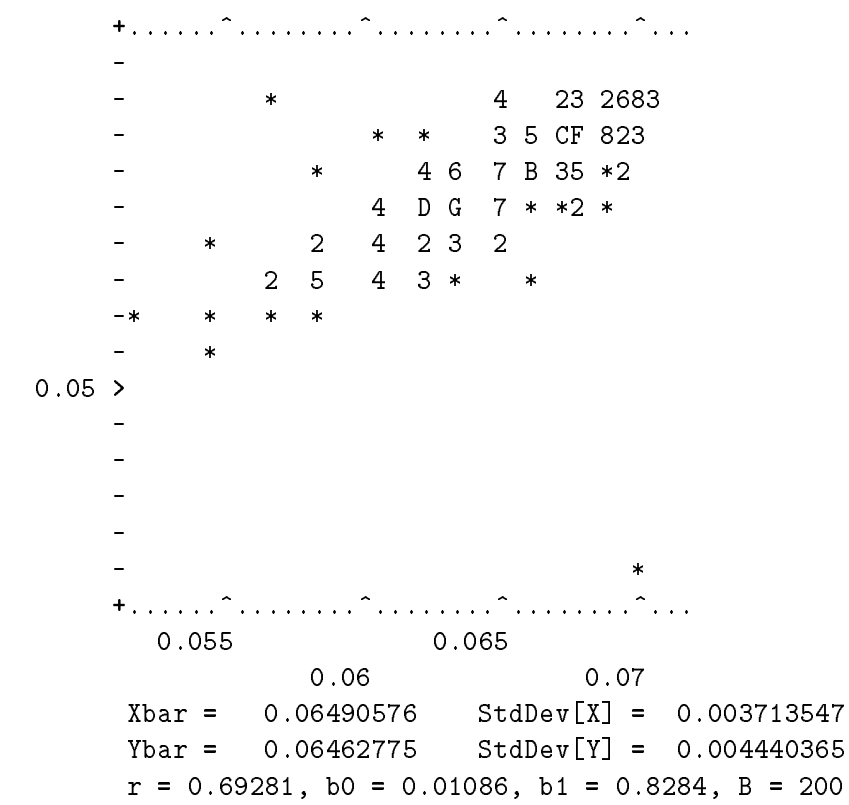

Figure 4.41: Scatterplot of $SE(L, multi - split)$ vs. $SE(L, multi - diff)$. n = 50, h = 1, $\mu_1 = 1$.

4.7.6.3 Plots of $SE(L)$

Figure 4.41 is a plot of $SE(L, multi - split)$ vs. $SE(L, multi - diff)$. There is one outlier point in the plot. Examination of the simulation data reveals that this is a plot where $\hat{L}$ is high (about .9) and therefore $SE(L)$ is low. This can be considered an extreme case (we have had 39 simulations of $\hat{\theta}(method - 1)$ vs. $\hat{\theta}(method - 2)$ and $SE(\theta, method - 1)$ vs. $SE(\theta, method - 2)$ with each simulation containing 200 replications). This outlier has been observed only once (in 7800 replications). (The results of all the simulations done have not been presented in the thesis). Using another random sample results in this outlier not being present, and the correlation coefficient goes up to 0.827 [Figure 4.42].

We see that whether we use $h = 1$ or $h = 5$ ($h = 21$) when the sample set is of size 50 (630) we get reasonably high correlation coefficients when using large sample sets although $h = 1$ does give better results. Thus while using $h = 1$ is preferable, we can use $h > 1$ to save computational resources. After observing the high values of the correlation coefficient r, and the fact that the intercept (b0) is close to zero, and the slope (b1) is close to 1, we conclude that the SR conjecture is plausible and can be used to estimate $SE(L)$.

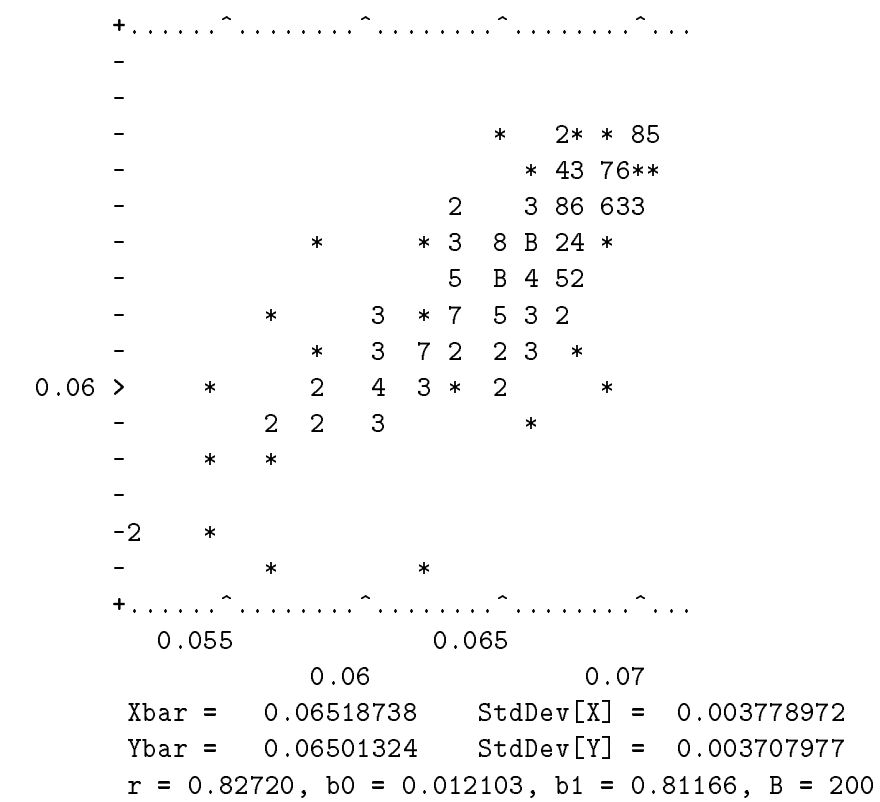

Figure 4.42: $SE(L, multi-split)$ vs. $SE(L, multi-diff)$. n = 50, h = 1, $\mu_1 = 1$. Random sample two.

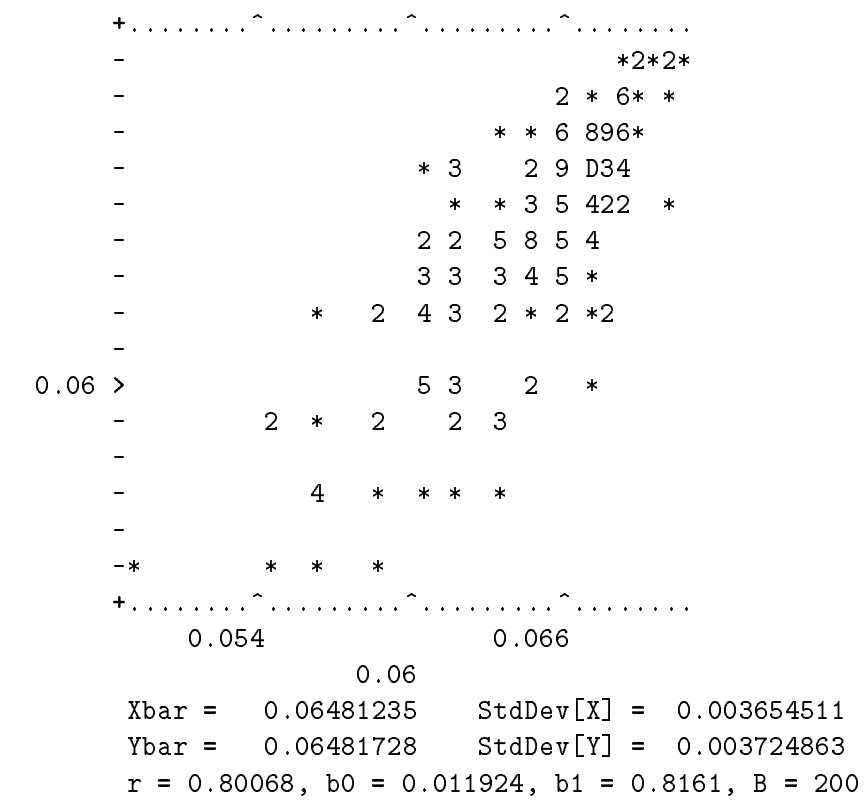

Figure 4.43: Scatterplot of $SE(L, multi - split)$ vs. $SE(L, multi - diff)$. n = 50, h = 5, $\mu_1 = 1$.

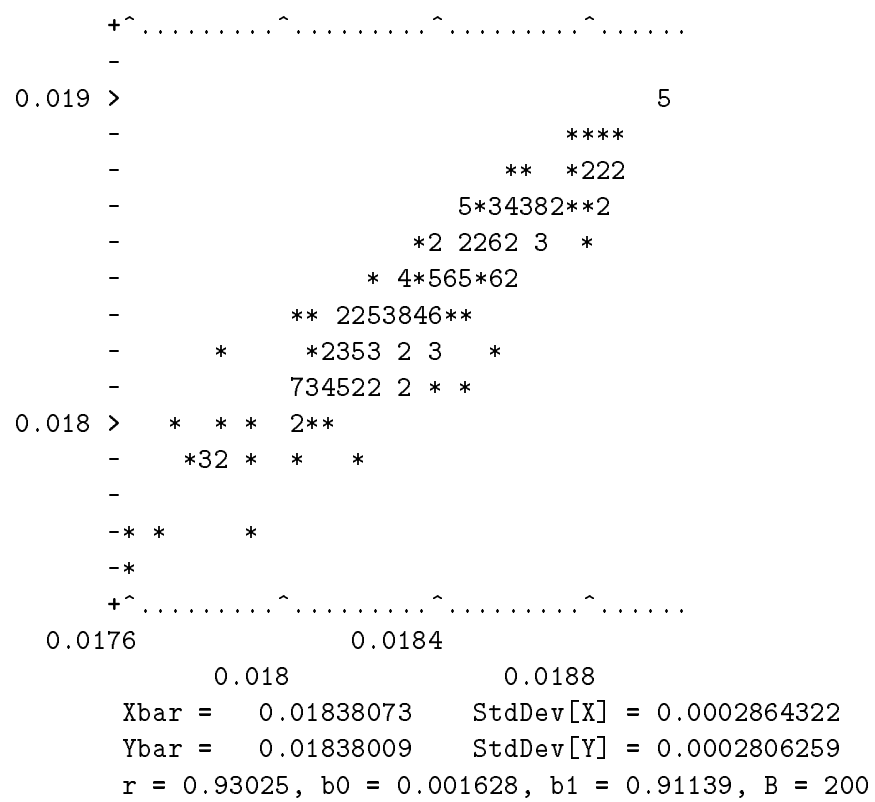

Figure 4.44: Scatterplot of $SE(L, multi-split)$ vs. $SE(L, multi-diff)$. n = 630 , h = 1, $\mu_1 = 1$.

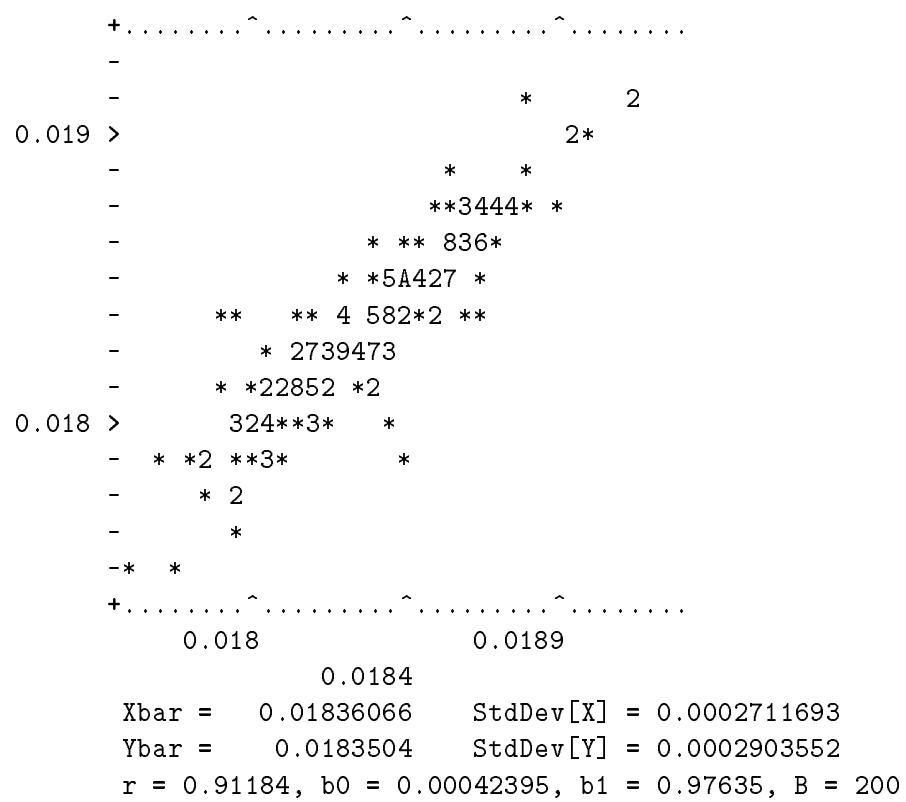

Figure 4.45: Scatterplot of $SE(L, multi - split)$ vs. $SE(L, multi - diff)$. n = 630 , h = 21, $\mu_1 = 1$.

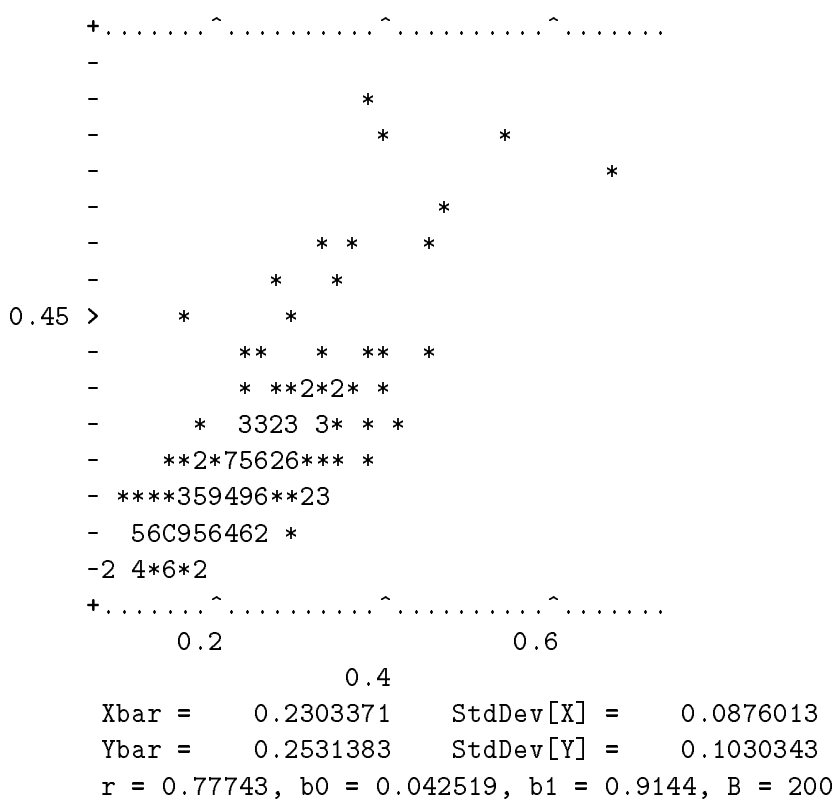

Figure 4.46: Scatterplot of $SE(\kappa_{b,w}, bst - diff)$ vs. $SE(\kappa_{b,w}, gjk - diff)$. n = 50, h = 5, B = 200, $\mu_1 = 1$.

4.7.6.4 Plots of $SE(\kappa_{b,w})$

$SE(\kappa_{b,w})$ cannot be computed using the SM technique based on multinomial results because the multinomial results are not known. Instead, we use the grouped jackknife (which reduces to the jackknife when $h = 1$) and the bootstrap. If the multinomial results were known, then the bootstrap estimates of $SE(\kappa_{b,w})$ would approach that of the multinomial results as the number of replications, $B \to \infty$. This is not true for the grouped jackknife (or the jackknife). Since the use of the SGJK technique can save computational resources we shall investigate results using the SGJK technique in addition to investigating the SB technique. The SB technique has a theoretical advantage compared to the SGJK technique for estimating $SE(\kappa_{b,w})$, i.e., it approaches the multinomial results as $B \to \infty$.

Figure 4.46 shows that there is a reasonably good correlation ($r = 0.777$) between $SE(\kappa_{b,w})$ as estimated by the grouped jackknife using $h = 5$ and the bootstrap using $B = 200$.

Figure 4.47 shows that there is a good correlation ($r = .988$) between $SE(\kappa_{b,w})$ as estimated by the (grouped) jackknife using $h = 1$ and the bootstrap using $B = 1000$. Observing this high correlation we conclude that the grouped jackknife can be used for estimating $SE(\kappa_{b,w})$ for practical purposes even though theoretically speaking the grouped jackknife is not guaranteed to approximate the multinomial results (if they were available).

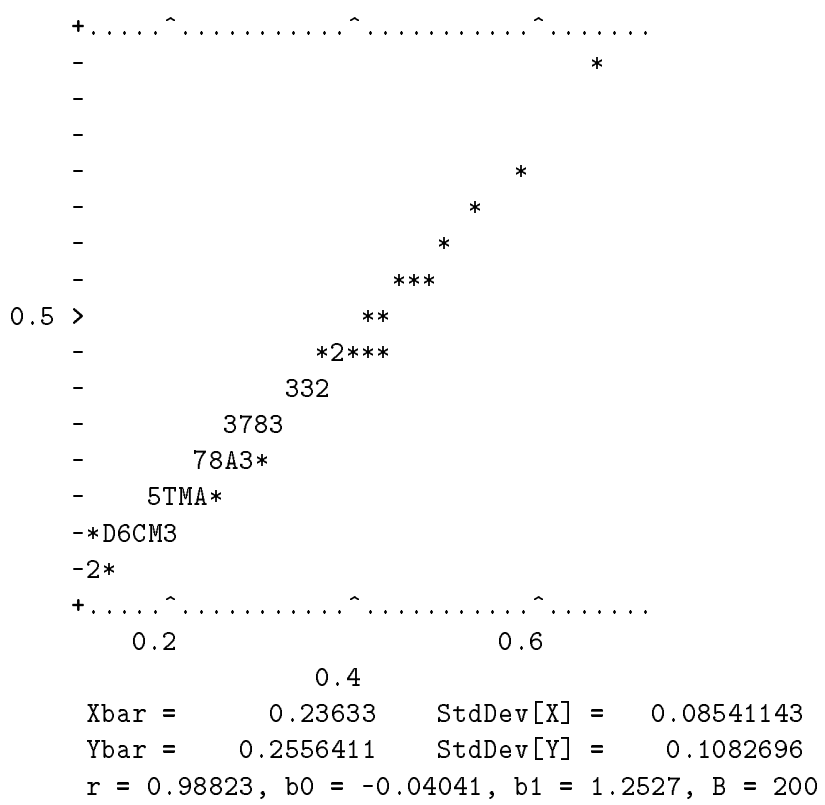

Figure 4.47: Scatterplot of $SE(\kappa_{b,w}, bst - diff)$ vs. $SE(\kappa_{b,w}, gjk - diff)$. n = 50, h = 1, B = 1000, $\mu_1 = 1$.

In the past, when high-speed digital computers were not so easily available, this would have been a useful observation as saving computational resources was important. The grouped jackknife can save computational resources as compared to the bootstrap. Nowadays, the computing power required for the bootstrap is easily available, especially since the information in S' can be represented in a contingency matrix, and the calculations done in memory. Therefore the bootstrap is preferred to the grouped jackknife.

Figure 4.48 is a plot of $SE(\kappa_{b,w}, gjk - split)$ vs. $SE(\kappa_{b,w}, gjk - diff)$ with $h = h_s = 1$. The correlation coefficient (r) is 0.875. Figure 4.49 is a plot of $SE(\kappa_{b,w}, gjk - split)$ vs. $SE(\kappa_{b,w}, gjk - diff)$ with $h = h_s = 5$. The correlation coefficient (r) is 0.867. Figure 4.50 is a plot of $SE(\kappa_{b,w}, gjk - split)$ vs. $SE(\kappa_{b,w}, gjk - diff)$ with $h = 5, h_s = 1$. The correlation coefficient (r) is 0.864. Figures 4.48, 4.49, and 4.50 have high correlation coefficients (r = 0.875, .866, and .864 respectively) and reasonably linear plots.

The three choices of h and h_s yield almost equivalent results except that $h = 5, h_s = 1$ has fewer outliers.

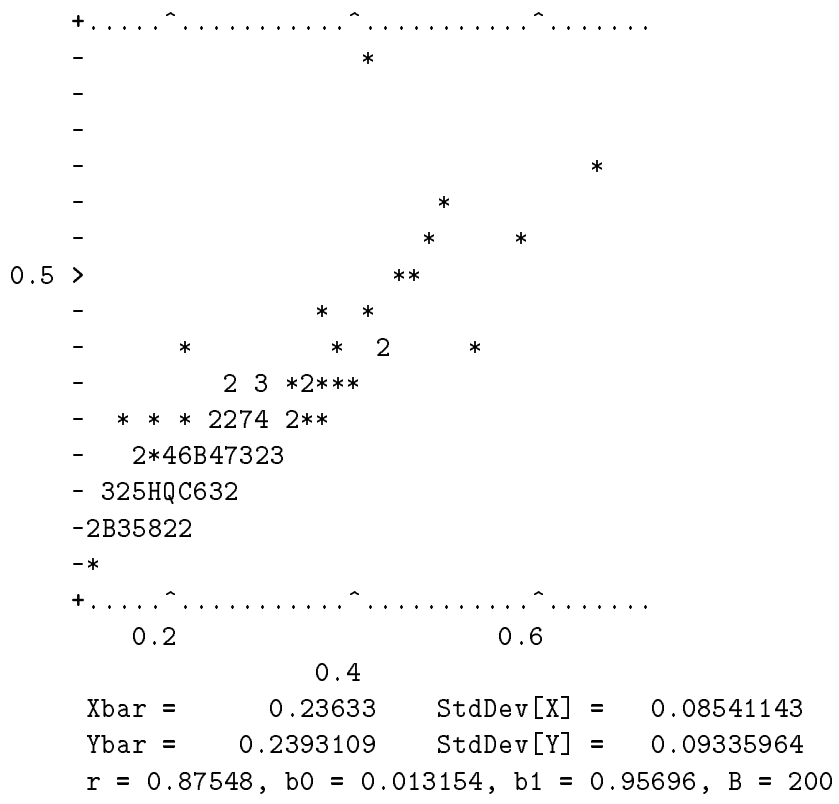

Figure 4.48: Scatterplot of $SE(\kappa_{b,w}, gjk - split)$ vs. $SE(\kappa_{b,w}, gjk - diff)$. n = 50, h = 1, $\mu_1 = 1$.

Figure 4.51 is a scatterplot of $SE(\kappa_{b,w}, bst - split)$ vs. $SE(\kappa_{b,w}, bst - diff)$. $h = 5$ and 200 replications are used. $r = 0.839$ indicates a reasonably good correlation. Figure 4.52 is a scatterplot of $SE(\kappa_{b,w}, bst - split)$ vs. $SE(\kappa_{b,w}, bst - diff)$. $h = 1$ and 1000 replications are used. $r = 0.864$ indicates a reasonably good correlation. A larger number of replications $B = 1000$ instead of $B = 200$ leads to an increase in r from 0.839 to 0.864. This is because the bootstrap errors are unpredictable and $B = 1000$ gives fewer bootstrap errors than $B = 200$. Equation 4.5 enables us to compute how many bootstrap replications are needed to estimate the standard error with a required degree of accuracy.

Figures 4.53, 4.54, and 4.55 show that there is a high correlation between the DIFF methods and the corresponding SPLIT methods showing that the SR conjecture is plausible. The correlation coefficient for $SE(\kappa_{b,w})$ between the bst-diff method and the SB method is smaller than the correlation coefficient for $SE(\kappa_{b,w})$ between the gjk-diff method and the SGJK method. However, the correlation coefficient for $SE(\kappa_{b,w})$ between the bst-diff method and the SB method increases considerably as the number of replications is increased from $B = 200$ to $B = 1000$. For sample size $n = 630$ and $h = 1$ the plots of $SE(\kappa_{b,w}, bst - split)$ vs. $SE(\kappa_{b,w}, bst - diff)$ are given in Figures 4.56, 4.57. $r = .652$ for $B = 200$, and $r = 0.828$ for $B = 1000$, showing a substantial increase for larger B. For sample size $n = 630$ and $h = 21$ the plots of $SE(\kappa_{b,w}, bst - split)$ vs. $SE(\kappa_{b,w}, bst - diff)$

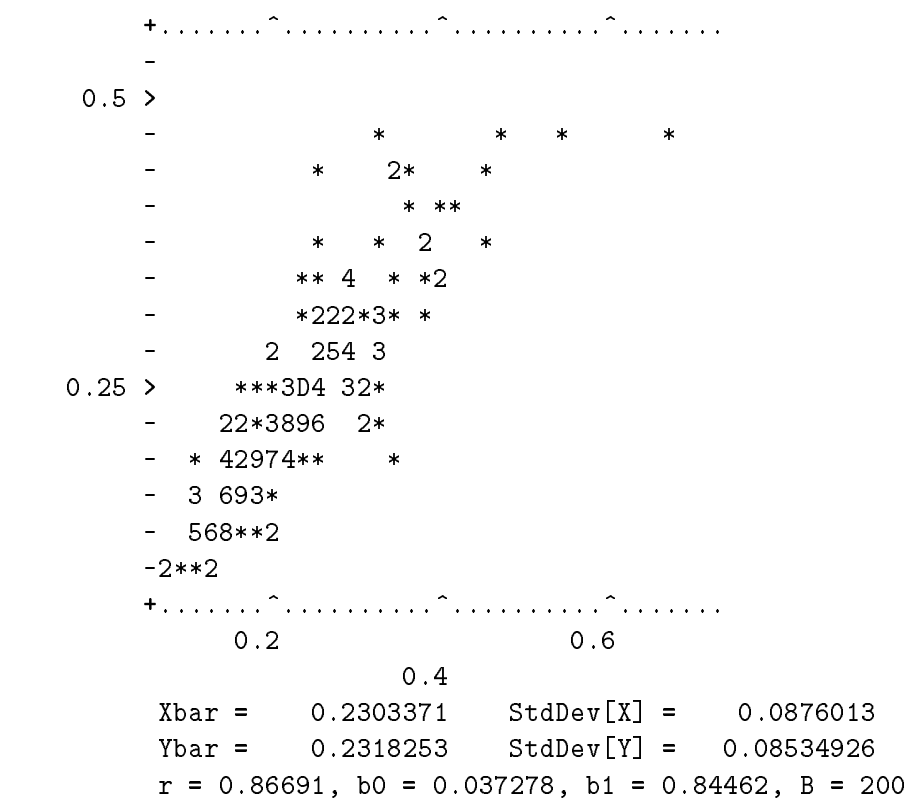

Figure 4.49: Scatterplot of $SE(\kappa_{b,w}, gjk - split)$ vs. $SE(\kappa_{b,w}, gjk - diff)$. n = 50, h = 5, $\mu_1 = 1$.

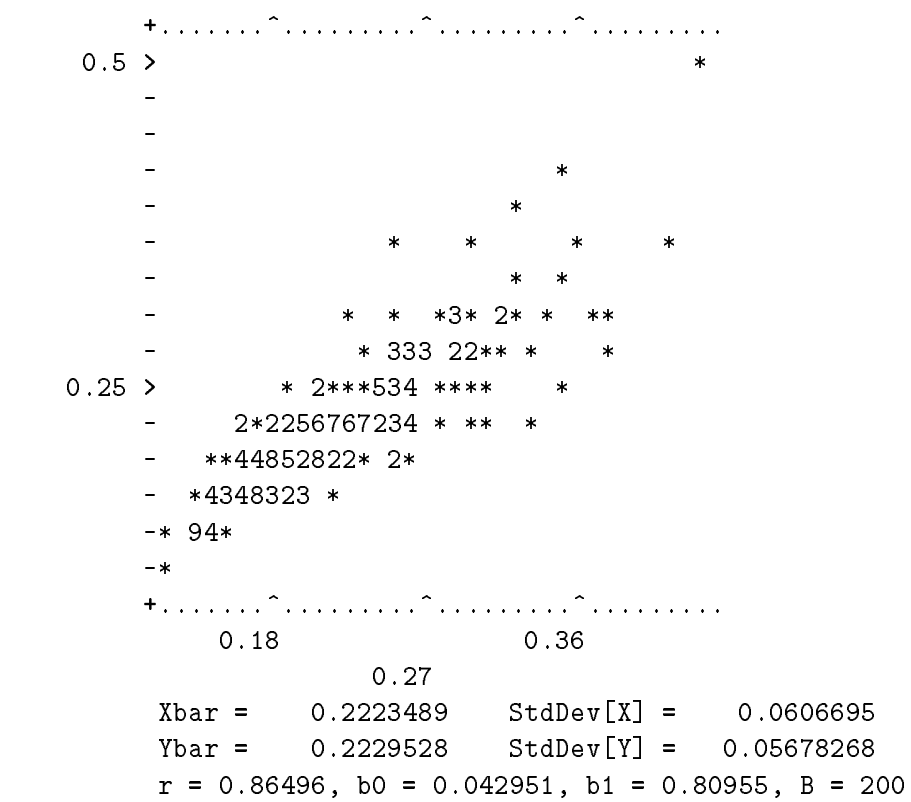

Figure 4.50: Scatterplot of $SE(\kappa_{b,w}, gjk - split)$ vs. $SE(\kappa_{b,w}, gjk - diff)$. n = 50, h = 5, $h_s = 1$, $\mu_1 = 1$.

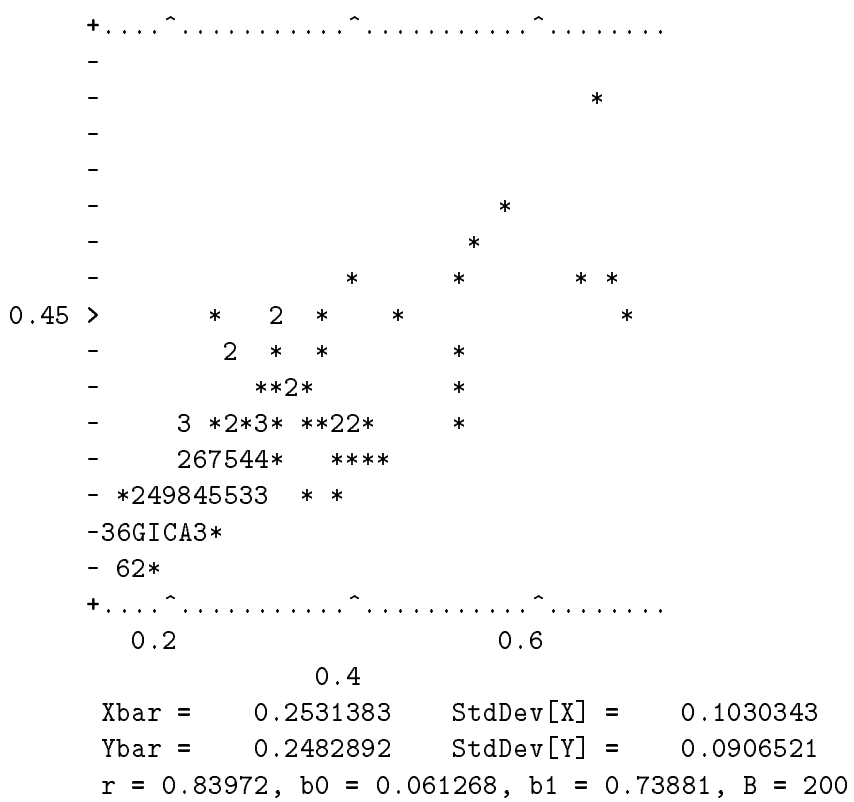

Figure 4.51: Scatterplot of $SE(\kappa_{b,w}, bst - split)$ vs. $SE(\kappa_{b,w}, bst - diff)$. n = 50, h = 5, B = 200, $\mu_1 = 1$.

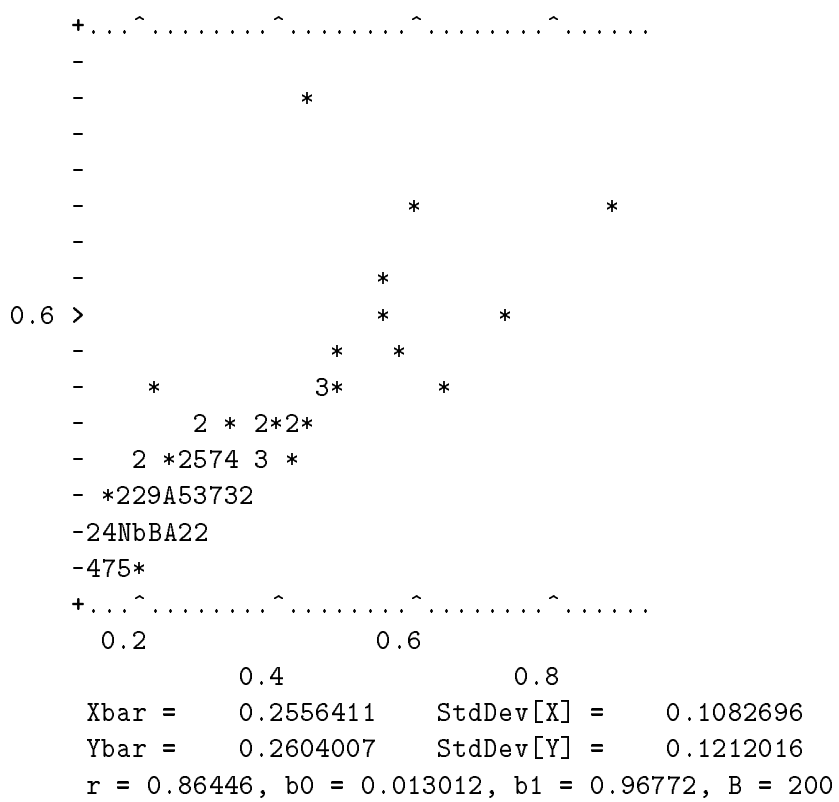

Figure 4.52: Scatterplot of $SE(\kappa_{b,w}, bst - split)$ vs. $SE(\kappa_{b,w}, bst - diff)$. n = 50, h = 1, B = 1000, $\mu_1 = 1$.

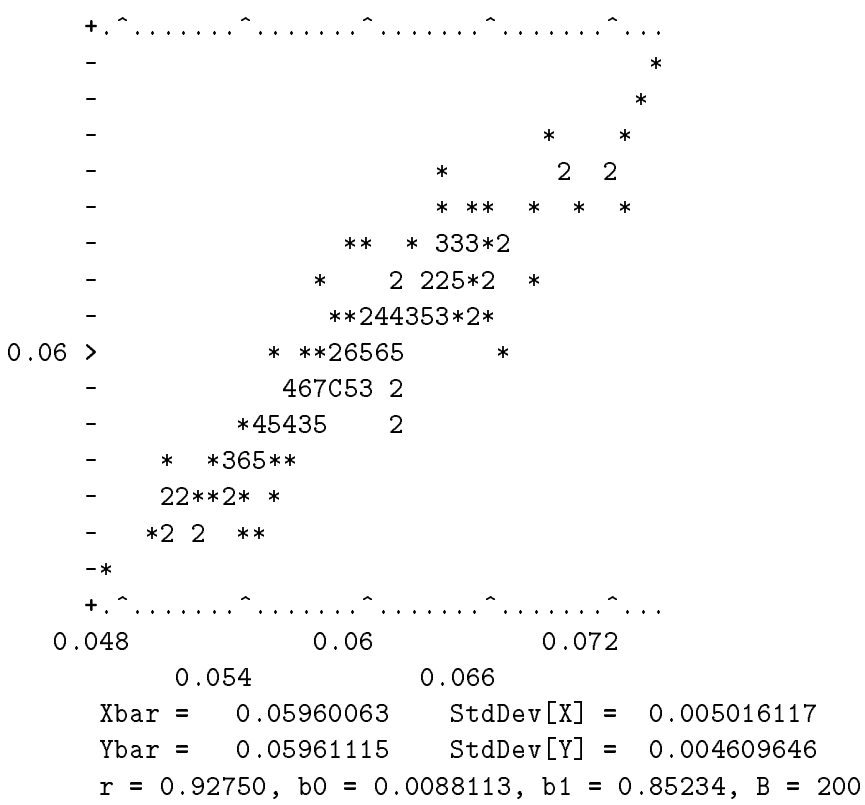

Figure 4.53: Scatterplot of $SE(\kappa_{b,w}, gjk - split)$ vs. $SE(\kappa_{b,w}, gjk - diff)$. n = 630 , h = 1, $\mu_1 = 1$.

are given in Figures 4.58 and 4.59. $r = 0.718$ for $B = 200$, and $r = 0.811$ for $B = 1000$, showing a substantial increase for larger B. The maximum kurtosis in these experiments is high (around 50) explaining the need for more replications. In Chapter 7, we see that for the AEPOSCH data the maximum absolute value of the kurtosis is less than 3. Thus we see that the kurtosis can vary significantly between different datasets.

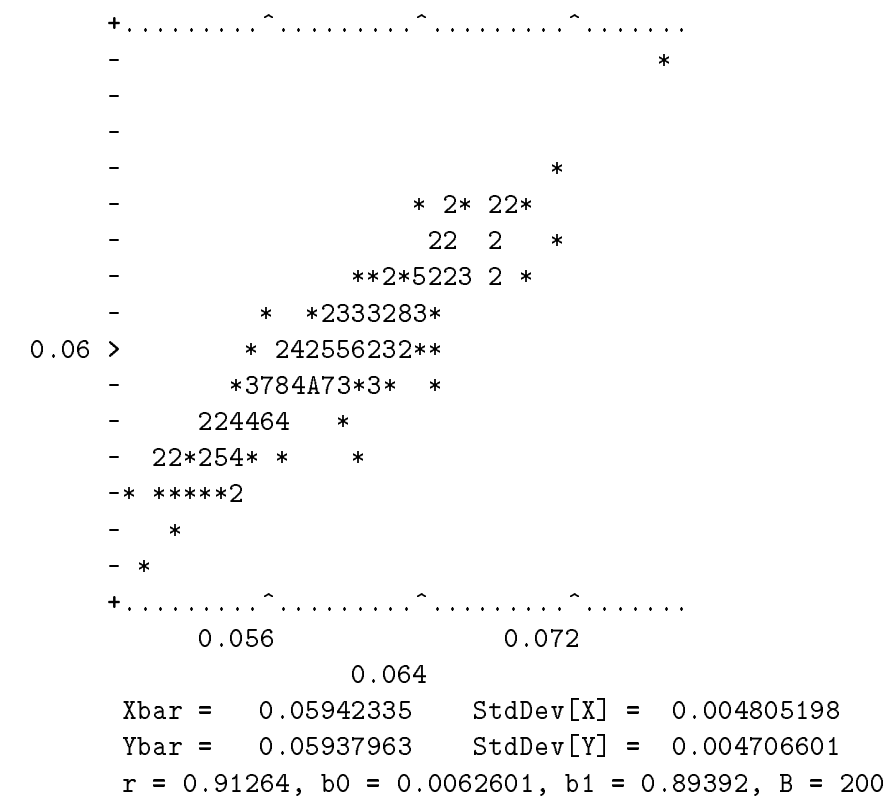

Figure 4.54: Scatterplot of $SE(\kappa_{b,w}, gjk - split)$ vs. $SE(\kappa_{b,w}, gjk - diff)$. n = 630 , h = 21, $h_s = 1$, $\mu_1 = 1$.

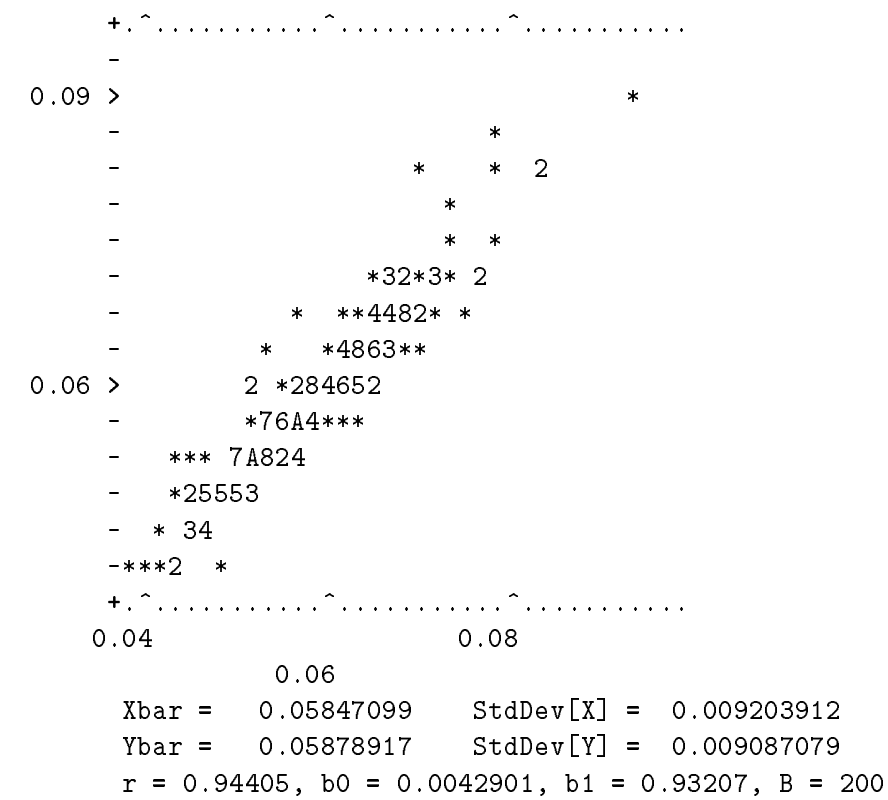

Figure 4.55: Scatterplot of $SE(\kappa_{b,w}, gjk - split)$ vs. $SE(\kappa_{b,w}, gjk - diff)$. n = 630 , h = 21, $\mu_1 = 1$.

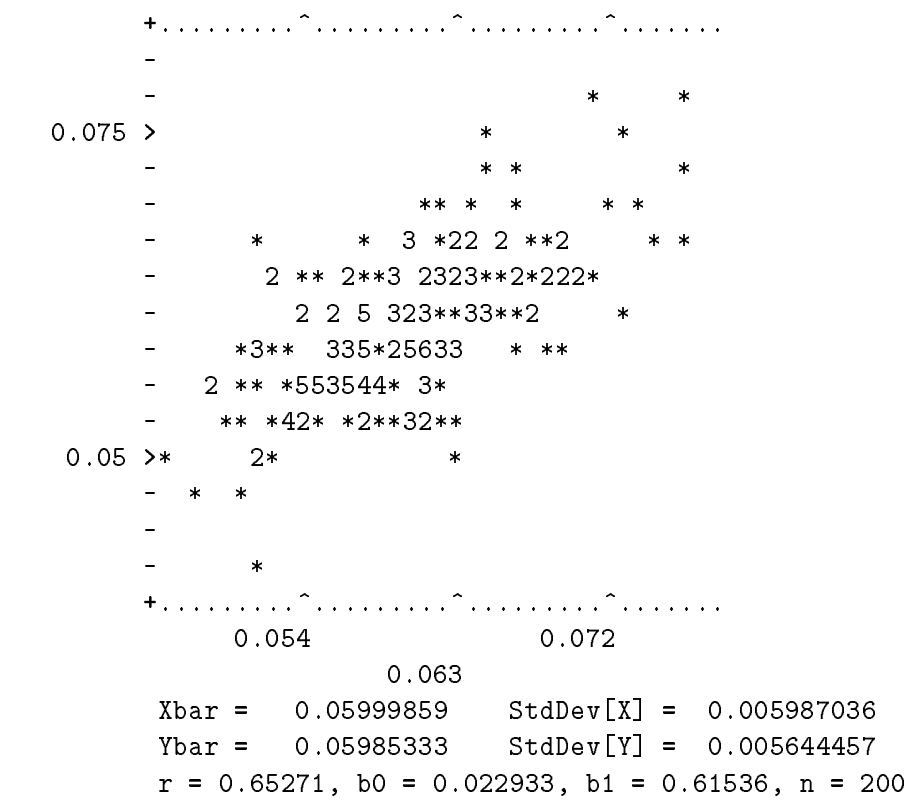

Figure 4.56: Scatterplot of $SE(\kappa_{b,w}, bst - split)$ vs. $SE(\kappa_{b,w}, bst - diff)$. n = 630 , h = 1, $\mu_1 = 1, B = 200$.

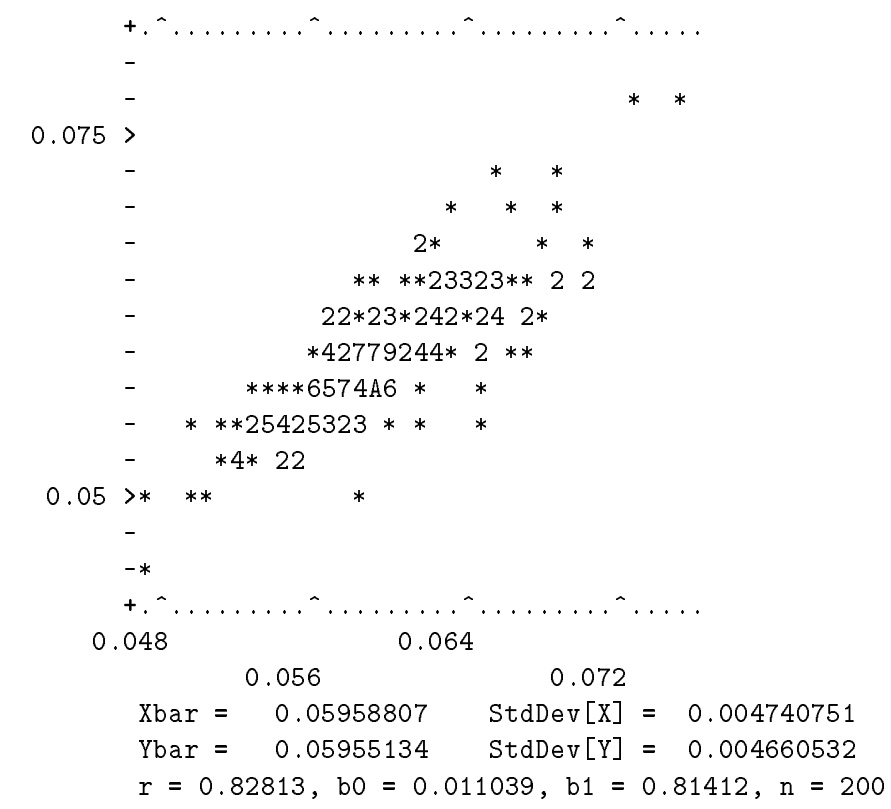

Figure 4.57: Scatterplot of $SE(\kappa_{b,w}, bst - split)$ vs. $SE(\kappa_{b,w}, bst - diff)$. n = 630 , h = 1, $\mu_1 = 1, B = 1000$.

4.7.7 Choosing among the method of repeated random halving, the SB method, and the SGJK method

As the results of the simulation show [Table 7.35], the SGJK method gives results that agree closely with the results of the gjk-diff method for $n = 50$, $h = 1$ or $h = 5$ and $n = 630$, $h = 1$ or $h = 21$ for less computation than the SB method. However, the estimate of $SE(\kappa_{b,w})$ by the SGJK method is not guaranteed to converge to the multinomial approximation if the multinomial formula is known.

The SB method requires many replications for accurate computation of $SE(\kappa_{b,w})$. Fortunately, this is not a severe constraint for today's computers. The RRH technique suffers from the difficulty that it severely limits the size of the training and test set. This reduction of cases in the training set leads to a pessimistic bias in the estimated loss for a trainable classifier [27, 28]. A reduction of cases in the test set leads to an increase in the standard error when estimating the loss [27, 28].

The SM, SGJK, and the SB methods allow almost all the samples present in the given sample to be used for training and testing a classifier. This results in a virtual doubling of the size of the sample set. These methods are based on the SR conjecture. The intuitive reasoning behind the SR conjecture is that the classifier trained on set G_{-i} is almost identical to the classifier trained on the entire set G for most real-world problems, especially if the

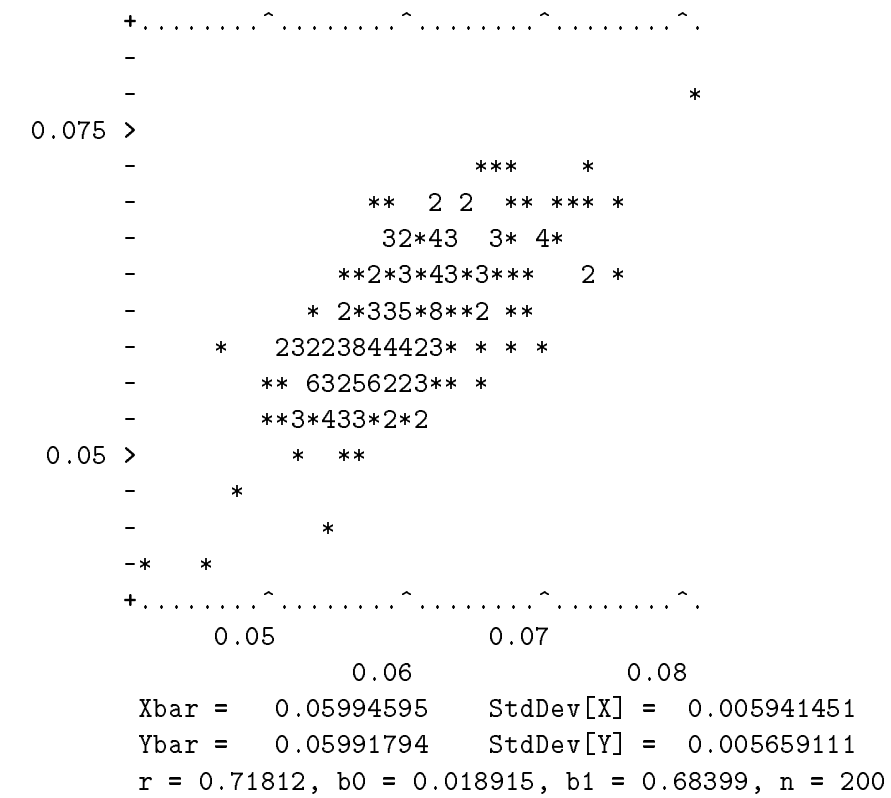

Figure 4.58: Scatterplot of $SE(\kappa_{b,w}, bst-split)$ vs. $SE(\kappa_{b,w}, bst-diff)$. n = 630 , h = 21, μ_1 = 1, B = 200.

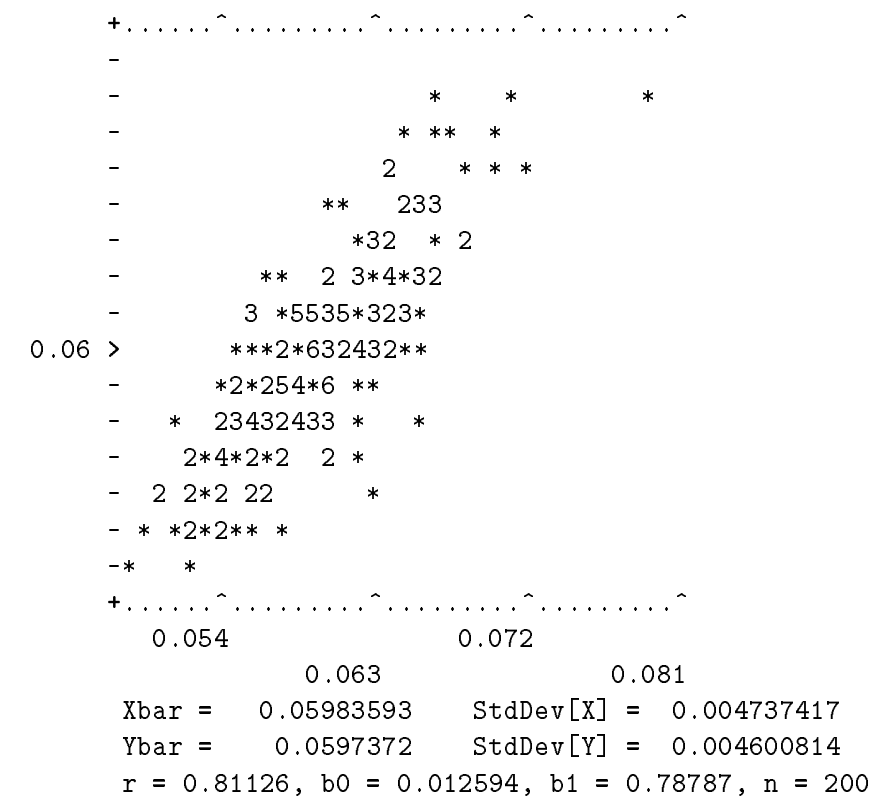

Figure 4.59: Scatterplot of $SE(\kappa_{b,w}, bst - split)$ vs. $SE(\kappa_{b,w}, bst - diff)$. n = 630 , h = 21, $\mu_1 = 1$, B = 1000

number of groups g is large. At the same time, the bootstrap or the grouped jackknife can be used for evaluating all the pairs $< y, y' >$ in S' since $< y, y' >$ represent gold standard and test evaluations of cases in S. The error shown in the bootstrap method for computing Expected Excess Error (Section 4.5.2) because of mixing of training and test cases does not occur here.

Since the estimates of $SE(\kappa_{b,w})$ obtained by the SB method are guaranteed to approach the multinomial results with increasing value of B, we use the SB technique to estimate $SE(\kappa_{b,w})$. A large number of replications, B, is needed to estimate $SE(\kappa_{b,w})$ as given by Equation 4.5. The convergence of the SB method to a single value with increasing number of replications should be observable for most real-world datasets.

4.8 Summary

In this chapter, we prove that the bootstrap method for estimating the expected excess error of a criterion for unseen cases is incorrect. This is because of a significant overlap of cases between the bootstrap replication samples and the bootstrap population. This leads to a breakdown of the bootstrap assumption that when a bootstrap replication sample is drawn from the entire available sample, then the relationship of the bootstrap replication sample to the entire available sample is analogous to the relationship of the entire available sample to the population. If the training set is large, and there is a continuity of properties between the training performance of the classifier and the test performance of the classifier then the consequences of this error are minor, because in the limit the test performance approaches the training performance as the number of cases increases. In some cases the training performance is sufficiently different from the test performance that this can cause a major difference. An example of this is given in the chapter.

To fill the need for a technique to estimate the value of criteria of a classifier over unseen (or test) cases we introduce the repeated random halving technique which should be correct because of its close correspondence to the bootstrap. The repeated random halving technique is not efficient in its use of the available cases to train and test the classifier. To efficiently use the sample set the SR conjecture is introduced to obtain an evaluated set of cases that can be used to compute $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ for the classifier based on multinomial results. This technique is known as the SM technique. The multinomial results to evaluate $SE(\kappa_{b,w})$ are not known. An experiment based on a very simple simulated dataset is done to show the plausibility of the SR conjecture. The experiment confirms the plausibility of the SR conjecture.

The SB method and the SGJK method are based on the SR conjecture and are introduced to compute statistics such as $SE(\kappa_{b,w})$ for which the multinomial results are not known. Between the SB method and the SGJK method, the SB method is preferred because with an increasing number of replications the results of the SB method approach that of the multinomial results. The SGJK method is based on the grouped jackknife and is not guaranteed to approximate the multinomial results for $SE(\kappa_{b,w})$ if they were known. Many replications, depending on the kurtosis and the accuracy desired, are needed for the SB method. With the large-scale availability of computing power this is fortunately not an issue in most cases. The kurtosis can vary significantly between different datasets and different criteria. The convergence of the SB method to a single value with an increasing number of replications should be observed.

The use of the SM, SGJK, and SB techniques almost doubles the effective size of the sample set that can be used for training and testing the classifiers. These techniques will be used to evaluate real-world classifiers and strategies for combining classifiers and the results presented in Chapter 7 but the theoretical validity of the SR conjecture needs to be proven.

Chapter 5

Improving performance by combining classifiers

We can try to improve performance by combining the assessments of more than one classifier. The strategy chosen might be an expert-defined ad-hoc strategy or it might be a criterion-based strategy. In this chapter we discuss two criterion-based strategies, the Classifier-based Overall Loss Minimization Strategy (COLMS), the Overall Loss Minimization Strategy (OLMS), and adaptations to them.

The COLMS is an independent discovery and development of the work done by Wernecke [83]. Wernecke's work does not deal with different degrees of mis-classification and deals only with classifiers. The OLMS can deal with general partitioning algorithms and is not limited to classifiers. A partitioning algorithm can partition the data into categories that are not necessarily identified with classes. A classifier is a special case of a partitioning algorithm. A specific ad-hoc expert-defined domain-specific strategy (EDSS) is introduced. Since an expert-defined strategy might decide to *reject* the assessment of all classifiers if there is a large discrepancy among the classifiers, a more general definition of loss than given in Equation 2.2 is required. We therefore define a procedure to specify the loss involved in rejecting a case.

5.1 Cartesian product of classes or partitions

A cartesian product of partitions is a special case of a partition. The COLMS is defined in the next section for such a partition. Consider a number of partitioning algorithms $P_1, \ldots, P_d$. A partitioning algorithm $P_p, p = 1, \ldots, d$ defines a function $h_p(\boldsymbol{t})$ with values $j = 1, \ldots, n_p$. This partitions $\boldsymbol{T} = \{\boldsymbol{t}\}$ into disjoint sets or cells $\delta_1, \ldots, \delta_{n_p}$ with

$$\delta_j = \{\boldsymbol{t} | h_p(\boldsymbol{t}) = j\}$$

$n_p, 1 \leq p \leq d$ is the number of cells that h_p will partition the data space into. In particular a classifier C can be considered a partitioning algorithm with

$$\mathcal{P} = \omega_1, \ldots, \omega_C$$

$$h(\boldsymbol{t}) = j, \text{when } C(\boldsymbol{t}) = \omega_j$$

Therefore

$$\delta_j = \{\boldsymbol{t} | C(\boldsymbol{t}) = \omega_j\}$$

We can take the cartesian product of the partitions of these partitioning algorithms (or classifiers) to yield the partition

$$\boldsymbol{\mathcal{P}} = \mathcal{P}_1 \times \mathcal{P}_2 \ldots \times \mathcal{P}_d$$

Each cell $\Delta_k, 1 \leq k \leq n_1 \times \ldots \times n_d$ of the partition $\boldsymbol{\mathcal{P}}$ is of the form

$$\Delta_k = (\delta_{1,j(1)}, \ldots, \delta_{i,j(i)}, \ldots, \delta_{d,j(d)})$$

Within each of the cells of the partition $\boldsymbol{\mathcal{P}}$ we can optimize the class selected. The COLMS is one strategy to do so. It is applicable only when all the partitioning algorithms are classifiers and is defined for a partition that is the cross-product of the classifiers' assessments. The OLMS is another strategy to optimize the class selected. It is applicable for general partitioning algorithms and is theoretically expected to give a loss not greater than the COLMS.

5.2 The Classifier-based Overall Loss Minimization Strategy (COLMS)

A technique that makes it possible to improve classifier evaluations by combining assessments from more than one classifier is desirable. Such a technique makes it possible to derive some gain in terms of improvement of classifier performance by applying more than one classifier to a given task. This gain in classifier performance can yield benefits for several real-world applications. A Classifier-based Overall Loss Minimization Strategy (COLMS) is introduced here to combine the evaluations of more than one classifier. The COLMS is identical to the strategy introduced for combining classifiers in [83] except that the coupling procedure in [83] uses only a zero-one loss function. This is equivalent to counting correct or incorrect classifications.

Let $C_m(\boldsymbol{t})$ be the class that classifier C_m assigns to $\boldsymbol{t}$, where $\boldsymbol{t}$ is the data on the case being classified. Each case $\boldsymbol{x} = (\boldsymbol{t}, y)$, where $\boldsymbol{t}$ is the data of the case $\boldsymbol{x}$ and y is the assessment. Let $\alpha_m = C_m(\boldsymbol{t}), m = 1, \ldots, d$. Define $P(\omega_m | \alpha_1, \ldots, \alpha_d)$ as the probability that the actual class is ω_m given that $C_1(\boldsymbol{t}) = \alpha_1, \ldots, C_d(\boldsymbol{t}) = \alpha_d$. $P(\omega_m | \alpha_1, \alpha_2, \ldots, \alpha_d)$ depends on population probabilities and can be expressed as

$$P(\omega_m | \alpha_1, \ldots, \alpha_d) = \frac{P(\alpha_1, \ldots, \alpha_d | \alpha_m) \pi_m}{P(\alpha_1, \ldots, \alpha_d)}$$

where

$$P(\alpha_1, \alpha_2, \ldots, \alpha_d) = \sum_{j=1}^{C} \pi_j P(\alpha_1, \alpha_2, \ldots, \alpha_d | \omega_j)$$

and $P(\alpha_1, \ldots, \alpha_d | \omega_j) = P(C_1(\boldsymbol{t}) = \alpha_1, \ldots, C_d(\boldsymbol{t}) = \alpha_d | \omega_j)$. Further define

$$L^{(m)} = \text{actual loss of } C_m(\boldsymbol{t})$$

$$L^{(m)} = \sum_{i=1}^{C}\sum_{j=1}^{C} P(\omega_i|C_m(\boldsymbol{t}) = \omega_j)P(C_m(\boldsymbol{t}) = \omega_j)$$

$$\begin{aligned} r_\alpha^{(m)} &= E[L^{(m)}|C_m(\boldsymbol{t}) = \alpha] \\ &= \sum_{i=1}^{C} L_{\omega_i,\alpha} P(\omega_i|C_m(\boldsymbol{t}) = \alpha) \end{aligned} \tag{5.1}$$

$r_\alpha^{(m)}$ is the risk of selecting an assessment α reported by classifier C when $C_1(\boldsymbol{t}) = \alpha$. We can extend $r_\alpha^{(m)}$ to $r_{\alpha_1,\ldots,\alpha_d}^{(m)}$ where $r_{\alpha_1,\ldots,\alpha_d}^{(m)}$ is the risk of selecting an assessment α_m reported by classifier $C_m, 1 \leq m \leq d$ when $C_1(\boldsymbol{t}) = \alpha_1, \ldots, C_d(\boldsymbol{t}) = \alpha_d$. $r_{\alpha_1,\ldots,\alpha_d}^{(m)}$ is given by

$$\begin{aligned} r_{\alpha_1,\ldots,\alpha_d}^{(m)} &= E[L^{(m)}|C_1(\boldsymbol{t}) = \alpha_1, \ldots, C_d(\boldsymbol{t}) = \alpha_d] \\ &= \sum_{i=1}^{C} L_{\omega_i,\alpha_m} P(\omega_i|C_1(\boldsymbol{t}) = \alpha_1, \ldots, C_d(\boldsymbol{t}) = \alpha_d) \end{aligned} \tag{5.2}$$

$r_{\alpha_1,\ldots,\alpha_d}^{(m)}$, is the natural extension of $r_\alpha^{(m)}$ to the case where the risk has to be computed on the basis of $d > 1$ classifiers. For $d = 1$, $r_{\alpha_1,\ldots,\alpha_d}^{(m)}$ reduces to $r_\alpha^{(m)}$. The overall expected loss if using the classifier C_m is given by:

$$E[L^{(m)}] = \sum_{\alpha_1=1}^{C} \cdots \sum_{\alpha_d=1}^{C} r_{\alpha_1,\ldots,\alpha_d}^{(m)} P(C_1(\boldsymbol{t}) = \alpha_1, \ldots, C_d(\boldsymbol{t}) = \alpha_d) \tag{5.3}$$

Let $l(\alpha_1, \ldots, \alpha_d)$ be defined by

$$r_{\alpha_1,\ldots,\alpha_d}^{(l(\alpha_1,\ldots,\alpha_d))} = \min_{m=1}^{d} r_{l(\alpha_1,\ldots,\alpha_d)}^{(m)}$$

that is, $C_{l(\alpha_1,\ldots,\alpha_d)}$ is the classifier having least risk given $C_1(\boldsymbol{t}) = \alpha_1, \ldots, C_d(\boldsymbol{t}) = \alpha_d$. Define the COLMS as the classifier that selects category $l(C_1(\boldsymbol{t}), \ldots, C_d(\boldsymbol{t}))$ on the basis of data $\boldsymbol{t}$, that is, the COLMS accepts the selection of the classifier with least risk[1]. Briefly,

$$C_{COLMS}(\boldsymbol{t}) = C_{l(C_1(\boldsymbol{t}),\ldots,C_d(\boldsymbol{t}))}(\boldsymbol{t})$$

Note that the COLMS selects one of the classes selected by the classifiers $C_1, \ldots, C_d$. This is what differentiates it from the OLMS. We have:

$$r_{\alpha_1,\ldots,\alpha_d}^{COLMS} = E[L^{COLMS}|C_1(\boldsymbol{t}) = \alpha_1, \ldots, C_d(\boldsymbol{t}) = \alpha_d]$$

Define:

$$R^{COLMS} = E[L^{COLMS}] = \sum_{\alpha_1=1}^{C} \cdots \sum_{\alpha_d=1}^{C} r_{\alpha_1,\ldots,\alpha_d}^{COLMS} P(\alpha_1, \ldots, \alpha_d)$$

[1] In this thesis, we shall be referring to the classifiers derived from the COLMS strategy and the OLMS strategy as "the COLMS" and "the OLMS." Since we have not specified a tie-breaking strategy when two classifiers have the same least risk we should refer to these classifiers as "a COLMS" and "an OLMS" as a family of classifiers based on the COLMS and OLMS can exist.

Then,

$$r^{COLMS}_{\alpha_1,\ldots,\alpha_d} = r^{l(\alpha_1,\ldots,\alpha_d)}_{\alpha_1,\ldots,\alpha_d}$$

and

$$\begin{aligned} R^{COLMS} &= E[L^{COLMS}] \\ &= \sum_{\alpha_1=1}^{C} \cdots \sum_{\alpha_d=1}^{C} r^{(l(\alpha_1,\ldots,\alpha_d))}_{\alpha_1,\ldots,\alpha_d} \times P(\alpha_1,\ldots,\alpha_d) \end{aligned} \tag{5.4}$$

Theorem: $E[L^{COLMS}] \leq E[L^{(m)}], \forall m = 1,\ldots,d$, i.e., the COLMS has no greater expected loss than any of its constituent classifiers.
Proof:

$$r^{(l(\alpha_1,\ldots,\alpha_d))}_{\alpha_1,\ldots,\alpha_d} = E[L^{COLMS}|C_1(\boldsymbol{t}) = \alpha_1,\ldots,C_d(\boldsymbol{t}) = \alpha_d]$$

We have,

$$r^{COLMS}_{\alpha_1,\ldots,\alpha_d} = r^{l(\alpha_1,\ldots,\alpha_d)}_{\alpha_1,\ldots,\alpha_d}$$

$$C_{COLMS}(\boldsymbol{t}) = C_{l(\alpha_1,\ldots,\alpha_d)} \text{ over } \{\boldsymbol{t}|C_1(\boldsymbol{t}) = \alpha_1,\ldots,C_d(\boldsymbol{t}) = \alpha_d\}$$

and hence conditional on this event

$$E[L^{COLMS}|C_1(\boldsymbol{t}) = \alpha_1,\ldots,C_d(\boldsymbol{t}) = \alpha_d] = r^{l(\alpha_1,\ldots,\alpha_d)}_{\alpha_1,\ldots,\alpha_d}$$

$$\begin{aligned} & E[L^{COLMS}] \qquad (5.5) \\ = & \sum_{\alpha_1=1}^{C} \cdots \sum_{\alpha_d=1}^{C} r^{(l(\alpha_1,\ldots,\alpha_d))}_{\alpha_1,\ldots,\alpha_d} P(C_1(\boldsymbol{t}) = \alpha_1,\ldots,C_d(\boldsymbol{t}) = \alpha_d) \\ = & \sum_{\alpha_1=1}^{C} \cdots \sum_{\alpha_d=1}^{C} \min_{k=1}^{d} r^{(m)}_{\alpha_1,\ldots,\alpha_d} P(\alpha_1,\ldots,\alpha_d) \\ \leq & \sum_{\alpha_1=1}^{C} \cdots \sum_{\alpha_d=1}^{C} r^{(m)}_{\alpha_1,\ldots,\alpha_d} P(\alpha_1,\ldots,\alpha_d), 1 \leq m \leq d \qquad (5.6) \\ = & E[L^{(m)}], 1 \leq m \leq d \end{aligned}$$

Inequality 5.6 follows because the right-hand side of inequality 5.6 is the same as $E[L^{(m)}]$ (Equation 5.3). Hence it is proved that $E[L^{COLMS}] \leq E[L^{(m)}], 1 \leq m \leq d$.
Corollary: The COLMS maximizes the value of $\kappa_{b,w}$ because minimizing L leads to maximizing $\kappa_{b,w}$ when $\kappa_{b,w}$ is evaluated over the same sample set using the same loss function for all the classifiers.

5.3 Adaptations to the COLMS

The COLMS as discussed above chooses an assessment that is made by one of the classifiers. We can modify the COLMS strategy of choosing the assessment made by the classifier with the least risk $r^{(l(\alpha_1,\ldots,\alpha_d))}_{\alpha_1,\ldots,\alpha_d}$ so that it can pick any one of the possible classes $\omega_1,\ldots,\omega_C$, without being restricted to selecting one of the classes $\alpha_1,\ldots,\alpha_d$. More generally, we can

have a selection algorithm that uses the output of partitioning algorithms where the cells δ_j are not necessarily associated in one to one correspondence with the classes used in the classification. We call a strategy that selects a class with minimum-loss condition in each cell δ_j as a Overall Loss Minimization Strategy (OLMS). In addition, we can use domain-specific strategies to supplement the COLMS and the OLMS and enrich their class-selection strategies.

5.3.1 Overall Loss Minimization Strategy (OLMS)

The OLMS is a strategy that assigns all cases in a cell to the same class based on the least-risk principle given below. The COLMS is a special case of the OLMS with the restriction that the COLMS chooses a class only from one of the classes reported by the constituent classifiers. As defined in Section 5.1 let h be an algorithm to partition the data space into cells $\delta_1, \ldots, \delta_g$ and let $h(\boldsymbol{t})$ be the function determined by the partitioning algorithm

$$h(\boldsymbol{t}) = j, \text{ when } \boldsymbol{t} \in \delta_j$$

Define $A_k(j)$, the risk of selecting class ω_k when the partitioning algorithm has reported δ_j as the cell to which $\boldsymbol{t}$ belongs, that is

$$A_k(j) = \sum_{i=1}^{C} L_{ik} P(\omega_i | \boldsymbol{t}' \in \delta_j)$$

Define $l(\boldsymbol{t})$ such that cell $l(\boldsymbol{t})$ gives the minimum risk of all the cells i.e.

$$A_{l(\boldsymbol{t})} = \min_{k=1}^{g} A_k(h(\boldsymbol{t}))$$

The OLMS chooses the assessment $\omega_{l(\boldsymbol{t})}$. $l(\boldsymbol{t})$ depends on $\boldsymbol{t}$ only through $h(\boldsymbol{t})$ and can be written as $f(h(\boldsymbol{t}))$.

Theorem: Given any classifier C that maps the cells $\delta_1, \ldots, \delta_g$ into the classes $\omega_1, \ldots, \omega_C$, then the OLMS has a risk that is not greater than the classifier C i.e.

$$E[L^{OLMS}] \leq E[L^C]$$

The proof is similar to that of the COLMS.

Corollary: Given the partitioning algorithms $P_1, \ldots, P_d$ that map the data into the cells of the partitions $\mathcal{P}_1, \ldots, \mathcal{P}_d$. Let some of the partitioning algorithms be classifiers whose partition $\mathcal{P}$ is given by the classes $\omega_1, \ldots, \omega_C$. Then construct the partition $\boldsymbol{\mathcal{P}}$ given by the cartesian cross product of the partitions $\mathcal{P}_1, \ldots, \mathcal{P}_d$

$$\boldsymbol{\mathcal{P}} = \mathcal{P}_1 \times \mathcal{P}_2 \ldots \mathcal{P}_d$$

The OLMS has no greater loss than any constituent classifier present in the partitioning algorithms, i.e.,

$$E[L^{OLMS}] \leq E[L^{(k)}], 1 \leq k \leq d$$

if $P^{(k)}$, the $k'th$ partitioning algorithm, $1 \leq k \leq d$ is a classifier algorithm.

Theoretically, the OLMS has a value of L no greater than the COLMS because it is not restricted to choosing its class from one of the classes that the classifiers have selected. However, in Section 7.4 we will see that, in an experiment, the estimated COLMS has a lower value of L than the estimated OLMS although the difference is not significant at the 5% significance level. In general, this is not surprising because unless all of the classifiers are poor, it is likely that at least one of them should select the class with the minimum risk. In our case, we have two good classifiers being combined by the COLMS or the OLMS.

5.3.2 Alternative class-selection strategies for class combination

The COLMS and the OLMS use a class-selection strategy based on minimum risk. Besides choosing the class with the minimum risk, other class-selection strategies are possible. One could consider combining such a classifier with the OLMS. However, that will only re-discover the OLMS since they are all defined on the same partition. The OLMS is theoretically the optimal combination strategy if the required classification probabilities were perfectly known, as they would be on a sample of infinitely large size. In a real-world situation, it may not be possible to estimate the classification probabilities accurately. In such a situation, a person may wish to try out the strategies below. One of these strategies may be used as the combination strategy instead of the COLMS or the OLMS. It may be possible to get better results (than if we use only the OLMS, the COLMS, or one of the strategies below) if we use one of the strategies below, and give its assessments as input to an OLMS or COLMS.

Selection when classes have associated numeric values Let the d partitioning algorithms be classifiers. Let there be C classes, $\omega_1, \ldots, \omega_C$. Let the classes have a numeric connotation $\gamma(j), j = 1, \ldots, C$. Let the d classifiers select $\alpha_1, \ldots, \alpha_d$ as their classes for a particular case. If the classes have numeric ranks then we can choose the class whose numeric rank is closest to the mean of the numeric ranks of the classes reported by the two classifiers. If the classes have numeric ranks and there are weights for the classes, we can choose the class whose numeric rank is closest to the weighted average of the numeric ranks of the two classifiers. A weighted-averaging strategy will select the class given by the numeric connotation α_k such that

$$|\gamma(\alpha_k) - (\beta_1\gamma(\alpha_1) + \ldots + \beta_d\gamma(\alpha_d))|$$

is minimized. We can select the weights $\beta_1, \ldots, \beta_d$ manually or by choosing a curve or line-fitting algorithm such as multiple linear regression. Ideally one would choose the β's to minimize the risk. In general, this is a non-trivial problem.

Selection based on ordered classes If we have two classifiers and C classes ordered as $\omega_1 < \omega_2 < \ldots < \omega_C$ then we can choose a class ω_i given by $\alpha_1 \leq \omega_i \leq \alpha_2$, as a compromise solution. The most reasonable choice would probably be the median class between α_1 and α_2. Of course, if C_2 is substantially better than C_1 for some combinations of classes where C_1 chooses α_1 and C_2 chooses α_2 then it is better to select α_2 (as the COLMS would have done) instead of seeking a compromise solution.

Selecting by a classification algorithm An algorithm, such as back-propagation, can be used to combine the classifiers $C_1, \ldots, C_d$. In this case, the algorithm will be

trained on the assessments of the classifiers and the correct class. The task of the algorithm will be to predict the correct class from the assessments of the classifiers.

Of course the problem with the above strategies is that they ignore the fact that one classifier is better than the other (which the OLMS considers). It should be noted that the OLMS is theoretically the best combination strategy for the constituent classifiers. The alternative strategies are useful when they reflect some domain-specific considerations that cannot be captured by the OLMS because the sample size is small.

5.4 An Expert-defined Domain-Specific combination Strategy (EDSS) for the coronary angiography application

A domain expert may supply a strategy based on common sense to combine two or more classifiers. For the coronary angiography application the domain-specific combination strategy for use with a human user, ESCA, and two evaluation algorithms has been obtained from Dr. David Hunter, a panel member who evaluates the coronary angiograms for POSCH and who participated in the development of ESCA. The overall approach is to select the assessment of the most-trusted classifier as the initial classification and possibly modify it depending on how badly the other classifiers disagree with it. Dr. Hunter considered the most-trusted classifier to be a "user." We are modifying his approach (with his approval) to the case where the most trusted classifier is a computerized one.

Dr. Hunter's strategy was designed for a user who also acts as a classifier, and three other classifiers. Since we do not have the user evaluating the cases, we need four computerized classifiers. We choose the four available classifiers viz. the expert system ESCA, multiple linear regression (MLR), back-propagation (BP), and C4.

The EDSS may result in the user changing his assessment when the EDSS asks the user to re-evaluate the case[2]. The EDSS asks the user to re-evaluate a case when it detects too much discrepancy in the assessments of the classifiers. If the user refuses to change his assessment then the EDSS rejects the case as unclassifiable because it has noted too much discrepancy in the assessment of the case by the individual classifiers. If we do not have a human user capable of changing his assessment as the most trusted classifier than "Re-evaluate" is synonymous with "reject."

The expert system and algorithms used in this thesis cannot re-evaluate the case. The experts' assessments that are available cannot be changed as the data has already been collected and it is not feasible to have the panel re-evaluate the cases. For this reason, we will treat re-evaluate as reject the assessment (with Dr. Hunter's approval), since there is no opportunity to change the panel's assessment.

ESCA has a one-dimensional evaluation scale from -3 to +3 viz. (-3, -2, -1, -0, +0, +1, +2, +3). Equivalently the scale can be coded from 1 to 8 viz. (1, 2, 3, 4, 5, 6, 7, 8) [Table 3.1]. In the description below we shall use the scale (1, 2, 3, 4, 5, 6, 7, 8) unless specified otherwise. Referring back to the original values (-3, -2, -1, -0, +0, +1, +2, +3) we define *sign(as) = -1* for *as = 1, 2, 3, 4* and *sign(as) = +1* for *as = 5, 6, 7, 8*. Let *as(M)* be the assessment by any classifier method where the method can be an expert, a user, an expert system, or a algorithmic classifier. *Within1(as(A), as(B))* is true if and

[2] The other classifiers are computer-based and were not capable of changing their assessments.

only if the assessment by classifier A and the assessment by classifier B occupy the same or adjacent categories on ESCA's eight-point assessment scale viz. (1, 2, 3, 4, 5, 6, 7, 8). *Between(as(A), as(user),as(B))* is true when $as(A) \leq as(user) \leq as(B)$. *sign-different(as1, as2)* is true if ($as1 \in \{1,2,3\}$ and $as2 \in \{6,7,8\}$) or if ($as2 \in \{1,2,3\}$ and $as1 \in \{6,7,8\}$). Here, +0 and -0 can take either + or - sign as needed to minimize the change in signs.

In the description "user" refers to the user or the most-trusted classifier (which can be an algorithm or an expert system). Classifiers A, B, and C are trusted in decreasing order.

Since we do not have a user or operator evaluating the cases, and we have ordered ESCA, MLR, BP, and C4 as being trusted in decreasing order we use the expert system ESCA's assessments instead of the user's assessments, MLR's assessments as Classifier A's assessments, BP's assessments as Classifier B's assessments, and C4's assessments as Classifier C's assessments [3].

The rules for the Expert-Defined Domain-Specific Strategy (EDSS) are:

start:

if *(as(A) = as(B))* and *within1(as(user),as(A)))*

then *as(EDSS) = as(A)* /* Select as(A). Since the agreement of A and B is more convincing than the user, believe A and B, changing the user's assessment if necessary */
goto done

end if

if *((as(A) ≠ as(B))* and
(((as(A) = as(user)) and *within1(as(B), as(user)))* or
((as(B) = as(user)) and *within1(as(A), as(user)))))*
/* one of A or B agrees with the user, and the other is within one category of the user */

then *as(EDSS) = as(user)* /* select the user's assessment. */
goto done

end if

if *(within1(as(user), as(A))* and *within1(as(user), as(B))* and
between(as(A), as(user),as(B)))

then /* Select the user's assessment because user is the 'average' */ *as(EDSS) = as(user)*
goto done

end if

if $(|as(A) - as(user)| = 2$ or $|as(B) - as(user)| = 2)$
/* One or both of A and B's assessments is two categories different from the user */

then

[3] This is done with Dr. Hunter's approval. But note that it affects the strategy since the expert system cannot re-assess a case when the EDSS asks it to re-evaluate it.

if *(sign-different(as(A), as(user))* or *sign-different(as(B), as(user)))*
/* the sign of the assessment of either A or B is different from that of the user (+0 and -0 can take either + or - sign as needed to minimize the change in signs) */

then Request a re-evaluation from the user.

if The user cannot change his assessment

then /* Reject this case */
as(EDSS) = 'reject'.
Goto done.

else
goto start.

end if

end if

end if

if $(as(A) \leq as(user) \leq as(B))$ or $(as(B) \leq as(user) \leq as(A))$ and one of A or B's assessments is 2 or 3 categories different from the user's assessment (let this method be α and the other method be β) and $|as(\beta) - as(user)| \leq 1$

then /* Consult C */

if *within1(as(C), as(user))*

then *as(EDSS) = as(user)* /* accept the user's assessment. */
goto done

end if

if *not (within1(as(C), as(user)))*

then Request a re-evaluation from the user.

if The user cannot change his assessment

then /* Reject this case */
as(EDSS) = 'reject'.
Goto done.

else
goto start.

end if

end if

end if

if One of A or B is 2 or 3 categories different from the user and has the same sign (call this method α) and if the other method (β) is one category away from the user and $(as(\alpha) < as(user)$ and $as(\beta) < as(user))$ or $(as(\alpha) > as(user)$ and $as(\beta) > as(user))$

then /* Check C */

if C is the same as either A or B

then as(EDSS) is that given by the majority vote of A, B, and C.
goto done

end if

if C does not agree with A or B

then Request a re-evaluation from the user.

if The user cannot change his assessment

then /* Reject this case */
as(EDSS) = 'reject'.
Goto done.

else
goto start.

end if

end if

end if

/* all other cases: re-evaluate */
Request a re-evaluation from the user.

if The user cannot change his assessment

then *as(EDSS) = 'reject'.*
Goto done

end if

/* Default condition to cover all other possible combinations of A, B, C, and the user's evaluations */
as(EDSS) = 'reject'
done : halt

5.5 Rejection and rejection loss

A classifier is said to reject a case if it refuses to give an assessment to a case. A strategy that combines classifiers is a classifier, albeit a hybrid classifier. Sometimes the expert will assign a rejection loss when a case is rejected or define how to obtain the loss for rejecting the case. If the expert has not given information on how to compute the rejection loss, we use the procedure below.

The procedure assumes that the classifier will always give a best guess, that is, a selection that would be made if the classifier were forced to do so. This best guess forms the basis for computing the rejection. The procedure is defined so that the rejection loss is high if the loss of a best guess as compared to the expert or gold standard is low, and vice versa.

Let $\alpha_1 = \omega_j$ be a best guess by the classifier C_i, and $\alpha_2 = \omega_i$ be the true classification (usually the assessment of the expert or gold standard). Then the rejection loss R_{ij} is defined as:

$$R_{ij} = 1 - L_{ij}$$

where L_{ij} is the loss if the classifier had chosen assessment ω_j and when the true class is ω_i. Using this strategy for computing the rejection loss, if a best assessment of the method was in fact good ($L_{ij} \approx 0$) then R_{ij} will be high and vice versa. In this strategy we do not consider the cost of evaluating a case. This is to avoid complications that can result because the cost involved in evaluating a case is different from the loss of an incorrect evaluation. For instance, cost may be measured in dollars while loss may be measured in terms of misdiagnosed diseases. This relation between cost and loss has to be explicitly introduced for a specific domain.

If we have C classes, then we introduce C additional *rejection classes*, $\omega_{C+1}, \ldots, \omega_{2C}$. If any classifier's best guess is ω_i but it decides to reject it, then the classification is reported as $\omega_{(C+i)}$. If we require that the true classification is always known, (or equivalently, the gold standard never rejects any sample as unclassifiable) we have,

$$p_{i|j} = 0, \ \forall i = C+1, \ldots, 2C, j = 1, \ldots, 2C$$

In such a situation, we can define an extended loss function as below defined on C additional classes. Let N_{ij} be the extended loss if the classifier reports class ω_j when the actual class is ω_i. Then

$$\begin{aligned} N_{ij} &= L_{ij}, \forall i = 1, \ldots, C, j = 1, \ldots, C \\ &= R_{ij} = 1 - L_{i,j-C}, \forall i = 1, \ldots, C, j = C+1, \ldots, 2C \\ &= 0, \forall i = C+1, \ldots, 2C, j = 1, \ldots, 2C \end{aligned}$$

In the last case above, the actual value of N_{ij} assigned is irrelevant because we have assumed that the gold standard never rejects any sample as unclassifiable. The value of L or $\kappa_{b,w}$ computed for a classifier that does not reject samples will be the same whether the loss function L or the extended loss function N is used. The extended loss function N can be used to compare classifiers that reject samples with classifiers that do not reject samples, if we assume that classifiers that reject samples are a generalized version of classifiers that do not reject samples. This assumption legitimizes the comparison when the values of R_{ij} are not expert-defined.

5.6 Summary

This chapter introduced the COLMS, a domain-independent strategy for combining classifiers. A generalization of the COLMS, the OLMS, was can use the output of partitioning algorithms that partition the data into cells other than the given classes. Alternative class-selection strategies for classifier combination are described. The COLMS, the OLMS, and the alternative class-selection strategies are useful tools when we wish to combine classifiers or partitioning algorithms from fields as diverse as statistics, neural networks, and expert systems.

The OLMS is important as it enables partitioning (clustering) algorithms to be used where the clusters need not be the output classes. This is useful for using techniques from

fields such as neural networks. The use of the OLMS was demonstrated in Chapter 7 and the results compared with the COLMS. It is shown there that $\hat{L}$ is higher for the OLMS than for the COLMS although the results are not significant at the 5% significance level.

An expert-defined strategy for combining classifiers (EDSS) was introduced. As we shall see in Chapter 7, the EDSS performs poorly, while the COLMS performs well for the AEPOSCH application. One conclusion from this is that it is hard to develop an ad hoc strategy to combine classifiers because a domain expert is not necessarily an expert at combining classifiers.

Chapter 6

Using perturbation-testing for a classifier

As we have seen in previous chapters [Chapter 2 and Chapter 4] the value of a performance criterion for a classifier can be estimated along with the standard error of the estimate. The classifier's acceptability can be determined by using the estimated value of this performance criterion and hypothesis testing. In addition, an expert can test the classifier using an expert-defined test (EDT) that tests the classifier on domain-specific tests. The use of an EDT can increase the expert's confidence in the classifier and may be important in accepting an algorithmic classifier.

Perturbation-testing is one example of an EDT. In perturbation-testing, the data values of a sample are perturbed according to a scheme defined by an expert. This results in a new perturbed sample. The perturbation method may use the classifier's assessment of a case as well as the data for the case in determining a suitable perturbation. The perturbed sample is evaluated by the classifier and the new assessment is compared with the previous assessment.

Three schemes for doing a perturbation are possible if one does not consider the gold standard's classification when doing the perturbation and considers the classifiers individually. They are:

Rigid scheme Let $\boldsymbol{t}$ be the unperturbed data for a case $\boldsymbol{x}$, g be the perturbation function, and $\tilde{\boldsymbol{t}}$ be the perturbed data of the case, with

$$\tilde{\boldsymbol{t}} = g(\boldsymbol{t})$$

In other words, for each classifier, the perturbation strategy applies the same perturbation to the data of the case. This is a "rigid" scheme, because it does not take into account differences among the classifiers.

Flexible scheme Let $\boldsymbol{t}$ be the unperturbed data of a case $\boldsymbol{x}$, $C(\boldsymbol{t})$ be the classification by the classifier C, and $\tilde{\boldsymbol{t}}_C$ be the perturbed data of the case. Then,

$$\tilde{\boldsymbol{t}}_C = g(C(\boldsymbol{t}), \boldsymbol{t})$$

In other words, the perturbation strategy considers the classification $C(\boldsymbol{t})$ of $\boldsymbol{t}$ when perturbing data for input to classifier C. If different classifiers make the same classification, the same perturbation is applied. It is not an arbitrary decision to apply

a different data-modification strategy to the same case for different classifiers. This is a "flexible" scheme, because it takes into account differences among the classifiers. This is not an "arbitrary" scheme because it requires that the criteria for applying different data-modification techniques to a case be objective and be the same for all classifiers.

Arbitrary scheme Let $\boldsymbol{t}$ be the unperturbed data of a case $\boldsymbol{x}$, and $\tilde{\boldsymbol{t}}_m$ be the perturbed data of the case to be classified by C_m. Then,

$$\tilde{\boldsymbol{t}}_m = g_m(\boldsymbol{t})$$

where g_m is a data-modification strategy specific to classifier C_m. Each case is evaluated by the given classifier. The choice of the data-modification strategy can be arbitrary depending on the classifier involved without any objective reason given for choosing a particular data-modification strategy. The choice of data-modification strategy to apply can depend on the identity of the classifier and need not rely on feature values or attributes. This is an "arbitrary" scheme because no reason need be given to perturb cases for one classifier differently from that of another classifier.

Comparisons based on perturbations using the rigid and flexible schemes above are valid. There can be questions as to the validity of comparisons based on perturbations using the arbitrary scheme. In our work, we are using the flexible scheme above. For each perturbation defined, the expert expects the classifier to respond by changing its assessment (if necessary) from its original assessment on unperturbed data. Each perturbation method is expected to result in a change in assessment by a classifier in the same way that the expert has predicted. The expert codifies this expectation in a table of *points* rewarding/penalizing the classifier for the "well-behaved" manner it changed its assessment in response to the perturbation in the data. These points are assigned by the expert for each combination of the original assessment and the perturbed assessment to obtain a more rigorous *perturbation evaluation* of a classifier.

6.1 Perturbation-testing in the AEPOSCH application

Four possible perturbations are defined for the AEPOSCH application. They are Perturb Slightly Low (PSL), Perturb Slightly High (PSH), Perturb Low (PL), and Perturb High (PH). It is usually possible to visualize certain segments clearly, viz., proximal-lad, mid-lad, proximal-left-circumflex, mid-left-circumflex, proximal-right-coronary, and mid-right-coronary. All perturbation methods make changes to the stenoses only of these arterial segments, trying to change the stenoses of each segment in the order listed above (proximal-lad first) until the required number of changes are made. The exact procedure for making changes in individual arterial segments is given in Section 6.1.1. In all the perturbation procedures below, if the stenosis cannot be adjusted in the required number of arteries, the perturbation is not carried out. Besides this case, there are times when a perturbation is not possible. For instance, where the assessment of a case is "-3" ("+3") it cannot be perturbed lower (higher). In other cases, the artery may have a high value of change (for example, 80%), and the classifier's assessment is "worse". In this case the artery cannot be perturbed using the steps described for PL in Section 6.1.4.

The basic purpose for the perturbation-testing is not to evaluate the accuracy of the classifier at predicting the assessment/class (this is done by using L and $\kappa_{b,w}$). Instead, it is to test whether the classifier responds appropriately to a perturbation in the data. This is done by comparing the new assessment of the classifier for the perturbed case with the assessment of the classifier for the original case. If the magnitude and direction of change in the assessment by the classifier satisfies the expert's expectation then we say that the classifier has passed the perturbation test. Another way of describing that the classifier has passed the perturbation test is to say that the "classifier is consistent with itself" because the assessment of the perturbed case is as expected (by the expert) given the perturbation and the assessment by the classifier of the original case.

The perturbation points in Tables 6.1, 6.3, 6.4, and 6.5 formalize the expert's expectation. Each value $v_{i,j}$ for a particular perturbation represents the expert-assigned score when a classifier responds by changing assessment i to assessment j for a particular case for the given perturbation strategy. The maximum value of 1 corresponds to a perfect match with expectations. At times, even a non-expert might argue that the perturbation value should not be as given by the expert. An example is given in the next section.

6.1.1 Procedure for making changes in arteries

The changes are made by applying the following steps in turn until one step is successful. If the required number of changes cannot be made by applying the following steps, then the required perturbation is not made.

Old-a and old-b are the stenoses in the unperturbed data for Film A and Film B of the artery. New-a and new-b are the stenoses in the perturbed data for Film A and Film B of the artery. It is tested that new-a and new-b will be between 0 and 100. A change refers to the change in stenosis in an artery

change = (stenosis in Film B) - (stenosis in Film A)

1. new-a = old-a
   new-b = new-a + change-stipulated

2. new-b = old-b
   new-a = new-b - change-stipulated

3. new-b = max(old-b, 90)
   new-a = new-b - change-stipulated

4. new-b = min(old-b, 10)
   new-a = new-b + change-stipulated

5. new-a = min(old-a, 10)
   new-b = new-a + change-stipulated

6. new-a = max(old-a, 90)
   new-b = new-a - change-stipulated

6.1.2 Procedure to Perturb Slightly Low (PSL)

PSL accepts the stenoses of a case and a classifier's assessments of a case and perturbs the stenoses so that a classifier exhibiting the correct perturbation behavior is expected to report an assessment that is the same or one category lower (worse) than before. The evaluation matrix for PSL is given in Table 6.1.

It might be argued that $v_{-3,-3}$, the perturbation score for keeping an assessment of -3 as -3, should be 1 and not .8 since it is not possible to perturb lower an assessment of -3. However, 0.8 is the value provided by the expert and that is what we will use.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3
-3	.8	-1	-1.5	-2	-3	-5	-8	-8
-2	1	.8	-1	-1.5	-2	-3	-5	-8
-1	.5	1	.8	-1	-1.5	-2	-3	-5
-0	0	.5	1	.8	-1	-1.5	-2	-3
+0	-.5	0	.5	1	.8	-1	-1.5	-2
+1	-1	-.5	0	.5	1	.8	-1	-1.5
+2	-2	-1	-.5	0	.5	1	.8	-1
+3	-3	-2	-1	-.5	0	.5	1	.8

Table 6.1: Points v_{ij} for Perturb Slightly Low in the AEPOSCH application.
The original assessments are in rows and the assessments of perturbed cases are in columns.

Three arterial segments are perturbed in PSL. The perturbation takes place by first perturbing $X, (X \leq 3)$ arterial segments with changes from 20-60% by increasing the changes by 20% in each arterial segment. Then, for $(3 - X)$ other arterial segments the changes are increased by 20% each. If there is no change in stenosis in an arterial segment under consideration a change of 20% is introduced in the segment.

Table 6.2 gives the stenoses in arterial segments before and after perturbing using PSL for a hypothetical example. Only the values of the six easily visualizable arterial segments are shown.

Art-segment	Unpert. A	Unpert. B	Unp. Ch.	Pert. A	Pert. B	Per. Ch.
prox-lad	30	70	40	30	90	60
mid-lad	70	90	20	50	90	40
prox-left-cir	10	10	0	10	30	20
mid-left-circ	35	45	10	35	45	10
prox-rt-cor	25	20	-5	25	20	-5
mid-rt-cor	0	0	0	0	0	0

Table 6.2: Stenoses in arterial segments before and after perturbing using PSL.

6.1.3 Procedure to Perturb Slightly High (PSH)

PSH accepts the stenoses of a case and perturbs the stenoses so that a classifier consistent with itself is expected to report an assessment that is the same or one category higher (better) than before. The evaluation matrix for PSH is given in Table 6.3.

Three arterial segments are perturbed in PSH. The perturbation takes place by first perturbing $X, (X \leq 3)$ arterial segments with changes from 40-80% by reducing the changes by 20% each. Then, for $(3 - X)$ other arterial segments, reduce the changes by 20% each. If there is no change in stenosis in an arterial segment under consideration a change of -20% is introduced in the segment.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3
-3	.8	1	.5	0	-.5	-1	-2	-3
-2	-1	.8	1	.5	0	-.5	-1	-2
-1	-1.5	-1	.8	1	.5	0	-.5	-1
-0	-2	-1.5	-1	.8	1	.5	0	-.5
+0	-3	-2	-1.5	-1	.8	1	.5	0
+1	-5	-3	-2	-1.5	-1	.8	1	0.5
+2	-8	-5	-3	-2	-1.5	-1	.8	1
+3	-8	-8	-5	-3	-2	-1.5	-1	.8

Table 6.3: Points v_{ij} for Perturb Slightly High in the AEPOSCH application.
The original assessments are in rows and the assessments of perturbed cases are in columns.

It might be argued that $v_{+3,+3}$, the perturbation score for keeping an assessment of +3 as +3, should be 1 and not .8 since it is not possible to perturb higher an assessment of +3. However, 0.8 is the value provided by the expert and that is what we will use.

6.1.4 Procedure to Perturb Low (PL)

PL accepts the stenoses of a case and a classifier's assessments of a case and perturbs the stenoses so that a classifier consistent with itself will report an assessment that is one or two categories lower (worse) than before. The evaluation matrix for PL is given in Table 6.4.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3
-3	X	X	X	X	X	X	X	X
-2	1	-1	-2	-8	-10	-14	-16	-16
-1	1	1	-1	-6	-8	-12	-16	-16
-0	.5	1	1	-2	-3	-8	-12	-16
+0	.5	1	1	0	-2	-8	-12	-16
+1	-2	.5	1	.5	.5	-2	-8	-12
+2	.8	-3	.5	1	1	.5	-1	-8
+3	-10	-4	-1	0.5	0.5	1	0.5	-1

Table 6.4: Points v_{ij} for Perturb Low in the AEPOSCH application.
The original assessments are in rows and the assessments of perturbed cases are in columns.
"X" denotes that the perturbation is not applied because it is not possible.

Perturb Low works as follows:

1. If the assessment is "much worse" take no action.

2. If the assessment is "worse" or "slightly worse," to X arteries, $(X \leq 4)$ with changes from 20% to 50% increase the changes by 40% each. To $(4 - X)$ arteries increase changes by 40% each or introduce changes of 40% each.

3. If the assessment is "imperceptibly worse" or "imperceptibly better," to X arteries, $(X \leq 4)$ with changes of 20% increase the changes by 40% each. To $(4 - X)$ arteries increase changes by 40% each or introduce changes of 40% each.

4. If the assessment is "slightly better" eliminate all changes in the arteries.

   new-a = new-b = (old-a + old-b) / 2

   To two arteries introduce changes of 30% each.

5. If the assessment is "better" or "much better," to X arteries, $(X \leq 4)$ with changes from -40% to -90% increase the changes by 40% each. To $(4 - X)$ arteries increase changes by 40% each or introduce changes of 40% each.

6.1.5 Procedure to Perturb High (PH)

PH accepts the stenoses of a case and a classifier's assessments of a case and perturbs the stenoses so that the assessment is higher (better) than before. The evaluation matrix for PH is given in Table 6.5.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3
-3	-1	.5	1	.5	.5	-1	-4	-10
-2	-8	-1	.5	1	1	.5	-3	-8
-1	-12	-8	-2	.5	.5	1	.5	-2
-0	-16	-12	-8	-2	0	1	1	.5
+0	-16	-12	-8	-3	-2	1	1	.5
+1	-16	-16	-12	-8	-6	-1	1	1
+2	-16	-16	-14	-10	-8	-2	-1	1
+3	X	X	X	X	X	X	X	X

Table 6.5: Points v_{ij} for Perturb High in the AEPOSCH application.
The original assessments are in rows and the assessments of perturbed cases are in columns. "X" denotes that the perturbation is not applied because it is not possible.

Perturb High works as follows:

1. If the assessment is "worse" or "much worse," to X arteries, $(X \leq 4)$ with changes from 40% to 90% reduce the changes by 40% each. To $(4 - X)$ arteries reduce changes by 40% each or introduce changes of -40% each.

2. If the assessment is "slightly worse" eliminate all changes in the arteries.

   new-a = new-b = (old-a + old-b) / 2

   To two arteries introduce changes of -30% each.

3. If the assessment is "imperceptibly worse" or "imperceptibly better," to X arteries, $(X \leq 4)$ with changes of -20% reduce the changes by 40% each. To $(4 - X)$ arteries reduce changes by 40% each or introduce changes of -40% each.

4. If the assessment is "better" or "slightly better," to X arteries, $(X \leq 4)$ with changes from -20% to -50% reduce the changes by 40% each. To $(4 - X)$ arteries reduce changes by 40% each or introduce changes of -40% each.

5. If the assessment is "much better" take no action.

6.2 Testing for generalization

Generalization testing refers to testing a classifier's ability to classify cases from a population different from the population from which the training cases were drawn. One way the populations might differ is in the underlying proportions. In this case the estimates of L and $\kappa_{b,w}$ can be computed by using the conditional error probabilities estimated from the original population and using the marginal probabilities appropriate for the new population. Another way that a population might differ is that the underlying characteristics may differ. For instance, if the task is to classify a person as male or female, the underlying characteristics differ for Caucasian versus Asian populations (weight, height, etc.). In such a situation, an expert may define a strategy to generate a modified set of cases from the available set by modifying the characteristics. These cases will differ from the available cases in that the mix of cases will more closely resemble what the expert expects to encounter for the new population. A classifier can then be tested on this new set to see whether it performs acceptably on the population with the different characteristics.

When testing a classifier for generalization we are testing it on a population that is different from the original population. For this reason it does not follow that if a classifier A does better than classifier B for the original population it will do better than B during testing on another population for generalization.

6.3 Summary

Sometimes a domain expert (or the knowledge engineer) may wish to examine the system more closely even if the classifier has acceptable values of $\hat{L}$ and $\hat{\kappa}_{b,w}$. An expert-defined test (EDT) can be used to inspect the algorithm in more detail to check the behavior of the classifier in terms of how well it satisfies certain tests specified by the domain expert. One such test is a Perturbation Test.

This chapter described four perturbation tests (PSL, PSH, PL, and PH) for the AE-POSCH application. Evaluation of perturbation scores for PSL, PSH, PL, PH and their sum P(SUM) for different classifiers is done in Chapter 7. The experts' comments on the results are given there. The perturbation methods did characterize the classifiers further and

gave more confidence to the experts about multiple linear regression. After reviewing the values of $\hat{L}, SE(L), \hat{\kappa}_{b,w}$, $SE(\kappa_{b,w})$, and the results of the perturbation-testing, the experts concluded that multiple linear regression could be used for the AEPOSCH application.

A brief discussion on how to generalize classifiers from one population to another is given. The new population may have different marginals or be characterized by new values for the feature vectors.

Chapter 7

Experiments over the AEPOSCH database

This chapter gives the results of experiments to estimate the values of performance criteria and their standard errors for the classifiers ESCA, multiple linear regression (MLR), back-propagation (BP), and C4. The data used comes from a subset of atherosclerosis evaluation data in the POSCH database. The atherosclerosis evaluation application is known as the AEPOSCH application (Chapter 3) and this data is known as the AEPOSCH data. The results of perturbation-testing of these classifiers on the AEPOSCH data are given.

Bonferronization[1] of p-values for comparison tests are done to account for the effect of multiple comparisons. When Bonferronizing the results of perturbation-testing we consider that while there are five perturbation methods, viz., PSH, PSL, PL, PH, and their sum $P(SUM)$, we are focusing in every comparison between classifiers tested on only one perturbation technique. Therefore, the Bonferronization takes into account that four classifiers, viz., ESCA, MLR, BP, and C4 are compared and does not include the effect of having multiple perturbation tests.

7.1 Atherosclerosis data in POSCH

The process of recording the data for each case in the POSCH atherosclerosis database is described in Chapter 3. All the cases are evaluated once as part of a *regular evaluation*. In addition, some cases are evaluated a second time as part of a *quality-control evaluation* for reproducibility purposes. The experts on the subpanels are not the same for evaluating all the quality-control cases, but in fact the quality-control cases are evaluated by recirculating them among the regular cases. For this reason, the same expert who evaluated a case for a regular evaluation, may evaluate another case for a quality-control evaluation as though it were a regular evaluation (and without knowing that it is a quality-control evaluation). Therefore it is not meaningful to give a cross-classification matrix in the form of *assessment during regular study* vs. *assessment during the quality-control study* because one cannot distinguish between the regular and the quality-control evaluations when doing the evaluations. An upper-triangular matrix is used to report the cross-classification results [Tables

[1] See page 65 for an explanation of the Bonferroni technique.

7.2 and 7.3]. We will be referring to the subpanel constituted of two experts as an expert in this Chapter. $Count(i,j), i \leq j$, is the number of cases where one classifier said ω_i and the other classifier said ω_j. L for the quality-control data in the quality-control dataset is evaluated using multinomial results (Equations 2.1). $SE(L)$ is difficult to compute because the experts constituting the subpanels and the procedure used to evaluate the cases are the same for the regular evaluations and the quality-control evaluations. $\kappa_{b,w}$ (and hence $SE(\kappa_{b,w})$) cannot be computed for the same reason. Since the experts constituting the subpanels and the procedure used to evaluate the cases are the same for the regular evaluations and the quality-control evaluations, the two subpanels can both be considered as gold standards and it is not possible to discriminate between the two. $\kappa_{b,w}$ is not a meaningful measure for measuring agreement adjusted for chance between independent raters who can be considered as gold standards. This is another reason why $\kappa_{b,w}$ should not be computed for the quality-control evaluations. κ_c (Chapter 2) is more appropriate to measure agreement adjusted due to chance for the quality-control evaluations because κ_c can be used to measure agreement adjusted for chance agreement between two independent raters who are considered as gold standards.

There are 696 cases in the AEPOSCH application dataset where one visit is at baseline and the other visit is three years after baseline. Of these, there are 44 quality-control cases that have been evaluated twice by POSCH subpanels for quality-control purposes. A description of the evaluation process in POSCH is given in Chapter 3. These quality-control cases constitute the AEPOSCH-B3-QA dataset or the B3-QA dataset. This leaves 652 cases that have been evaluated only once. Since we would like to use 90% or more of the cases for training and we need an integral multiple of cases for the SB method and the SGJK method we choose the first 630 cases (of the remaining 652 cases) and do not consider the last 22 cases in our evaluation. Choosing 630 cases lets us divide our sample into 21×30 cases. This corresponds to using 97% of the cases for training as 609 cases are used for training in each replication of the classifier. Since the cases were not ordered in any particular sequence, we consider this sample a random sample. These 630 cases will be referred to as the AEPOSCH-B3 dataset or B3 dataset. The upper-triangular cross-classification table of the experts evaluating the B3-QA dataset is given in Table 7.2. Table 7.1 gives the regression coefficients for multiple linear regression when the entire B3 dataset is used for training.

In addition to the cases in the B3 and the B3-QA datasets there are 1170 cases where at least one of the visits is at five, seven, or ten years after baseline (the second visit can be at any time). There can be cases that have been evaluated two or more times (for example, one at five years after baseline, and the other at, say, ten years after baseline). Of these, 65 are quality-control cases that have been evaluated by two subpanels. These 65 cases constitute the AEPOSCH-BR-QA dataset (or BR-QA dataset). The remaining 1105 cases constitute the AEPOSCH-BR dataset (or BR dataset). The upper-triangular cross-classification table of the experts evaluating the BR-QA dataset is given in Table 7.3.

In the development process of ESCA (Section 3.1) 8 cases were initially presented to the expert in the first knowledge engineering session and the majority of the rules were obtained from the expert in that session. Subsequently, the results of running the prototype system for 200 cases were presented to the expert, and some additional rules obtained. The identity of these over 200 cases are not known. Since over 200 cases in the B3 dataset have been encountered by the expert before, it can be argued that the values of the performance

Artery segment	regression-coef.	t-value
Constant term	4.49529	0.08350
Right-ostium (Film A)	-0.00089	0.01004
(Film B)	-0.00242	0.01065
Prox-rt-coronary (Film A)	0.01871	0.00215
(Film B)	-0.01976	0.00214
mid-rt-coronary (Film A)	0.02236	0.00232
(Film B)	-0.02204	0.00236
distal-rt-coronary (Film A)	0.01876	0.00286
B	-0.01674	0.00287
prox-posterior-desc (Film A)	-0.01019	0.00574
B	0.01035	0.00583
distal-posterior-desc (Film A)	-0.03894	0.02430
(Film B)	0.03825	0.02659
left-ostium (Film A)	-0.01932	0.02663
(Film B)	aliased	
left-main-coronary (Film A)	0.01466	0.00688
(Film B)	-0.01243	0.00634
proximal-lad (Film A)	0.01941	0.00274
(Film B)	-0.01884	0.00269
mid-lad (Film A)	0.02426	0.00272
(Film B)	-0.02286	0.00269
distal-lad (Film A)	0.01405	0.00447
(Film B)	-0.01549	0.00431

Table 7.1: Multiple linear regression coefficients fitted by least squares when training on entire B3 dataset
$R^2 = 0.699323, \hat{\sigma} = 0.789211, N = 630$, Degrees of freedom = 584.
Left-ostium (Film B) is aliased to left-ostium (Film A). When regressing left-ostium (Film B) versus the stenoses in the remaining 45 arterial segments, the regression coefficient with left-ostium (Film A) = 1 and $R^2 = 1$.
Table continued on next page.

Artery segment	regression-coef.	t-value
prox-left-circumflex (Film A)	0.01533	0.00372
(Film B)	-0.01618	0.00368
mid-left-circumflex (Film A)	0.01266	0.00307
(Film B)	-0.01379	0.00296
distal-left-circumflex (Film A)	0.02019	0.00627
(Film B)	-0.01943	0.00601
first-diagonal-lad (Film A)	0.01489	0.00372
(Film B)	-0.01548	0.00360
second-diagonal-lad (Film A)	0.01354	0.00438
(Film B)	-0.01403	0.00431
third-diagonal-lad (Film A)	0.00249	0.00726
(Film B)	-0.00440	0.00834
first-obtuse-marginal (Film A)	0.01850	0.00341
(Film B)	-0.01695	0.00333
second-obtuse-marginal (Film A)	0.00603	0.00427
(Film B)	-0.00775	0.00406
third-obtuse-marginal (Film A)	0.01987	0.00714
(Film B)	-0.02006	0.00680
other-segment-a (Film A)	0.01235	0.00433
(Film B)	-0.01153	0.00445
other-segment-b (Film A)	0.01493	0.00446
(Film B)	-0.01631	0.00444
other-segment-c (Film A)	-0.00655	0.00525
(Film B)	0.00704	0.00526

Table 7.1 (Contd.): Multiple linear regression coefficients fitted by least squares when training on entire B3 dataset (Contd.).

expert-1 $\downarrow$\expert-2. $\rightarrow$	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	0	0	0	0	0	0	0	0	0
-2		1	1	2	0	0	0	0	4
-1			0	6	0	0	0	0	6
-0				4	10	0	0	0	14
+0					4	7	1	0	12
+1						3	4	0	7
+2							0	1	1
+3								0	0
Sum		1	1	12	14	10	5	1	44

Table 7.2: Upper-triangular cross-classification table of B3-QA dataset.

expert-1 ↓\expert-2. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	1	0	0	0	0	0	0	0	1
-2		2	3	2	0	0	0	0	7
-1			9	13	0	0	0	0	22
-0				4	6	1	0	0	11
+0					5	4	0	0	9
+1						2	7	0	9
+2							1	3	4
+3								2	2
Sum	1	2	12	19	11	7	8	5	65

Table 7.3: Upper-triangular cross-classification table of BR-QA dataset.

criteria obtained for the B3 dataset are not solely due to unseen cases, but rather, a limited form of training for ESCA has taken place. For this reason, strictly speaking, only the values of performance criteria of ESCA over the BR dataset should be considered as the values expected over unseen cases that have one visit at baseline and the other at five, seven, or ten years after baseline (Section 3.1).

7.2 Estimation of L and $\kappa_{b,w}$ over the atherosclerosis data

$\hat{L}$ and $\hat{\kappa}_{b,w}$ for each classifier was computed using the SM, RRH, SGJK, and the SB techniques. Non-bias-corrected values are reported because the estimated bias is negligible (about 10% of the standard error). The SM method is based on theoretical results (Equations 2.1 and 2.4), and is used to compute $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ for the test cases obtained from the split-randomization. The RRH, SM, SGJK, and SB techniques were used to evaluate ESCA for comparison purposes even though ESCA does not require training and the grouped jackknife, the bootstrap, or the multinomial results (Equations 2.1 and 2.4) can be applied directly without recourse to the split recombination procedure. More appropriately, the same multinomial results (Equations 2.1 and 2.4) and the bootstrap method were used to evaluate ESCA. This is possible because ESCA does not require training. 200 bootstrap replications are used. The quality-control cases were evaluated using multinomial results (Equations 2.1 and 2.4) and the bootstrap. For the quality-control evaluation, 1000 bootstrap replications were used. Since ESCA does not require training, it is not necessary to use the SM, SGJK, or the SB methods for ESCA. L and $\kappa_{b,w}$ for ESCA can easily be estimated from the cross-classification table. Using known results for multinomials [Equation 2.4] we can compute $SE(L)$. To evaluate $SE(\kappa_{b,w})$ we can use the bootstrap technique.

The loss matrix used was provided by a POSCH expert, Dr. David Hunter, and is given in Table 7.4.

The assessments are grouped according to the scheme given in Table 7.5 to test the effect of grouping assessments after classification. The grouped losses are not comparable to ungrouped losses because:

1. The loss values in the table can be arbitrarily defined by the expert.

expert ↓\Cl. →	-3	-2	-1	-0	+0	+1	+2	+3
-3	0	5	40	80	85	90	95	100
-2	5	0	5	75	80	85	90	95
-1	40	5	0	5	50	70	85	90
-0	80	75	5	0	2	50	80	85
+0	85	80	50	2	0	5	75	80
+1	90	85	70	50	5	0	5	40
+2	95	90	85	80	75	05	0	5
+3	100	95	90	85	80	40	5	0

Table 7.4: Loss values ($\times 100$) for ungrouped assessments in the AEPOSCH application.

Ungrouped class	Ungrouped code	Grouped class	Grouped code
Much Worse	-3	Worse	1
Worse	-2	Worse	1
Slightly Worse	-1	Worse	1
Imperceptibly Worse	-0	No Change	2
Imperceptibly Better	+0	No Change	2
Slightly Better	+1	Better	3
Better	+2	Better	3
Much Better	+3	Better	3

Table 7.5: Grouping scheme for atherosclerosis evaluation.

2. Since the proportion of cases classified in each group change for each classifier, no equivalence between the loss values of grouped and ungrouped assessments can be defined that is valid for all classifiers.

Table 7.6 gives the loss function for the grouped classes. This loss function is also obtained from Dr. David Hunter.

The B3 dataset is evaluated as follows:

1. The SR procedure is applied to the B3 dataset with 21 sets ($g = 21$) and 30 cases in each set ($h = 30$).

2. The RRH, SM, SGJK, and SB methods are applied for each classifier, including ESCA.

3. Since ESCA does not require training, the value of criteria for ESCA's evaluations can be estimated using the multinomial results (Equations 2.1 and 2.4), and the bootstrap in addition to the techniques in the Step 2 above.

For the RRH technique, 50 replications were done. In each replication, 314 cases were drawn at random from the B3 dataset for training the classifiers (for those classifiers that need training) and the remaining 316 cases were used for testing the classifiers. Of the 316

expert $\downarrow$\Cl. $\rightarrow$	1	2	3
1	0	50	100
2	50	0	5
3	100	50	0

Table 7.6: Loss values ($\times$100) for grouped assessments in the AEPOSCH application.

test cases, 158 were randomly drawn for parameter evaluation of the constituent classifiers for the COLMS, and the remaining 158 were used for performance evaluation of the COLMS. The estimates of performance criteria reported for repeated random halving of ESCA, MLR, BP, C4, and the EDSS and comparisons among them were reported using the evaluations of cases from the test set. Performance statistics using repeated random halving involving the COLMS or a comparison between the COLMS and any other classifier was done using the performance evaluation set.

The quality-control data in the B3-QA dataset is evaluated using multinomial results (Equations 2.1). Tables 7.26 and 7.27 give the results of computing L using Equations 2.1 and 2.4 on the quality-control data and the cases evaluated by ESCA.

Table 7.7 gives the summary cross-classification of results of evaluating ESCA's assessments of the 630 cases in the B3 dataset. We see that there are considerably more cases assessed by the expert in the categories -3 to -1 (158 cases) than in the categories +1 to +3 (116 cases). The experts have assessed 329 cases in the categories -3 to -0 and 301 cases in the categories +0 to +3. Reasons for this apparent bias were given in Chapter 3. Since there is so little difference between the categories -0 (imperceptibly worse) and +0 (imperceptibly better) we do not consider the categories -0 and +0 when deciding whether there is a bias towards the "worse" categories by the expert. ESCA follows this suspected bias, giving 105 cases in the categories -3 to -1 and 94 cases in the categories +1 to +3.

Table 7.8 gives the results of evaluating the 630 cases in the B3 dataset after grouping. The -1 (+1) groups in Table 7.8 corresponds to the (-3 to -1) (+1 to +3) groups respectively in Table 7.7 and the bias suspected before is readily apparent.

Table 7.18 is the cross-classification table of experts' assessments and ESCA's assessments for the BR dataset. If we look at this cross-classification table, we see that the classifications by the experts appear even more biased (towards "worse") than for the B3 dataset. This is consistent with the observations in Chapter 3, because for the BR dataset there is an additional cause of bias, i.e., patients who have had a bypass or angioplasty will have the film with the operation as "Film B." In the BR dataset there are 451 cases evaluated by the experts in the categories -3 to -1 and 274 cases in the categories +1 to +3. This bias (of more cases in the -3 to -1 categories than in the +1 to +3 categories) is reflected in the results of most of the classifiers and can be seen in the cross-classification tables [Tables 7.18 - 7.24].

To detect whether this suspected bias is present in the readings of percent stenosis of arteries by the experts, we do the following test.

We divide the 23 arterial segments into two groups. The first group consists of the easily visualizable arteries (see Section 6.1) viz. proximal-lad, mid-lad, proximal-left-circumflex, mid-left-circumflex, proximal-right-coronary, and mid-right-coronary. The second group

consists of the remaining arteries. For the B3 and BR datasets we test whether there is any statistically significant difference at the 5% significance level between measurement(artery, Film A) and measurement(artery, Film B). Since we are doing multiple comparisons it is also determined whether the results are significant using Bonferroni p-values to adjust for the effect of multiple comparisons. The results of this comparison for the B3 dataset are in Table 7.9. After Bonferronizing, the mid-left-circumflex and the second-obtuse-marginal for the B3 dataset [Table 7.9] show a bias at the 5% significance level towards having a larger arterial stenosis recorded in Film B than in Film A. The results of a similar comparison for the BR dataset are given in Table 7.10. For the BR dataset [Table 7.10] a similar bias is observed after Bonferronizing for the following arterial segments: proximal-right-coronary, mid-right-coronary, distal-right-coronary, mid-lad, distal-lad, proximal-left-circumflex, mid-left-circumflex, first-obtuse-marginal, and other-segment-a. Reasons for this bias are given in Chapter 3.

For the B3 dataset, two arterial segments are different at the 5% significance level after taking into account the effect of multiple comparisons. Therefore we conclude that there is bias at least in these two arteries. Both these arteries show a positive difference. However, only five arterial segments show a negative mean difference, while 17 arterial segments show a positive difference. Therefore, for this reason we conclude that the experts are biased towards saying that the stenosis in the arteries has increased for the B3 dataset (see Chapter 3 for reasons for this bias).

For the BR dataset, eight arteries show a difference greater than 0 at the 5% significance level after taking into account the effect of multiple comparisons. Only two arterial segments show a negative mean difference but these are not significant at the 5% level even if we do not consider the effect of multiple comparisons. 17 arterial segments show a positive difference. 8 of these are significantly different at the 5% level if we do not consider the effect of multiple comparisons and 7 of these are significantly different even after considering the effect of multiple comparisons. For these reasons we conclude that the experts are biased towards saying that the stenosis in the arteries has increased for the BR dataset and that this bias is much more for the BR dataset than for the B3 dataset. See Chapter 3 for reasons for this bias.

From the BR dataset, the 65 cases that have been evaluated during quality-control studies are removed. The remaining 1105 cases are used for evaluating the classifiers.

7.2.1 Cross-classification tables for methods over the B3 dataset

Tables 7.11 - 7.17 give the summary cross-classification of results of evaluating the cases in the B3 dataset. The assessments are not grouped. The classifiers used are ESCA, MLR, BP, C4, COLMS, and EDSS. These are typical results and they are the results of the performance evaluation set for the first of the 50 replications of the RRH strategy. All matrices are based on the same sample.

From the tables we see that for the B3 dataset ESCA, MLR, and the COLMS are doing well as most of their assessments are close to the diagonal. BP and EDSS have their assessments around the diagonal but they are relatively more scattered. C4's assessments tend to be scattered the most. The marginal totals of all the classifiers are approximately well distributed. One of the reasons, why C4 does not do well is that C4 is based on a discriminant tree while the task requires all arterial segments to be considered and given

approximately equal weightage. C4 presumably produces a discriminant tree based on one set of arteries during training, and these are not as important during testing.

Expert ↓\ESCA →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	10	0	2	0	0	0	0	0	12
-2	6	12	23	10	0	0	0	0	51
-1	0	5	27	60	2	1	0	0	95
-0	0	2	14	109	43	2	0	1	171
+0	0	2	2	51	113	16	1	0	185
+1	0	0	0	5	34	31	7	0	77
+2	0	0	0	1	3	11	12	6	33
+3	0	0	0	0	0	2	2	2	6
Sum	16	21	68	236	195	63	22	9	630

Table 7.7: Cross-classification of experts' consensus opinion and ESCA's assessments for the B3 dataset.
Expert's assessments are in rows and ESCA's assessments are in columns.

Expert ↓\ESCA →	-1	0	+1	Sum
-1	85	72	1	158
0	20	316	20	356
+1	0	43	73	116
Sum	105	431	94	630

Table 7.8: Cross-classification of experts' grouped (consensus) opinion and ESCA's grouped assessments for the B3 dataset.
Expert's assessments are in rows and ESCA's assessments are in columns.

arterial segment	mean	t	p	Bonf.-p
right-ostium	0.014	0.12	0.902	$> .5$
prox-right-coronary (V)	1.184	1.87	0.060	.365
mid-right-coronary (V)	0.276	0.48	0.625	$> .5$
distal-right-coronary	0.269	0.61	0.537	$> .5$
prox-posterior-descending	0.089	0.36	0.713	$> .5$
distal-posterior-descending	-0.114	-1.87	0.061	$> .5$
left-ostium	0.000	X	X	X
left-main-coronary	0.057	0.29	0.770	$> .5$
prox-lad (V)	0.668	1.39	0.162	$> .5$
mid-lad (V)	0.908	1.82	0.068	.409
distal-lad	0.510	1.65	0.099	$> .5$
prox-left-circumflex (V)	0.142	0.40	0.684	$> .5$
mid-left-circumflex (V)	1.494	3.33	9.0e-4	.005
distal-left-circumflex	0.316	1.45	0.146	$> .5$
first-diagonal-lad	0.955	2.66	0.007	.133
second-diagonal-lad	-0.180	-0.61	0.540	$> .5$
third-diagonal-lad	-0.302	-1.65	0.097	$> .5$
first-obtuse-marginal	0.303	0.76	0.442	$> .5$
second-obtuse-marginal	1.234	3.58	3.6e-4	.006
third-obtuse-marginal	0.302	1.74	0.081	$> .5$
other-segment-a	-0.379	-1.16	0.245	$> .5$
other-segment-b	-0.004	-0.01	0.989	$> .5$
other-segment-c	0.424	1.58	0.114	$> .5$

Table 7.9: Bonferroni p-value for arterial differences on the B3 dataset.
(V) indicates that the artery is one of the six arteries which are usually visualizable clearly.
t = mean/se(mean). prox stands for proximal.

Arterial segment	mean	t	p	Bonf.-p
right-ostium	-0.098	-1.13	0.255	> 0.5
prox-right-coronary (V)	3.052	4.20	2.8e-05	1.7e-4
mid-right-coronary (V)	1.558	2.66	.007	.047
distal-right-coronary	1.532	3.16	0.002	.027
prox-posterior-descend	0.577	1.81	0.070	> 0.5
distal-posterior-descend	0.026	0.20	0.836	> 0.5
left-ostium	0.017	0.36	0.715	> 0.5
left-main-coronary	0.432	1.94	0.052	> 0.5
prox-lad (V)	3.471	0.59	0.554	> 0.5
mid-lad (V)	3.972	6.31	3.9e-10	2.3e-9
distal-lad	1.686	4.18	3.1e-05	5.3e-4
prox-left-circumflex (V)	2.487	4.93	9.0e-07	5.4e-6
mid-left-circumflex (V)	2.548	4.41	1.1e-05	6.6e-5
distal-left-circumflex	0.601	2.53	0.011	.196
first-diagonal-lad	0.735	1.94	0.05191	$> .5$
second-diagonal-lad	0.656	2.36	0.018	.307
third-diagonal-lad	0.299	1.77	0.076	$> .5$
first-obtuse-marginal	1.604	3.19	0.001	.024
second-obtuse-marginal	1.123	2.92	0.003	.060
third-obtuse-marginal	0.606	2.33	0.019	.332
other-segment-a	2.043	3.60	3.2e-4	.005
other-segment-b	0.743	2.18	0.028	.492
other-segment-c	-0.079	-0.31	0.750	$> .5$

Table 7.10: Bonferroni p-value for arterial differences on the BR dataset.
(V) indicates that the artery is one of the six arteries which are usually visualizable clearly.
t = mean/se(mean). prox stands for proximal.

Expert ↓\ESCA →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	2	0	1	0	0	0	0	0	3
-2	0	3	8	1	0	0	0	0	12
-1	0	1	7	15	1	0	0	0	24
-0	0	1	3	24	16	0	0	1	45
+0	0	2	0	15	22	3	0	0	42
+1	0	0	0	2	9	11	1	0	23
+2	0	0	0	0	1	2	3	1	7
+3	0	0	0	0	0	0	1	1	2
Sum	2	7	19	57	49	16	5	3	158

Table 7.11: Cross-classification of experts' assessment (gold standard) and ESCA's assessment for the B3 dataset.
Results of cross-classifying the 158 cases in the repeated random halving technique used for performance evaluation of the COLMS.

Expert ↓\MLR →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	2	0	1	0	0	0	0	0	3
-2	0	1	8	3	0	0	0	0	12
-1	0	1	8	14	1	0	0	0	24
-0	0	0	6	21	16	1	0	1	45
+0	0	0	2	15	23	2	0	0	42
+1	0	0	0	3	13	7	0	0	23
+2	0	0	0	0	2	2	3	0	7
+3	0	0	0	0	0	1	1	0	2
Sum	2	2	25	56	55	13	4	1	158

Table 7.12: Cross-classification of experts' assessment (gold standard) and MLR's assessment for the B3 dataset.
Results of cross-classifying the 158 cases in the repeated random halving technique used for performance evaluation of the COLMS.

Expert ↓\BP →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	0	2	1	0	0	0	0	0	3
-2	0	4	6	2	0	0	0	0	12
-1	0	2	10	5	6	1	0	0	24
-0	0	0	6	14	20	3	2	0	45
+0	0	1	2	7	23	8	1	0	42
+1	0	0	2	3	2	9	5	2	23
+2	0	0	0	0	2	0	5	0	7
+3	0	0	0	0	0	0	2	0	2
Sum	0	9	27	31	53	21	15	2	158

Table 7.13: Cross-classification of experts' assessment (gold standard) and BP's assessment for the B3 dataset.
Results of cross-classifying the 158 cases in the repeated random halving technique used for performance evaluation of the COLMS.

Expert ↓\C4 →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	0	1	1	1	0	0	0	0	3
-2	0	4	2	0	2	4	0	0	12
-1	0	3	7	3	7	4	0	0	24
-0	0	6	3	15	19	2	0	0	45
+0	1	3	8	13	10	3	4	0	42
+1	0	3	1	8	7	3	1	0	23
+2	0	0	1	2	2	1	1	0	7
+3	0	0	1	1	0	0	0	0	2
Sum	1	20	24	43	47	17	6	0	158

Table 7.14: Cross-classification of experts' assessment (gold standard) and C4's assessment for the B3 dataset.
Results of cross-classifying the 158 cases in the repeated random halving technique used for performance evaluation of the COLMS.

Expert ↓\COLMS →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	2	0	1	0	0	0	0	0	3
-2	0	3	9	0	0	0	0	0	12
-1	0	1	11	11	1	0	0	0	24
-0	0	1	6	21	16	0	0	1	45
+0	0	2	0	15	22	3	0	0	42
+1	0	0	0	2	9	11	1	0	23
+2	0	0	0	0	1	2	4	0	7
+3	0	0	0	0	0	0	2	0	2
Sum	2	7	27	49	49	16	7	1	158

Table 7.15: Cross-classification of experts' assessment (gold standard) and COLMS's assessment for the B3 dataset.
Results of cross-classifying the 158 cases in the repeated random halving technique used for performance evaluation of the COLMS.

Exp. ↓\EDSS →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	2	0	1	0	0	0	0	0	3
-2	0	2	8	1	0	0	0	0	11
-1	0	1	7	12	2	0	0	0	22
-0	0	0	5	20	18	0	0	1	44
+0	0	1	1	10	26	3	0	0	41
+1	0	0	0	3	9	10	1	0	23
+2	0	0	0	0	1	2	3	0	6
+3	0	0	0	0	0	0	2	0	2
Sum	2	4	22	46	56	15	6	1	152

Table 7.16: Cross-classification of experts' assessment (gold standard) and EDSS's unrejected assessment for the B3 dataset.
Results of cross-classifying the cases that were not rejected in the 158 cases in the repeated random halving technique used for performance evaluation of the COLMS.

Exp. ↓\EDSS →	R(-3)	R(-2)	R(-1)	R(-0)	R(+0)	R(1)	R(2)	R(3)	Sum
-3	0	0	0	0	0	0	0	0	0
-2	0	0	0	1	0	0	0	0	1
-1	0	0	1	1	0	0	0	0	2
-0	0	0	0	1	0	0	0	0	1
+0	0	0	0	1	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	1
3	0	0	0	0	0	0	0	0	0
Sum	0	0	1	4	0	0	1	0	6

Table 7.17: Cross-classification of experts' assessment (gold standard) and EDSS's rejected assessment for the B3 dataset.
Results of cross-classifying the cases that were rejected in the 158 cases in the repeated random halving technique used for performance evaluation of the COLMS.

7.2.2 Cross-classification tables for methods over the BR dataset

The classifiers were trained on all the cases in the B3 dataset and tested on all the cases in the BR dataset. Tables 7.18 - 7.24 give the summary cross-classification of results of evaluating the 1105 cases in the BR dataset when the assessments are not grouped. The classifiers evaluated are ESCA, MLR, BP, C4, COLMS, and EDSS. Table 7.25 gives the results of grouping ESCA's evaluation of the 1105 cases in the BR dataset.

We see from Tables 7.18 - 7.24 that ESCA, MLR, and BP's assessments are chiefly along the diagonal signifying that they agree with the experts. COLMS's assessments are chiefly along the diagonal and more so than those of ESCA, MLR, and BP. C4's assessments are scattered all over the matrix indicating that it is doing a poor job of classifying the cases. Section 7.2.1 gives some reasons why this is so. The EDSS's assessments are also along the diagonal. Besides this visual assessment we can make use of the estimated value of L, $\kappa_{b,w}$ and estimated differences of L and $\kappa_{b,w}$ in this chapter to decide whether one classifier is better than another.

Expert $\downarrow$\ESCA $\rightarrow$	-3	-2	-1	-0	+0	1	2	3	Sum
-3	54	10	2	0	0	0	0	0	66
-2	64	42	52	10	1	1	0	0	170
-1	12	21	64	106	9	2	1	0	215
-0	2	2	10	103	48	4	1	1	171
+0	0	1	1	55	130	20	2	0	209
+1	0	0	0	6	54	72	24	8	164
+2	0	0	0	1	3	29	35	26	94
+3	0	0	0	0	0	2	3	11	16
Sum	132	76	129	281	245	130	66	46	1105

Table 7.18: Cross-classification of experts' assessment (gold standard) and ESCA's assessment for the BR dataset.

In Table 7.24, 24 cases are rejected when the best assessment is -3 to -0 while 61 cases are rejected when the best assessment is +0 to +3. Therefore there are many more rejected cases in the better categories (+3 to +0) than in the worse categories (-0 to -3). Examining Table 7.24, we see that most of the rejects in the better categories are due to the cell (+2, R(+3)). This indicates that the experts' assessment was +2, and the best method reported an assessment of +3, but the other methods did not agree with the best method. ESCA was chosen as the best method, and indeed looking at the cell (expert = +2, ESCA = +3) in Table 7.18, we see that the count value of this cell is 26 which is much higher than the corresponding cell counts of the other techniques (i.e., MLR count (+2,+3) =19; BP count (+2, +3) = 2; C4 count (+2, +3) = 1). So the methods will tend to disagree more in this category, causing a higher number of disagreements in the better categories.

Expert ↓\MLR →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	56	8	1	0	0	0	1	0	66
-2	62	38	50	18	1	0	1	0	170
-1	8	23	56	104	22	1	1	0	215
-0	1	7	9	85	68	0	1	0	171
+0	0	1	1	50	141	15	1	0	209
+1	0	0	1	8	74	58	20	3	164
+2	0	0	0	1	11	32	31	19	94
+3	0	0	0	1	1	1	3	10	16
Sum	127	77	118	267	318	107	59	32	1105

Table 7.19: Cross-classification of experts' assessment (gold standard) and MLR's assessment for BR dataset.

Expert ↓\BP →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	13	49	4	0	0	0	0	0	66
-2	19	109	32	10	0	0	0	0	170
-1	4	45	82	71	11	2	0	0	215
-0	0	9	34	85	37	6	0	0	171
+0	0	2	3	82	89	31	1	1	209
+1	0	0	0	14	43	100	5	2	164
+2	0	0	0	2	12	75	3	2	94
+3	0	0	0	0	2	14	0	0	16
Sum	36	214	155	264	194	228	9	5	1105

Table 7.20: Cross-classification of experts' assessment (gold standard) and BP's assessment for BR dataset.

Expert ↓\C4 →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	7	19	18	16	4	2	0	0	66
-2	10	43	42	48	19	6	2	0	170
-1	5	25	50	62	61	10	2	0	215
-0	2	5	18	67	64	14	1	0	171
+0	0	8	29	68	85	17	2	0	209
+1	1	4	35	36	58	15	11	4	164
+2	4	3	10	17	33	11	15	1	94
+3	1	1	2	2	1	0	9	0	16
Sum	30	108	204	316	325	75	42	5	1105

Table 7.21: Cross-classification of experts' assessment (gold standard) and C4's assessment for the BR dataset.

Expert ↓\COLMS →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	53	11	2	0	0	0	0	0	66
-2	55	51	55	7	1	1	0	0	170
-1	8	24	77	94	9	2	1	0	215
-0	1	3	14	99	49	4	1	0	171
+0	0	0	1	56	122	29	1	0	209
+1	0	0	0	6	44	98	15	1	164
+2	0	0	0	1	5	40	34	14	94
+3	0	0	0	0	0	3	5	8	16
Sum	117	89	149	263	230	177	57	23	1105

Table 7.22: Cross-classification of experts' assessment (gold standard) and COLMS's assessment for the BR dataset.

Exp. ↓\EDSS →	-3	-2	-1	-0	+0	1	2	3	Sum
-3	52	10	2	0	0	0	0	0	64
-2	50	52	45	10	0	0	0	0	157
-1	8	24	58	106	9	1	0	0	206
-0	1	4	10	100	47	3	1	0	166
+0	0	0	1	55	127	21	1	0	205
+1	0	0	0	7	56	75	13	1	152
+2	0	0	0	1	3	33	26	3	66
+3	0	0	0	0	1	0	3	4	8
Sum	111	90	116	279	243	133	44	8	1024

Table 7.23: Cross-classification of experts' assessment (gold standard) and EDSS's for BR dataset for unrejected cases.

Expert ↓\EDSS →	R(-3)	R(-2)	R(-1)	R(-0)	R(+0)	R(1)	R(2)	R(3)	Sum
-3	1	0	1	0	0	0	0	0	2
-2	7	1	3	1	0	1	0	0	13
-1	0	3	3	2	0	0	1	0	9
-0	1	0	0	0	2	1	0	1	5
+0	0	1	0	1	0	2	0	0	4
+1	0	0	0	0	0	3	2	7	12
+2	0	0	0	0	0	2	4	22	28
+3	0	0	0	0	0	1	0	7	8
Sum	9	5	7	4	2	10	7	37	81

Table 7.24: Cross-classification of experts' assessment (gold standard) and EDSS's for BR dataset for rejected cases.
Experts' assessment is in rows and EDSS's assessments is in columns. R(α) means ESCA (chosen as the most-trusted classifier) suggested assessment α but the EDSS rejected the case because the discrepancy among the classifiers was large.

Expert $\downarrow$\ESCA $\rightarrow$	-1	0	+1	Sum
-1	321	126	4	451
0	16	336	28	380
+1	0	64	210	274
Sum	337	526	242	1105

Table 7.25: Cross-classification of experts' grouped assessment (gold standard) and ESCA's grouped assessment for the BR dataset.
Expert's assessments are in rows and ESCA's assessments are in columns.

Looking at Table 7.25, we see that the experts have reported many more assessments in the -3 to -1 categories than in the +3 to +1 categories for the BR dataset. This is as expected from the observations in Chapter 3 and Section 7.2 where it was concluded that the experts are biased towards reporting positive differences in the stenosis. Therefore the assessments tend towards the worse categories.

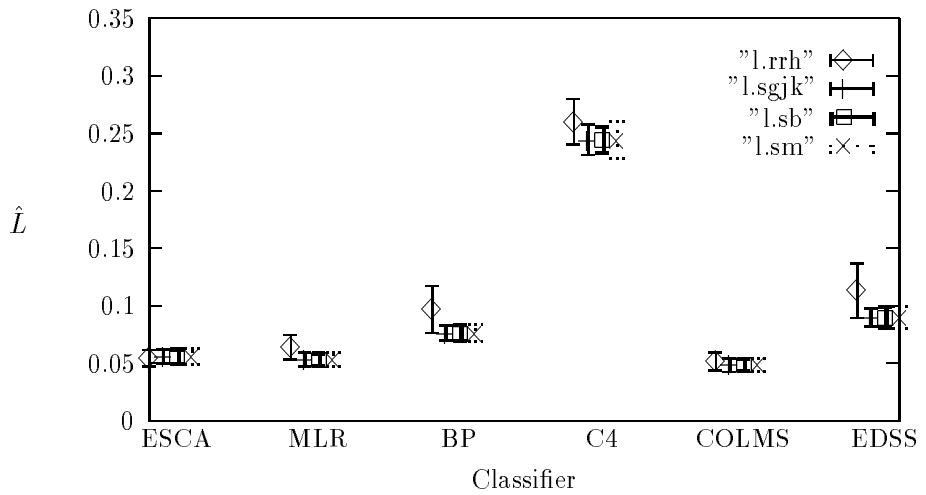

Figure 7.1: $\hat{L}$ and $SE(L)$ for various classifiers computed over the B3 dataset.

7.3 Summary figures for the B3 and BR datasets

Before looking at any more tables, it is instructive to look at the following summary figures. In all experiments, it is observed that the bias is less than 10% of the standard error of the quantity being estimated. For this reason, the bias is not shown in the figures. For all the figures [Figure 7.1, Figure 7.2, Figure 7.3, and Figure 7.4] $SE(L)$ and $SE(\kappa_{b,w})$ computed by the RRH technique is about twice that of the SGJK and SB techniques. This is because the SGJK and SB techniques use all the available cases for testing, while the RRH technique can use only one-fourth the available number of cases for testing, and the estimated standard error of the mean is inversely proportional to the square root of the number of cases.

Figure 7.1 gives the value of $\hat{L}$ and its standard error $SE(L)$ for different classifiers computed using the RRH, SM, SGJK, and the SB techniques over the B3 dataset. The bars in the figure indicate estimated mean $\pm$ standard error. The values for the SM technique are obtained from Table 7.29, those for the RRH technique are obtained from Table 7.31, those for the SGJK technique are obtained from Table 7.32, and those for the SB technique obtained from Table 7.33. From Figure 7.1, we see that for the B3 dataset, ESCA, MLR, and the COLMS have approximately the same value of $\hat{L}$ while C4 has a considerably higher value of $\hat{L}$.

Figure 7.2 gives the value of $\hat{L}$ and its standard error $SE(L)$ for different classifiers computed using the RRH, SGJK, and the SB techniques over the B3 dataset after grouping the assessments. The values for the RRH technique are obtained from Table 7.34, those for the SGJK technique are obtained from Table 7.35, and those for the SB technique obtained from Table 7.36. From Figure 7.2 we see that, for grouped assessments over the B3 dataset, ESCA, MLR, and the COLMS have approximately the same value of $\hat{L}$ while C4 has a considerably higher value of $\hat{L}$.

Figure 7.3 gives the value of $\hat{\kappa}_{b,w}$ and $SE(\kappa_{b,w})$ for different classifiers computed using

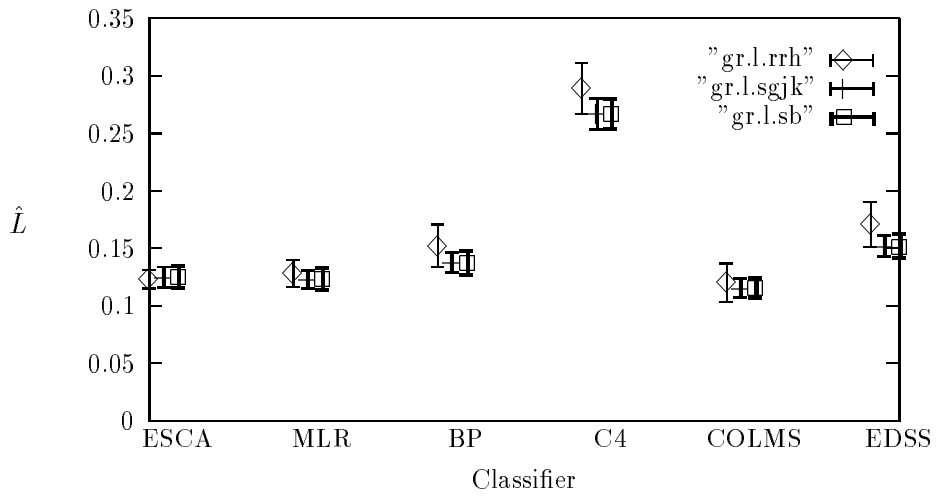

Figure 7.2: $\hat{L}$ and $SE(L)$ for various classifiers computed over the B3 dataset. The assessments are grouped.

the RRH, SM, SGJK, and the SB techniques over the BR dataset. The values for the RRH technique are obtained from Table 7.31, those for the SGJK technique are obtained from Table 7.32, and those for the SB technique obtained from Table 7.33. For the B3 dataset, we see from Figure 7.3 that ESCA, MLR, and the COLMS have approximately the same value of $\hat{\kappa}_{b,w}$ while C4 has a considerably lower value of $\hat{\kappa}_{b,w}$.

Figure 7.4 gives the values of $\hat{\kappa}_{b,w}$ and $SE(\kappa_{b,w})$ for different classifiers computed using the RRH, SGJK, and the SB techniques over the B3 dataset after grouping the assessments. The values for the RRH technique are obtained from Table 7.34, those for the SGJK technique are obtained from Table 7.35, and those for the SB technique obtained from Table 7.36. From Figure 7.4 we see that ESCA, MLR, and the COLMS have approximately the same value of $\hat{\kappa}_{b,w}$, while C4 has a considerably lower value of $\hat{\kappa}_{b,w}$ than the other classifiers.

The summary results of computing $\hat{L}$, $SE(L)$, $\hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$ over the BR dataset for classifiers trained over all the cases in the B3 dataset are given in Figures 7.5 and 7.6. The values for these figures are obtained from Table 7.39. Only the bootstrap method is used for evaluating the BR dataset, since it can compute $SE(\kappa_{b,w})$, and the multinomial results cannot be used to evaluate $SE(\kappa_{b,w})$. We see from Figure 7.5 that ESCA, MLR, BP, and COLMS have approximately the same value of $\hat{L}$, EDSS has a larger value of $\hat{L}$, and C4 has a considerably larger value of $\hat{L}$ than the other classifiers. By observing the Figures for the B3 dataset and those for the BR dataset, we see that because the number of cases in the BR dataset (1170 cases) is larger than the number of cases in the B3 dataset (630 cases), and there is no variation because of training, the standard errors in the BR dataset are less than those in the B3 dataset.

We see from Figure 7.6 that ESCA, MLR, BP, and COLMS have approximately the same value of $\hat{\kappa}_{b,w}$, EDSS has a smaller value of $\hat{\kappa}_{b,w}$, and C4 has a considerably lower value of $\kappa_{b,w}$ than the other classifiers.

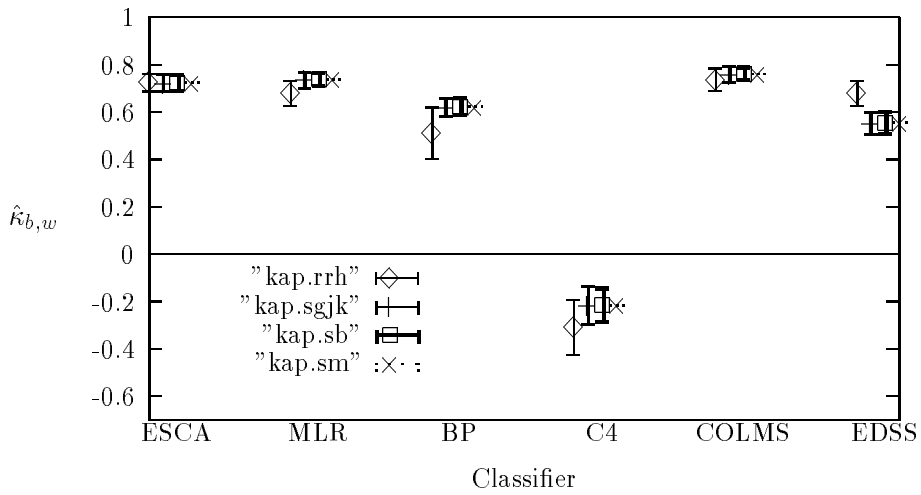

Figure 7.3: $\hat{\kappa}_{b,w}$ and $SE(\kappa_{b,w})$ for various classifiers computed over the B3 dataset.

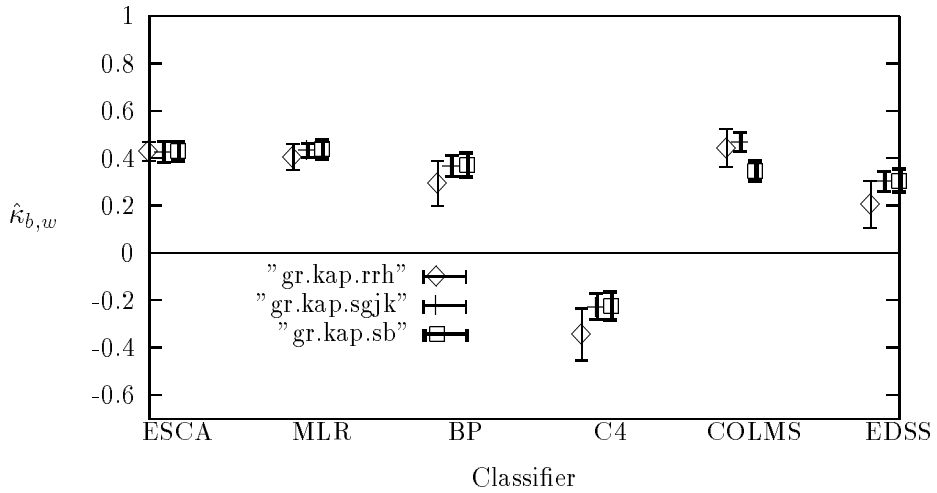

Figure 7.4: $\hat{\kappa}_{b,w}$ and $SE(\kappa_{b,w})$ for various classifiers computed over the B3 dataset. The assessments are grouped.

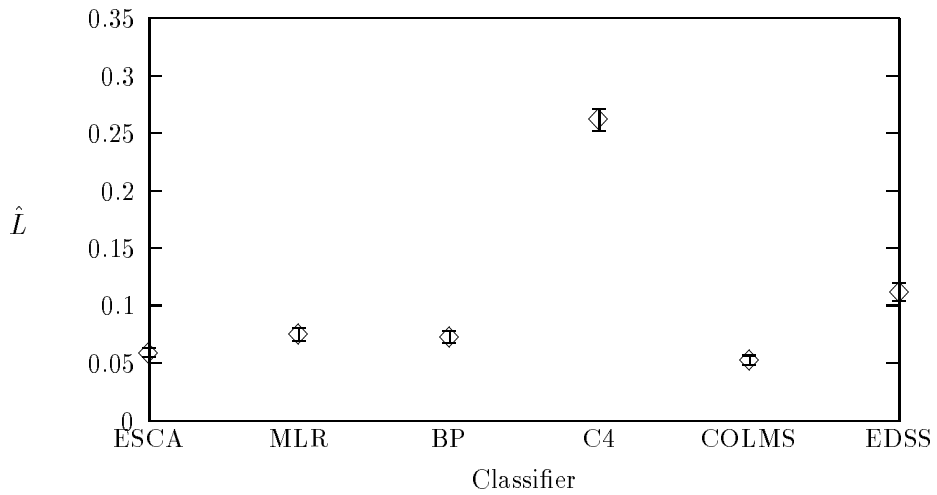

Figure 7.5: $\hat{L}$ and $SE(L)$ computed using the bootstrap for classifiers over the BR dataset. 200 replications are used.

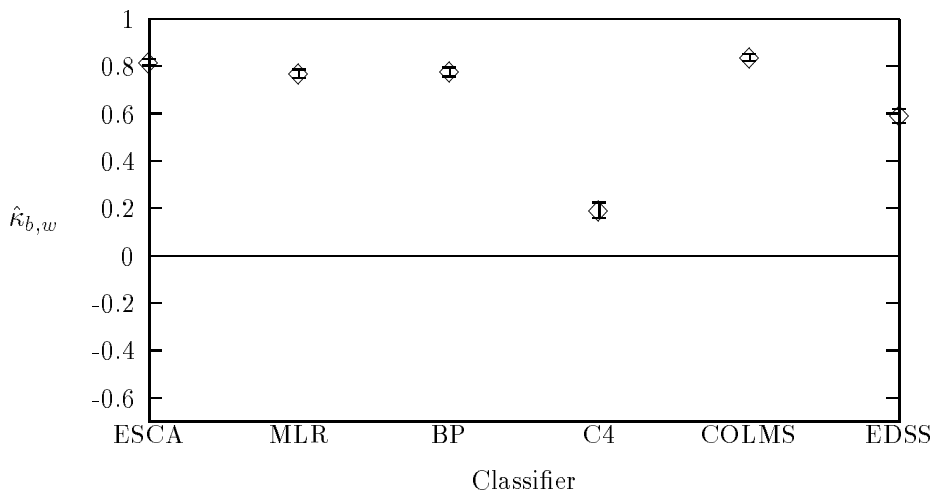

Figure 7.6: $\hat{\kappa}_{b,w}$ and $SE(\kappa_{b,w})$ computed using the bootstrap for classifiers over the BR dataset.
200 replications are used.

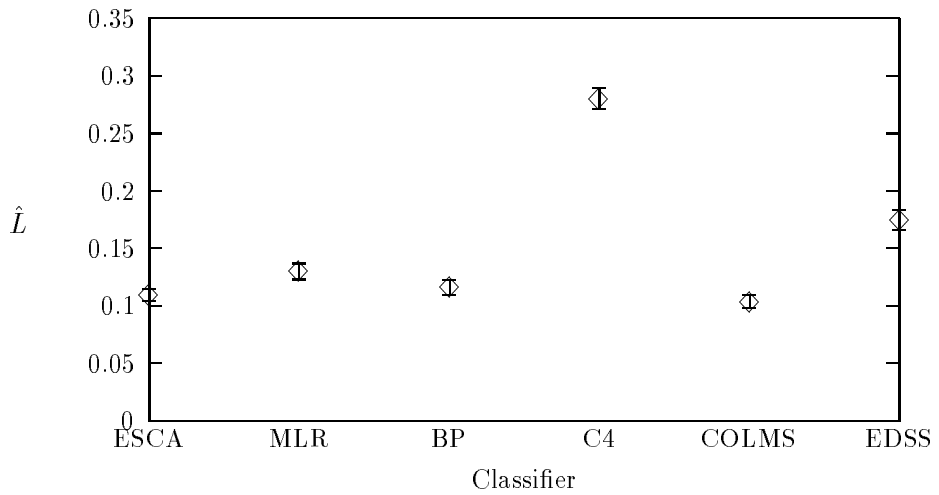

Figure 7.7: $\hat{L}$ and $SE(L)$ computed using the bootstrap for classifiers over the BR dataset. The assessments are grouped.
200 replications are used.

The results over the BR dataset for grouped assessments for classifiers trained over all the cases in the B3 dataset are given in Figures 7.7 and 7.8. The figures are plotted using data from Table 7.41. The bootstrap method is used again for the same reasons as for estimating parameters over the BR dataset for ungrouped assessments.

We see from Figure 7.7 that ESCA, MLR, BP, and COLMS have approximately the same value of $\hat{\kappa}_{b,w}$, EDSS has a larger value of $\hat{L}$, and C4 has a considerably larger value of $\hat{L}$ than the other classifiers.

We see from Figure 7.8 that, for grouped assessments, ESCA, MLR, BP, and COLMS have approximately the same value of $\hat{\kappa}_{b,w}$, EDSS has a smaller value of $\hat{\kappa}_{b,w}$, while C4 has a considerably larger value of $\hat{\kappa}_{b,w}$ than the other classifiers.

7.4 Summary statistics for the B3 and BR datasets

From Table 7.26, we see that ESCA's loss for the B3 and BR datasets is lower than the inter-expert loss on the B3-QA and the BR-QA datasets. This might be interpreted as saying that ESCA does a better job than experts at doing the task of classifying the case in the -3 to +3 scale. Of course, since the experts have coded the assessments from which ESCA works, ESCA has one less source of error/uncertainty than the experts who assessed the case working directly from the films. What is happening is that on the average ESCA agrees with any one of the subpanels more than the two subpanels agree with each other. One way for this to happen is for ESCA to agree with the experts when they agree with each other. If they disagree, ESCA is presumably likely to choose between them. ESCA does better over the B3 dataset for which it was designed than it does over the BR dataset.

Table 7.28 gives the result of bootstrapping ESCA using 200 replications over the entire

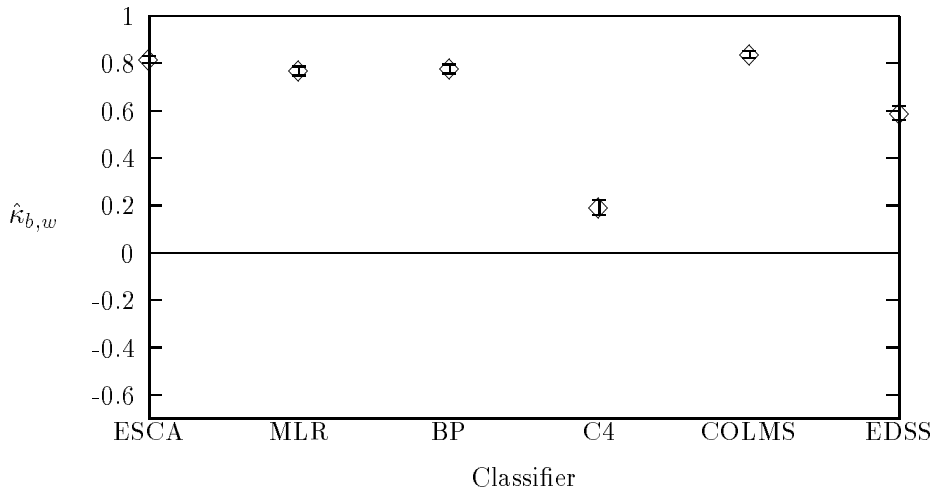

Figure 7.8: $\hat{\kappa}_{b,w}$ and $SE(\kappa_{b,w})$ computed using the bootstrap for classifiers over the BR dataset. The assessments are grouped.
200 replications are used.

Evaluations used	Number of cases	L	$SE(L)$	$\hat{\kappa}_{b,w}$
B3-QA	44	.077273	n/a	n/a
BR-QA	65	.069538	n/a	n/a
B3 (ESCA)	630	.055603	.006450	.722644
BR (ESCA)	1105	.059240	.004721	.816664

Table 7.26: Multinomial calculation of $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ for ungrouped assessments over the B3-QA and BR-QA datasets compared with ESCA's $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ for B3 and BR datasets.
For the quality-control assessments $SE(L)$, $\kappa_{b,w}$, and $SE(\kappa_{b,w})$ cannot be computed. L and $SE(L)$ are theoretically estimated using Equations 2.1 and 2.4.

Evaluations used	Number of cases	L	$SE(L)$	$\hat{\kappa}_{b,w}$
B3-QA	44	.181818	n/a	n/a
BR-QA	65	.153846	n/a	n/a
B3 (ESCA)	630	.124603	.010006	.427007
BR (ESCA)	1105	.109502	.007154	.666207

Table 7.27: Multinomial calculation of $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ for grouped assessments over the B3-QA and BR-QA datasets compared with ESCA's L, $SE(L)$, and $\hat{\kappa}_{b,w}$ for B3 and BR datasets.
L and $SE(L)$ using Equations 2.1 and 2.4. For the quality-control assessments $SE(L)$, $\kappa_{b,w}$, and $SE(\kappa_{b,w})$ cannot be computed.

Assessments	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias($\kappa_{b,w}$)
Ungrouped	.055603	.006078	.000492	.722644	.032460	-.006451
Grouped	.124603	.009386	.000048	.427007	.041544	.000489

Table 7.28: $\hat{L}$ and $\hat{\kappa}_{b,w}$ computed for ESCA using the bootstrap with 200 bootstrap replications over the B3 dataset.
The maximum absolute value of kurtosis is 0.83.

B3 dataset for grouped and ungrouped assessments. We see that $\hat{L}$ for grouped assessments is higher than $\hat{L}$ for ungrouped assessments. This appears counter-intuitive because we would expect that the classifier would do better when there are fewer categories. However, we should remember that the two results are genuinely incomparable and we should not compare them. As stated before, one reason for the results to be incomparable is because there is no relationship between the loss values used in grouped assessments and the loss values used in ungrouped assessments. Grouping the assessments can shift many assessments from a low loss-value cell to a higher loss-value cell. This explains why $\hat{L}$ goes up after grouping assessments, even though the classifier has to separate the cases into fewer categories.

Table 7.30 gives the values of $\hat{L}$ and $\hat{\kappa}_{b,w}$ with their standard errors for ungrouped assessments for trainable classifiers using the RRH technique over the training set. We do not need to compute the value of $\hat{L}$ and $\hat{\kappa}_{b,w}$ for training for ESCA using repeated random halving because ESCA does not require training. Table 7.28 gives the values of $\hat{L}$ and $\hat{\kappa}_{b,w}$ for ESCA, which can be used as training or test values.

Tables 7.29, 7.31, 7.32, and 7.33 give the values of $\hat{L}$ and $\hat{\kappa}_{b,w}$ with their standard errors for ungrouped assessments using the SM method based on multinomial results (Equations 2.1 and 2.4), repeated random halving, the SGJK technique, and the SB method, respectively. The values for $\hat{L}$, $SE(L)$, $\hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$ computed using the SGJK and SB methods are approximately the same. The standard errors reported by the RRH technique are greater than $\sqrt{2}\times$ (standard errors reported by the SGJK and SB methods) for ESCA, MLR, BP, C4 and EDSS, even though the size of the test set for the above classifiers is about half the size of the entire set used in the SGJK and SB methods. For the COLMS, the size of the test cases of the RRH technique is about one-fourth the size of the entire set used in SGJK and SB to evaluate the COLMS. However, $SE(L)$ and $SE(\kappa_{b,w})$ reported by the RRH technique is more than $\sqrt{4} = 2\times$ (standard error reported by the SGJK and the SB methods). This shows that the RRH technique is taking into account the variation in the training set when estimating the standard error. For a correct evaluation, the technique used should not consider the variation in the training set, because when the classifier is used in practice it will be trained over all the cases in the sample.

Comparing the values of $\hat{L}$ and $\hat{\kappa}_{b,w}$ in Tables 7.30 and 7.31, we note a large increase for C4 and a smaller, but still appreciable, increase for BP in Table 7.30.

Table 7.29 gives the values of $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ for classifiers over the B3 dataset using the SM method. The value of $SE(L)$ is obtained using known results for multinomials [Equation 2.1] and is the value that the SB and SGJK methods are trying to approximate in Tables 7.32 and 7.33.

Classifier	L	$SE(L)$	$\kappa_{b,w}$
ESCA	.055603	.006450	.722644
MLR	.052825	.006130	.736500
BP	.075905	.007729	.621378
C4	.243984	.015806	-.217023
COLMS	.048270	.005683	.759224
EDSS	.089635	.009710	.552890

Table 7.29: $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ computed using the SM technique for classifiers over the B3 dataset.
L, $SE(L)$, and $\hat{\kappa}_{b,w}$ are computed using Equations 2.1, 2.4, and 2.6 respectively. All the cases in the test set of the split recombination procedure are used in the computation. As theoretically expected, L and $SE(L)$ are the same after randomizing the order of the assessments.

Classifier	L	$SE(L)$	$\kappa_{b,w}$	$SE(\kappa_{b,w})$
MLR	.040145	.006221	.800361	.027783
BP	.028362	.007480	.858913	.035072
C4	.082195	.012954	.590269	.062684

Table 7.30: L and $\kappa_{b,w}$ estimated for the training set using the RRH technique over the B3 dataset.
50 replications are used.

Table 7.30 gives the values of $\hat{L}$, $SE(L)$, $\hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$ obtained using the RRH technique over the training set for the B3 dataset. Comparing these values with those from multinomial results [Table 7.29] or repeated random halving [Table 7.31], we see that

$$L(train, M) < L(test, M) \forall M \in \{MLR, BP, C4\}$$

where $L(train, M)$ is the loss for method M when it is trained and tested on the same training set. $L(test, M)$ is the loss for method M when it is trained on the same training set used to obtain $L(train, M)$ but tested on a different test set.

This confirms that the results over the training set are optimistically biased. Since ESCA does not need training, we do not include ESCA in the comparison. Table 7.31 indicates the above result as well as Table 7.29 using multinomial calculations from the split recombination test set. This is a welcome result since the validity of the SR conjecture has not been proven.

Classifier	L	$SE(L)$	$\kappa_{b,w}$	$SE(\kappa_{b,w})$
ESCA	.054649	.007237	.724810	.037647
MLR	.064098	.010455	.677805	.051592
BP	.096982	.020553	.511057	.109020
C4	.260022	.019765	-.309568	.116253
COLMS	.051830	.007898	.736392	.047389
EDSS	.113397	.023580	.677805	.051592

Table 7.31: L and $\kappa_{b,w}$ estimated using the RRH technique over the B3 dataset. The values for the classifiers besides the COLMS are estimated over the test cases of the RRH technique. The values for the COLMS are estimated over the performance evaluation cases. 50 replications are used.

Classifier	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias$(\kappa_{b,w})$
ESCA	.055603	.006040	.000006	.722644	.036181	-.001417
MLR	.052825	.006215	.000011	.736500	.034365	-.001057
BP	.075905	.006465	-.000007	.621378	.036763	-.001367
C4	.243984	.012879	.000004	-.217023	.081276	-.004021
COLMS	.048270	.005888	-.000005	.759224	.035981	-.001523
OLMS	.050111	.006035	-.000004	.746075	.036647	-.001541
EDSS	.089635	.007596	-.000005	.552890	.047682	-.002050

Table 7.32: L and $\kappa_{b,w}$ estimated using the SGJK method over the B3 dataset.

Tables 7.34, 7.35, and 7.36 give the values of $\hat{L}$ and $\hat{\kappa}_{b,w}$ with their standard errors for grouped assessments using the RRH technique, the SGJK technique, and the SB technique, respectively, over the B3 dataset. The values for $\hat{L}$, $SE(L)$, $\hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$ computed using the SGJK and SB methods are approximately the same. The standard errors reported by the RRH technique are greater than those expected when compared to the standard errors reported by the SGJK and SB techniques for reasons similar to those discussed before. We see that when going from the B3 to the BR dataset, the value of $\hat{L}$ for BP actually decreases, while $\hat{L}$ for the other classifiers increases. $\hat{L}$ for ESCA increases less than $\hat{L}$ for MLR, showing that ESCA generalizes better to the BR dataset than MLR. For grouped assessments, when going from the B3 dataset to the BR dataset, $\hat{L}$ for ESCA, BP, and COLMS decreases while that for MLR, C4, and EDSS increases. Again, ESCA is better than MLR when generalizing to the BR dataset.

The difference $\hat{L}(ESCA) - \hat{L}(MLR)$ increases in magnitude with negative sign for ungrouped assessments when going from the B3 to the BR dataset. $\hat{L}(ESCA) - \hat{L}(MLR)$ decreases from 0.002 which is not significant at the 5% significance level, to -0.02, which is significant at the 5% significance level (even after Bonferronizing for $6 \times 5/2 = 15$ multiple comparisons). This is an advantage when using ESCA because it generalizes better to unseen datasets. The OLMS has been introduced at a later stage and is therefore not considered when Bonferronizing the p-values of the classifiers.

Classifier	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias$(\kappa_{b,w})$
ESCA	.055603	.006078	.000492	.722644	.032327	-.003016
MLR	.052825	.005347	.000898	.736500	.027398	-.004894
BP	.075905	.007090	.000413	.621378	.035224	-.002528
C4	.243984	.011155	.000266	-.217023	.072025	-.003755
COLMS	.048270	.005165	.000527	.759224	.027319	-.003114
OLMS	.050111	.005610	.000707	.746075	.020907	-.003086
EDSS	.089635	.009077	.000358	.552890	.047606	-.002602

Table 7.33: L and $\kappa_{b,w}$ estimated using the SB method over the B3 dataset. The maximum absolute value of kurtosis is .70. 200 bootstrap replications are used.

The COLMS is significantly better (in terms of lower $\hat{L}$) at the 5% significance level than ESCA, EDSS, and C4 (but not MLR) for the B3 dataset. Comparisons with BP are not done because BP has a high value of $\hat{L}$. After Bonferronizing with a multiplier of 4, the COLMS is not significantly better at the 5% significance level than ESCA. The COLMS is significantly better (in terms of lower $\hat{L}$) at the 5% significance level than the EDSS. The COLMS is significantly better at the 5% significance level than ESCA, MLR, C4, and EDSS for the BR dataset for ungrouped and grouped assessments. The comparison with BP has not been made because BP's $\hat{L}$ was high on the B3 dataset. After Bonferronizing the 4 paired z-score statements we see that the COLMS is significantly better for the BR dataset at the 5% level than MLR and EDSS but not ESCA (Bonferronized p-value = .09152 > .05).

The difference $\hat{L}(ESCA - MLR)$ is .001 for grouped assessments in the B3 dataset, which is not significant at the 5% significance level. The COLMS is significantly better at the 5% significance level than ESCA, EDSS, and C4 (comparisons for BP are not done) but not MLR for grouped assessments over the B3 dataset. For grouped assessments over the BR dataset, ESCA is significantly better at the 5% significance level than MLR (even after Bonferronizing with a multiplier of 15, while the COLMS is significantly better (has a lower $\hat{L}$) at the 5% significance level than ESCA, MLR, C4, and EDSS (comparison with BP is not done) and significantly better (has a lower $\hat{L}$) at the 5% significance level than MLR and EDSS after Bonferronizing with a multiplier of 15.

MLR has a lower value of $\hat{L}$ than ESCA for the B3 dataset, but the difference is not statistically significant at the 5% significance level ($z = 0.567$). MLR has a higher value of $\hat{\kappa}_{b,w}$ than ESCA for the B3 dataset but the difference is not statistically significant at the 5% significance level ($z = -0.5540$). MLR has a lower value of $\hat{L}$ than ESCA for the BR dataset and the difference is statistically significant at the 5% significance level ($z = -3.064$). After multiplying the p-value by $6 \times 5/2 = 15$ we get a Bonferroni p-value of $15 \times 0.002277 = 0.03415 < .05$. Therefore, ESCA is better than MLR at the 5% significance level in terms of $\hat{L}$ even after taking into consideration the effect of multiple comparisons for the BR dataset. ESCA has a higher value of $\hat{\kappa}_{b,w}$ than MLR for the BR dataset and the difference is statistically significant at the 5% significance level ($z = 3.009$). After multiplying the p-value by $6 \times 5/2 = 15$ we get a Bonferroni p-value of $15 \times 0.002726 = 0.04089 < .05$. Bonferronizing the z-scores in this case is an over-correction for the effect of multiple values since we did not plan on doing all possible pairwise comparisons between the classifiers. Looking at the Bonferronized and the non-Bonferronized p-values for the comparisons gives us a better picture than looking at either one of them alone. As stated before, using ESCA has an advantage versus using MLR because ESCA generalizes better to unseen datasets.

Tables 7.37 and 7.38 give the values of $\hat{L}$ and $\hat{\kappa}_{b,w}$ with their standard errors for grouped assessments using the grouped jackknife, and the SB method after randomizing the order of assessments. The order of assessments has been randomized to remove any effects of the order of assessments on the statistics computed by the SGJK technique.

Classifiers trained on all the cases in the B3 dataset and ESCA (which does not require training) are tested on all the cases in the BR dataset. This is a test of how well classifiers trained using the B3 dataset generalize to the BR dataset. The bootstrap method using 200 replications is used to determine performance criteria and their standard errors over the cases of the BR dataset. The results are given in Table 7.39 and in Figures 7.5 and 7.6. The results for grouped assessments are given in Table 7.41 and in Figures 7.7 and 7.8. The differences among estimates of performance criteria and their standard errors are computed

Classifier	L	$SE(L)$	$\kappa_{b,w}$	$SE(\kappa_{b,w})$
ESCA	.123386	.008097	.426512	.040988
MLR	.128070	.011691	.404887	.054873
BP	.152025	.018631	.292574	.095091
C4	.289177	.022077	-.344346	.109850
COLMS	.120063	.016615	.440913	.080093
EDSS	.170854	.019697	.205321	.098605

Table 7.34: L and $\kappa_{b,w}$ for grouped assessments estimated using the RRH technique over the B3 dataset.
The values for the classifiers besides the COLMS are estimated over the test cases of the RRH technique. The values for the COLMS are estimated over the performance evaluation cases. The values are estimated using the RRH technique.

Classifier	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias$(\kappa_{b,w})$
ESCA	.124603	.008819	.000013	.427007	.043992	-.000915
MLR	.123016	.007998	.000005	.434307	.028735	-.000189
BP	.137302	.008893	-.000013	.368613	.043243	-.000901
C4	.266667	.013381	-.000006	-.226277	.055738	-.001155
COLMS	.115079	.008319	.000019	.470803	.041175	-.000843
EDSS	.151587	.009095	-.000001	.302920	.041814	-.000819

Table 7.35: L and $\kappa_{b,w}$ for grouped assessments estimated using the SGJK technique over the B3 dataset.

Classifier	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias$(\kappa_{b,w})$
ESCA	.124603	.009386	.000048	.427007	.041544	.000489
MLR	.123016	.009307	.000464	.434307	.039073	-.001300
BP	.137302	.010257	.000428	.368613	.050410	-.001568
C4	.266667	.012852	.000543	-.226277	.058020	-.001191
COLMS	.115079	.009046	.000183	.344720	.040899	.125827
EDSS	.151587	.010371	-.000035	.302920	.048543	.000822

Table 7.36: L and $\kappa_{b,w}$ for grouped assessments estimated using the SB method over the B3 dataset.
The maximum absolute value of kurtosis is 1.26. 200 bootstrap replications are used.

Classifier	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias($\kappa_{b,w}$)
ESCA	.124603	.008893	.000013	.427007	.036102	-.000290
MLR	.123016	.007227	.000005	.434307	.030510	-.000477
BP	.137302	.009392	-.000013	.368613	.042895	-.000670
C4	.266667	.011266	-.000006	-.226277	054015	-.001463
COLMS	.115079	.009140	.000018	.470803	.038506	-.000310
EDSS	.151587	.010177	-.000001	.302920	.041043	-.000403

Table 7.37: L and $\kappa_{b,w}$ for grouped, randomized assessments estimated using the SGJK technique over the B3 dataset.

Classifier	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias($\kappa_{b,w}$)
ESCA	.124603	.008142	.001207	.427007	.038609	-.007033
MLR	.123016	.008854	-.000183	.434307	.036581	-.000255
BP	.137302	.009876	.000488	.368613	.052143	-.004317
C4	.266667	.013204	.000031	-.226277	.064345	-.003330
COLMS	.115079	.008025	.000742	.470803	.038867	-.004844
EDSS	.151587	.009626	.001052	.302920	.042374	-.006431

Table 7.38: L and $\kappa_{b,w}$ for grouped, randomized assessments estimated using the SB method over the B3 dataset.
The maximum absolute value of kurtosis is 0.62. 200 bootstrap replications are used.

Classifier	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias($\kappa_{b,w}$)
ESCA	.059240	.004141	-.000378	.816664	.014252	.000004
MLR	.074986	.005512	.000464	.767932	.018906	-.002940
BP	.072652	.005473	-.000150	.775158	.018115	-.000939
C4	.261439	.009277	.000753	.190898	.032360	-.007232
COLMS	.052715	.004181	-.000316	.836858	.014159	-.000073
EDSS	.111928	.007780	-.000564	.589308	.029556	.000165

Table 7.39: L and $\kappa_{b,w}$ estimated using the bootstrap over the BR dataset. 200 replications are used. The maximum absolute value of kurtosis is 0.60.

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	-.015746	.005138	-3.064	.048732	.016195	3.009
COLMS	ESCA	-.006525	.002860	-2.281	.020194	.008894	2.270
COLMS	MLR	-.022271	.004259	-5.229	.068926	.013506	5.103
COLMS	BP	-0.020103	0.005742	-3.501	0.062567	0.017819	3.511
COLMS	EDSS	-.059213	.006722	-8.808	.247550	.024580	10.071

Table 7.40: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ computed using the bootstrap over the BR dataset.
200 replications are used. The maximum absolute value kurtosis is 0.22. $\max(|\widehat{bias}|/SE) \leq 0.48$

and the results given in Table 7.40 for ungrouped assessments and Table 7.42 for grouped assessments. The bootstrap is used because it can compute $SE(\kappa_{b,w})$.

Tables 7.43, 7.44, and 7.45 give the results of computing the difference in performance statistics of some classifiers using the RRH technique, the SGJK technique, and the SB technique, respectively. The COLMS has a value of $\hat{L}$ which is lower at the 5% significance level than the value of $\hat{L}$ for ESCA. This difference is detectable by the SB and the SGJK techniques but not by the RRH technique.

To check whether there is an effect of the ordering of the cases on the statistics computed by the grouped jackknife, we randomize the order of the assessments in the last obtained set before computing statistics for the classifiers. The statistics obtained for the randomized assessments by using Equations 2.1 and 2.4, the grouped jackknife, and the SB methods are given in Tables 7.29, 7.46, and 7.47, respectively.

Classifier	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias($\kappa_{b,w}$)
ESCA	.109502	.005387	.000317	.666207	.017454	-.000833
MLR	.129864	.006747	.000631	.604138	.018630	-.001689
BP	.115837	.006370	-.000163	.646897	.021189	.000516
C4	.280090	.009006	.000532	.146207	.027179	-.001322
COLMS	.103620	.005833	.000274	.684138	.018079	-.000741
EDSS	.174661	.008954	-.000193	.467586	.026214	.000821

Table 7.41: L and $\kappa_{b,w}$ estimated using the bootstrap over the BR dataset for grouped assessments.
200 replications are used. The maximum absolute value of kurtosis is .38.

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	-.020676	.005776	-3.57964	.062926	.017238	3.65042
COLMS	ESCA	-.005925	.003105	-1.90821	.017931	.009406	1.90634
COLMS	MLR	-.026602	.004889	-5.44119	.080948	.014218	5.69335
COLMS	BP	-.011781	.006406	-1.839	.035985	.019718	1.824
COLMS	EDSS	-.070575	.007811	-9.03533	.214990	.023149	9.28723

Table 7.42: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ computed using the bootstrap over the BR dataset for grouped assessments.
200 replications are used. The maximum absolute value of kurtosis is .50. $\max(|\widehat{bias}|/SE) \leq$ .08

Cl. 1	Cl. 2	$L_1 - L_2$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	-.009404	.009844	-0.955303	.047006	.050897	0.923551
COLMS	ESCA	-.000868	.005633	-0.154092	.004475	.028811	0.155323
COLMS	MLR	-.011416	.011260	-1.013854	.055816	.056978	0.979606
COLMS	EDSS	-.062981	.027228	-2.313097	.321655	.147486	2.180919

Table 7.43: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ computed using the RRH technique over the B3 dataset.
The differences between ESCA and MLR are estimated using the test cases of the RRH technique. The remaining differences involving the COLMS are estimated using the performance evaluation cases of the RRH technique.

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	.002778	.004480	0.620089	-.013856	.022576	-0.613749
COLMS	OLMS	-.001841	.001638	-1.1239	.009184	.008241	1.114427
COLMS	ESCA	-.007333	.003168	-2.314710	.036580	.015432	2.370399
COLMS	MLR	-.004555	.003585	-1.270572	.022724	.017336	1.310798
COLMS	EDSS	-.041365	.007894	-5.240056	.206334	.039604	5.209928

Table 7.44: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ computed using the SGJK technique over the B3 dataset.
$\max(|\widehat{bias}|/SE) \leq .02$

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	.002778	.004896	0.567402	-.013856	.025009	-0.554041
COLMS	OLMS	-.001841	.001814	-1.0148	.009184	.008916	1.0300
COLMS	ESCA	-.007333	.003409	-2.151071	.036580	.017284	2.116408
COLMS	MLR	-.004555	.003588	-1.269509	.022724	.017909	1.268859
COLMS	BP	-.027520	.006651	-4.13772	.137260	.032467	4.22767
COLMS	EDSS	-.041365	.008233	-5.024292	.206334	.041428	4.980545

Table 7.45: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ computed using the SB method for the B3 dataset.
200 bootstrap replications are used. The maximum absolute value of kurtosis is 1.91.
$\max(|\widehat{bias}|/SE) \leq .10$

Classifier	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias$(\kappa_{b,w})$
ESCA	.055603	.006362	.000007	.722644	.032599	-.000649
MLR	.052825	.005139	.000012	.736500	.025279	-.000490
BP	.075905	.006736	-.000007	.621378	.038296	-.001241
C4	.243984	.012682	.000003	-.217023	.071010	-.002928
COLMS	.048270	005537	-.000006	.759224	.027073	-.000418
EDSS	.089635	.009007	-.000003	.552890	.041679	-.000579

Table 7.46: L and $\kappa_{b,w}$ for randomized assessments estimated using the SGJK technique over the B3 dataset.

Classifier	L	$SE(L)$	bias(L)	$\kappa_{b,w}$	$SE(\kappa_{b,w})$	bias$(\kappa_{b,w})$
ESCA	.055603	.006205	.000468	.722644	.031260	-.003942
MLR	.052825	.006415	-.000010	.736500	.030268	-.001245
BP	.075905	.006673	.000325	.621378	.036296	-.004040
C4	.243984	.013462	.000387	-.217023	.075904	-.009295
COLMS	.048270	.005514	.000259	.759224	.027425	-.002637
EDSS	.089635	.008585	.001048	.552890	.044293	-.007857

Table 7.47: L and $\kappa_{b,w}$ for randomized assessments estimated using the SB technique over the B3 dataset.
The maximum absolute value of kurtosis is .69. 200 bootstrap replications are used.

The values of differences in statistics among some classifiers measured on the randomized assessments for the SGJK technique and the SB method are given in Tables 7.48 and 7.49, respectively. The values obtained from the SGJK and SB techniques are similar to each other. The values obtained from the randomized order of cases is similar to the values obtained from a non-randomized order of cases.

Table 7.50 gives the values of differences in statistics for some classifiers using the RRH technique when the assessments are grouped.

Tables 7.51 and 7.52 give the difference in $\hat{L}$ and $\hat{\kappa}_{b,w}$ and their standard errors for grouped assessments using the SGJK technique and the SB method, respectively. Comparing the values in Tables 7.50, 7.51, and 7.52, we notice that the standard error obtained by the RRH method is greater than $\sqrt{2}\times$ (standard error computed for the same parameter by the SGJK or SB methods) for $\hat{L}(ESCA) - \hat{L}(MLR)$. The standard error computed by the RRH technique for COLMS - ESCA, COLMS - EDSS is greater than $\sqrt{4} = 2\times$ (standard error computed by the SGJK or the SB techniques). This is because the standard error reported by the RRH technique has the variability incorporated of the training cases in addition to that of the test cases.

Tables 7.53 and 7.54 give the difference in statistics for grouped, randomized assessments of some classifiers using the SGJK and the SB techniques, respectively. We note here that the COLMS has a lower value of $\hat{L}$ than ESCA but not MLR at the 5% significance level.

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	.002778	.004735	0.586695	-.013856	.023760	-0.583165
COLMS	ESCA	-.007333	.003412	-2.149179	.036580	.017449	2.096395
COLMS	MLR	-.004555	.003719	-1.224792	.022724	.018642	1.218968
COLMS	BP	-0.027635	0.006986	-3.955	0.137846	0.032937	4.185
COLMS	EDSS	-.041365	.007265	-5.693737	.206334	.034782	5.932206

Table 7.48: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for randomized assessments computed using the SGJK technique over the B3 dataset.
$\max(|\widehat{bias}|/SE) \leq .02$

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	.002778	.004963	0.559742	-.013856	.025158	-0.550759
COLMS	ESCA	-.007333	.003616	-2.027931	.036580	.018255	2.003835
COLMS	MLR	-.004555	.003374	-1.350030	.022724	.016670	1.363167
COLMS	EDSS	-.041365	.010858	-3.809633	.206334	.053721	3.840844

200 bootstrap replications are used. The maximum absolute value of kurtosis is 0.48.
$\max(|\widehat{bias}|/SE) \leq .10$

Table 7.49: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers computed using the SB method for randomized assessments in the B3 dataset.

Criteria for the OLMS are not computed since the OLMS was introduced later in the research, after the main body of the research was completed.

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi))$
ESCA	MLR	-.004684	.010755	-0.435	.021625	.049970	0.432
COLMS	ESCA	-.001582	.010836	-0.145	.007358	.049703	0.148
COLMS	MLR	-.007975	.015729	-0.507	.036429	.071383	0.510
COLMS	BP	-0.011781	0.006406	-1.839	0.035985	0.019718	1.824
COLMS	EDSS	-.052278	.025583	-2.043	.244760	.120772	2.026

Table 7.50: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for grouped assessments computed using the RRH technique over the B3 dataset.
The differences between ESCA and MLR are estimated using the test cases of the RRH technique. The remaining differences involving the COLMS are estimated using the performance evaluation cases of the RRH technique.

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	.001587	.007778	0.204037	-.007300	.036000	-0.202778
COLMS	ESCA	-.009524	.003128	-3.044757	.043796	.014545	3.011069
COLMS	MLR	-.007937	.007071	-1.122472	.036496	.031855	1.145691
COLMS	EDSS	-.036508	.007956	-4.588738	.167883	.035464	4.733899

Table 7.51: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for grouped assessments computed using the SGJK technique over the B3 dataset.
$\max(|\widehat{bias}|/SE) \leq .16$

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	.001587	.007222	0.219745	-.007300	.033386	-0.218655
COLMS	ESCA	-.009524	.003718	-2.561592	.043796	.016912	2.589640
COLMS	MLR	-.007937	.006121	-1.296684	.036496	.027847	1.310590
COLMS	EDSS	-.036508	.006953	-5.250683	.167883	.032514	5.163407

Table 7.52: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for grouped assessments computed using the SB method over the B3 dataset.
200 bootstrap replications are used. The maximum absolute value of kurtosis is 0.81.
$\max(|\widehat{bias}|/SE) \leq .06$

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	.001587	.005463	0.290500	-.007300	.025109	-0.290732
COLMS	ESCA	-.009524	.004037	-2.359178	.043796	.018469	2.371325
COLMS	MLR	-.007937	.005737	-1.383476	.036496	.026541	1.375080
COLMS	BP	-.022223	.010146	-2.190	.102190	.046350	2.204
COLMS	EDSS	-.036508	.006800	-5.368824	.167883	.031182	5.383972

Table 7.53: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for grouped, randomized assessments computed using the SGJK technique over the B3 dataset. $\max(|\widehat{bias}|/SE) \leq 0.19$

Cl. 1	Cl. 2	$\hat{\theta}$	$SE(\theta)$	$z(\theta)$	$\hat{\phi}$	$SE(\phi)$	$z(\phi)$
ESCA	MLR	.001587	.006892	0.230267	-.007300	.031997	-0.228146
COLMS	ESCA	-.009524	.003661	-2.601475	.043796	.016658	2.629127
COLMS	MLR	.007937	.005791	1.370575	.036496	.026287	1.388367
COLMS	EDSS	-.036508	.006903	-5.288715	.167883	.030641	5.479031

The maximum absolute value of kurtosis is 0.77. 200 bootstrap replications are used. $\max(|\widehat{bias}|/SE) \leq .21$

Table 7.54: Differences $\hat{\theta} = \hat{L}_1 - \hat{L}_2$ and $\hat{\phi} = \hat{\kappa}_{b,w,1} - \hat{\kappa}_{b,w,2}$ among classifiers for grouped, randomized assessments computed using the SB technique over the B3 dataset.

7.5 Interpretation of performance evaluation results

For each classifier, we see that L and $\kappa_{b,w}$ estimated by any two evaluation techniques are within one standard error of each other. One can use theoretical results (Equations 2.1, 2.4) to compute $\hat{L}$, $SE(L)$, and $\hat{\kappa}_{b,w}$ for a classifier that does not need training. The bootstrap (or the grouped jackknife) is the preferred technique to compute bias($\hat{L}$), $SE(\kappa_{b,w})$, and bias($\hat{\kappa}_{b,w}$) when the classifier does not require training. When the classifier requires training, the SB technique and the SGJK technique are superior to using the RRH technique, because they do not require splitting the sample set into half for training and half for testing. The consequences of this effect are noticed when the RRH technique cannot detect the difference in $\hat{L}$ between ESCA and the COLMS at the 5% significance level, but the SGJK technique and the SB technique can.

Fukunaga predicts theoretically [27, 28] that reducing the size of the training set will increase the estimate of L for a classifier, while reducing the size of the test set will increase the estimated standard error of $\hat{L}$. This effect can be noticed when comparing the results of the RRH technique with the SB or SGJK techniques. The rest of this section uses the values estimated by the SB technique because the SB technique approximates the value computed by the multinomial results if they were available. The classifiers both in ascending order of $\hat{L}$ and descending order for $\hat{\kappa}_{b,w}$ for ungrouped assessments are the COLMS, MLR, ESCA, BP, EDSS, and C4, if one does not take into consideration the effects of multiple comparisons.

When doing pairwise comparisons, the COLMS is better than ESCA at the 5% significance level even after Bonferronizing with a multiplier of 4. The COLMS is better than MLR but not at the 5% significance level. The EDSS is worse than all but one of its constituent classifiers. This failure of the EDSS can be explained when considering that unlike the COLMS, the combination scheme in the EDSS fails to consider each classifier's accuracy in classifying cases from different classes. Another possible reason is that Dr. David. Hunter designed the combination system with a human user in mind, who could change his assessments. In our case, a computer based classifier is the most trusted classifier (with Dr. David Hunter's approval) that cannot change its assessments but must reject the case.

7.6 Results of perturbation-testing

Perturbation-testing is done by training the classifiers MLR, BP, and C4 (ESCA does not need training) on 314 cases of the B3 dataset and doing perturbation-testing over the remaining 316 test cases. Only the cases that were successfully perturbed are used when computing the perturbation scores.

To compute the scores, a selector function f is defined that selects only those cases that were successfully perturbed by the perturbation method under consideration. Let

$v_{ij(M)}$ = expert-defined points for perturbation method M if the original assessment is i and the perturbed assessment is j,

$\hat{p}_{ij,(k,M)}$ = estimated probability that the original assessment is ω_i and the perturbed assessment is ω_j for classifier C_k and perturbation method M.

Then,

$$
\begin{aligned}
V_{(k,M)} &= \sum_{i=1}^{C}\sum_{j=1}^{C} \hat{p}_{ij,(k,M)} v_{ij(M)}, \qquad (7.1)\\
&\quad \forall M \in \{PSL, PSH, PL, PH\}\\
&\quad \forall k \in \{ESCA, MLR, BP, C4\}\\
V_{k,sum} &= V_{k,PSL} + V_{k,PSH} + V_{k,PL} + V_{k,PH}
\end{aligned}
$$

PSL, PSH, PL, and PH are the various perturbation methods, viz., perturb slightly low, perturb slightly high, perturb low, and perturb high, respectively. These methods are defined in Chapter 6 for the AEPOSCH dataset.

The perturbation scores and their standard errors are obtained using the bootstrap method. 200 replications are used. Tables 7.55 to 7.70 give the results of perturbation tests for the individual classifiers. The scores for the perturbation tests for the different classifiers are given in Table 7.71.

The difference $\hat{V}_{MLR,sum} - \hat{V}_{ESCA,sum}$ is estimated by the bootstrap using 200 replications. The results are given in Table 7.71. Looking at the table, we see that $\hat{V}_{MLR,sum} - \hat{V}_{ESCA,sum} > 0$ at the 5% significance level, even after taking into consideration the Bonferroni effect. MLR does better in perturbation-testing over the B3 dataset than ESCA.

Table 7.55 gives the results of ESCA on the perturb slightly low (PSL) test. ESCA passes the PSL test, as almost all the perturbed assessments are 0, 1, or 2 categories below (worse than) the original assessment.

Orig. $\downarrow$\Pert. $\rightarrow$	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	5	0	0	0	0	0	0	0	5
-2	4	2	0	0	0	0	0	0	6
-1	3	9	15	0	0	0	0	0	27
-0	1	3	27	44	0	0	0	0	75
+0	0	0	0	61	3	0	0	0	64
+1	0	0	0	8	6	1	0	0	15
+2	0	0	0	0	0	2	3	0	5
+3	0	0	0	0	0	1	1	0	2
Sum	13	14	42	113	9	4	4	0	199

Table 7.55: Results of Perturb Slightly Low (PSL) – ESCA.
The original assessments are in rows and the perturbed assessments are in columns.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	2	2	1	0	0	0	0	0	5
-2	0	0	3	3	0	0	0	0	6
-1	0	0	1	14	12	0	0	0	27
-0	0	0	0	7	15	52	0	0	74
+0	0	0	0	0	0	45	21	0	66
+1	0	0	0	0	0	1	14	1	16
+2	0	0	0	0	0	0	1	5	6
+3	0	0	0	0	0	0	0	2	2
Sum	2	2	5	24	27	98	36	8	202

Table 7.56: Results for Perturb Slightly High (PSH) – ESCA.
The original assessments are in rows and the perturbed assessments are in columns.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	0	0	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0	0	0
-1	16	0	0	0	0	0	0	0	16
-0	42	2	0	0	0	0	0	0	44
+0	22	8	1	0	0	0	0	0	31
+1	1	5	4	0	0	0	0	0	10
+2	0	7	0	0	0	0	0	0	7
+3	0	1	0	0	0	0	0	0	1
Sum	81	23	5	0	0	0	0	0	109

Table 7.57: Results for Perturb Low (PL) – ESCA.
The original assessments are in rows and the perturbed assessments are in columns.

Table 7.56 gives the results of ESCA on the perturb slightly high (PSH) test. ESCA passes the PSH test, as almost all the perturbed assessments are 0, 1, or 2 categories above (better than) the original assessment.

Table 7.57 gives the results of ESCA on the perturb low (PL) test. The results seem to indicate that ESCA is aggressive in shifting the assessments towards the lower (or worse) side. This is because several perturbed assessments fall in the -3 category, even when the original assessment is as high as +0.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	0	0	0	1	0	0	0	0	1
-2	0	0	0	1	0	3	0	0	4
-1	0	0	0	0	0	2	6	5	13
-0	0	0	0	0	0	11	73	5	89
+0	0	0	0	0	0	3	16	27	46
+1	0	0	0	0	0	0	0	12	12
+2	0	0	0	0	0	0	0	4	4
+3	0	0	0	0	0	0	0	1	1
Sum	0	0	0	2	0	19	95	54	170

Table 7.58: Results for Perturb High (PH) – ESCA.
The original assessments are in rows and the perturbed assessments are in columns.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	5	0	0	0	0	0	0	0	5
-2	9	0	0	0	0	0	0	0	9
-1	3	22	0	0	0	0	0	0	25
-0	0	11	51	1	0	0	0	0	63
+0	0	0	20	55	0	0	0	0	75
+1	0	0	0	5	11	0	0	0	16
+2	0	0	0	0	2	3	0	0	5
+3	0	0	0	0	0	0	1	0	1
Sum	17	33	71	61	13	3	1	0	199

Table 7.59: Results for Perturb Slightly Low (PSL) – MLR.
The original assessments are in rows and the perturbed assessments are in columns.

Table 7.58 gives the results of ESCA on the perturb high (PH) test. The results seem to indicate that ESCA is aggressive in shifting the assessments towards the higher (or better) side. This is because several perturbed assessments fall in the +2 category, even when the original assessment is as low as -0. However, ESCA is not as aggressive on the perturb high test as it was on the perturb low test. This is because in perturb low it shifted a very large fraction of assessments to the lowest extreme (-3), while in perturb high it did not shift too many cases to the highest extreme (+3) but shifted them to the +2 category instead.

Table 7.59 gives the results of MLR on the perturb slightly low (PSL) test. MLR passes the PSL test, as almost all the perturbed assessments are 0, 1, or 2 categories below (worse than) the original assessment.

Table 7.60 gives the results of MLR on the perturb slightly high test. MLR passes the PSH test, as almost all the perturbed assessments are 0, 1, or 2 categories above (better than) the original assessment.

Table 7.61 gives the results of MLR on the perturb low test. The results seem to indicate

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	2	1	2	0	0	0	0	0	5
-2	0	0	5	4	0	0	0	0	9
-1	0	0	0	17	8	0	0	0	25
-0	0	0	0	0	44	21	0	0	65
+0	0	0	0	0	1	63	11	0	75
+1	0	0	0	0	0	0	15	1	16
+2	0	0	0	0	0	0	0	6	6
+3	0	0	0	0	0	0	0	1	1
Sum	2	1	7	21	53	84	26	8	202

Table 7.60: Results for Perturb Slightly High (PSH) – MLR.
The original assessments are in rows and the perturbed assessments are in columns.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	2	0	0	0	0	0	0	0	2
-2	7	0	0	0	0	0	0	0	7
-1	15	0	0	0	0	0	0	0	15
-0	34	3	0	0	0	0	0	0	37
+0	13	24	1	0	0	0	0	0	38
+1	0	4	4	0	0	0	0	0	8
+2	0	0	1	3	0	0	0	0	4
+3	0	0	0	0	0	0	0	0	0
Sum	71	31	6	3	0	0	0	0	111

Table 7.61: Results for Perturb Low (PL) – MLR.
The original assessments are in rows and the perturbed assessments are in columns.

that MLR is aggressive in shifting the assessments towards the lower (or worse) side. This is because several perturbed assessments fall in the -3 category, even when the original assessment is as high as +0. However, MLR is not as aggressive as ESCA on the perturb low test.

Table 7.62 gives the results of MLR on the perturb high test. The results seem to indicate that MLR is aggressive in shifting the assessments towards the higher (or better) side. This is because several perturbed assessments fall in the +3 category, even when the original assessment is as low as +0.

Table 7.63 gives the results of BP on the perturb slightly low test. BP is too aggressive in changing assessments towards the lower (or worse) side. The assessment of several cases is shifted to two categories below their original assessment.

Table 7.64 gives the results of BP on the perturb slightly high test. BP is too aggressive in changing assessments towards the higher (or better) side. The assessment of several cases is shifted to two categories above their original assessment.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	0	0	1	1	1	0	0	0	3
-2	0	0	0	0	6	0	0	0	6
-1	0	0	0	0	0	13	0	0	13
-0	0	0	0	0	0	0	23	13	36
+0	0	0	0	0	0	0	5	38	43
+1	0	0	0	0	0	0	0	11	11
+2	0	0	0	0	0	0	0	4	4
+3	0	0	0	0	0	0	0	0	0
Sum	0	0	1	1	7	13	28	66	116

Table 7.62: Results for Perturb High (PH) – MLR.
The original assessments are in rows and the perturbed assessments are in columns.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	0	0	0	0	0	0	0	0	0
-2	1	28	0	0	0	0	0	0	29
-1	0	29	5	0	0	0	0	0	34
-0	0	20	25	19	0	0	0	0	64
+0	0	3	27	7	1	0	0	0	38
+1	0	0	10	12	2	1	0	0	25
+2	0	0	0	0	3	3	3	0	9
+3	0	0	0	0	0	0	0	0	0
Sum	1	80	67	38	6	4	3	0	199

Table 7.63: Results for Perturb Slightly Low (PSL) – BP.
The original assessments are in rows and the perturbed assessments are in columns.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	0	0	0	0	0	0	0	0	0
-2	0	8	8	12	1	0	0	0	29
-1	0	0	3	8	15	9	0	0	35
-0	0	0	0	15	16	29	5	0	65
+0	0	0	0	0	6	14	18	0	38
+1	0	0	0	0	0	0	22	3	25
+2	0	0	0	0	0	0	7	2	9
+3	0	0	0	0	0	0	0	1	1
Sum	0	8	11	35	38	52	52	6	202

Table 7.64: Results for Perturb Slightly High (PSH) – BP.
The original assessments are in rows and the perturbed assessments are in columns.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	0	0	0	0	0	0	0	0	0
-2	0	17	0	0	0	0	0	0	17
-1	0	20	3	0	0	0	0	0	23
-0	1	27	8	0	0	0	0	0	36
+0	0	18	3	0	0	0	0	0	21
+1	0	9	2	0	0	0	0	0	11
+2	0	1	3	0	1	0	0	0	5
+3	0	0	0	0	0	0	0	0	0
Sum	1	92	19	0	1	0	0	0	113

Table 7.65: Results for Perturb Low (PL) – BP.
The original assessments are in rows and the perturbed assessments are in columns.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	0	0	0	0	0	0	0	0	0
-2	0	0	0	0	5	13	16	0	34
-1	0	0	0	2	5	15	17	3	42
-0	0	0	0	9	16	28	33	4	90
+0	0	0	0	0	10	22	26	3	61
+1	0	0	0	0	0	7	21	2	30
+2	0	0	0	0	0	5	7	0	12
+3	0	0	0	0	0	0	1	1	2
Sum	0	0	0	11	36	90	121	13	271

Table 7.66: Results for Perturb High (PH) – BP.
The original assessments are in rows and the perturbed assessments are in columns.

Table 7.65 gives the results of BP on the perturb low test. BP does not seem to respond well to this test, as almost all the cases' assessments are shifted to -2.

Table 7.66 gives the results of BP on the perturb high test. BP seems to respond better on this test than ESCA and MLR because the assessments of perturbed cases are well distributed in the +0 to +3 categories.

As we shall see, C4 does poorly on all the perturbation tests where it does not change its assessment in response to the perturbation. The fact that C4 does not do well can be explained by considering that C4 is based on a discriminant tree while the task requires all arterial segments to be considered and given approximately equal weightage. C4 presumably produces a discriminant tree based on one set of arteries during training, but these arteries need not be the ones whose stenoses are perturbed. Therefore C4 does not detect the change and does not respond to the perturbation.

Table 7.67 gives the results of C4 on the perturb slightly low test. C4 fails the PSL test, as almost all the assessments are unchanged after perturbation.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	4	0	0	0	0	0	0	0	4
-2	0	16	1	0	0	0	0	0	17
-1	0	3	36	2	0	0	0	0	41
-0	0	5	11	33	0	0	0	0	49
+0	0	1	5	12	30	0	0	0	48
+1	0	1	2	3	0	26	0	0	32
+2	0	0	0	0	0	0	6	0	6
+3	0	0	0	0	0	0	0	2	2
Sum	4	26	55	50	30	26	6	2	199

Table 7.67: Results for Perturb Slightly Low (PSL) – C4.
The original assessments are in rows and the perturbed assessments are in columns.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	4	0	0	0	0	0	0	0	4
-2	0	15	0	2	1	0	0	0	18
-1	0	0	28	7	4	1	0	0	40
-0	0	0	0	43	6	1	0	0	50
+0	0	0	0	2	46	0	0	0	48
+1	0	0	0	0	0	34	0	0	34
+2	0	0	0	0	2	0	4	0	6
+3	0	0	0	0	0	0	0	2	2
Sum	4	15	28	54	59	36	4	2	202

Table 7.68: Results for Perturb Slightly High (PSH) – C4.
The original assessments are in rows and the perturbed assessments are in columns.

Table 7.68 gives the results of C4 on the perturb slightly high test. C4 fails the PSH test as almost all the assessments are unchanged after perturbation.

Table 7.69 gives the results of C4 on the perturb low test. Many assessments are unchanged, which is highly unacceptable.

Table 7.70 gives the results of C4 on the perturb high test. Many assessments are unchanged, which is highly unacceptable.

In Table 7.72, we use $P(SUM)$ for the comparison on the entire perturbation test for the various classifiers. We see that MLR does better than ESCA, although the difference is not significant at the 5% significance level, even before the effect of multiple comparisons on statistical significance is taken into account. On PSL and PH, MLR is not significantly better than ESCA at the 5% significance level. On PSH and PL, MLR is significantly better than ESCA at the 5% significance level, even after taking into consideration the effect of multiple comparisons.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	0	0	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0	0	0
-1	0	5	18	0	0	0	0	0	23
-0	0	7	13	7	1	0	0	0	28
+0	0	8	9	3	9	0	0	0	29
+1	0	2	2	1	0	13	0	0	18
+2	0	1	2	0	0	0	4	0	7
+3	0	0	0	0	0	0	0	0	0
Sum	0	23	44	11	10	13	4	0	105

Table 7.69: Results for Perturb Low (PL) – C4.
The original assessments are in rows and the perturbed assessments are in columns.

Orig. ↓\Pert. →	-3	-2	-1	-0	+0	+1	+2	+3	Sum
-3	0	0	0	0	0	0	0	0	0
-2	0	8	0	0	1	0	0	0	9
-1	0	0	12	6	7	0	0	0	25
-0	0	0	2	48	21	4	0	0	75
+0	0	0	0	0	29	2	1	0	32
+1	0	0	2	0	2	14	0	0	18
+2	0	0	0	0	2	0	2	0	4
+3	0	0	0	0	0	0	0	0	0
Sum	0	8	16	54	62	20	3	0	163

Table 7.70: Results for Perturb High (PH) – C4.
The original assessments are in rows and the perturbed assessments are in columns.

M →	PSL	PSH	PL	PH	$P(SUM)$
$\hat{V}(ESCA, M)$	.883	.763	.353	.782	2.782
$SE(V(ESCA, M))$	.010	.015	.104	.038	.105
$\hat{V}(MLR, M)$	.890	.879	.747	.771	3.289
$SE(V(MLR, M))$	.013	.013	.027	.025	.046
$\hat{V}(BP, M)$	.721	.695	.553	.214	2.184
$SE(V(BP, M))$	.019	.021	.083	.073	.121
$\hat{V}(C4, M)$	.760	.753	-.376	-1.567	-.428
$SE(V(C4, M))$	.019	.021	.128	.148	.200
$\hat{V}(MLR, M) - \hat{V}(ESCA, M)$	.007	.116	.394	-.010	.507
$SE(V(MLR, M) - V(ESCA, M))$	.016	.021	.112	.049	.120

Table 7.71: Perturbation scores $\hat{V}$ and $SE(V)$ for classifiers on the B3 dataset. The difference in perturbation scores between MLR and ESCA is shown. The maximum absolute values of bias and kurtosis are .03 and .73 respectively.

7.6.1 Expert opinion on perturbation results and interpretation of statistical results

The perturbation results of MLR and ESCA were compared by the expert, Dr. David Hunter. According to him, MLR looks much better on PSL than ESCA. This is because ESCA has too many cases where it does not make any change and there are more changes by -3 categories while in MLR almost all changes are by -1 or -2 categories.

On PSH, MLR has fewer cases than ESCA where there are no changes. MLR has some more cases changed by +1 category with no abnormally high counts of changes by +3 or more categories for either classifier. MLR is just slightly better.

In comparing the performance of the classifiers on PH, MLR is slightly better than ESCA because it has fewer changes that are by +1 categories and it has no cases where there was no change. ESCA has one case where there is no change and other than that there is not much difference between the classifiers.

In comparing the performance of ESCA and MLR on PL, MLR had two cases that were unchanged and seven changes that changed by only one category. ESCA has a bad fault of its own. ESCA had nine changes that were by -5 or -6 categories, which is large. So ESCA is too aggressive in changing its assessments towards worse, while MLR is not aggressive enough. The two classifiers have their flaws and weaknesses but are both usable in the study.

Overall, Dr. David Hunter stated that MLR does better than ESCA on perturbation-testing. This view is supported by the comparison of $\hat{V}_{MLR,sum} > \hat{V}_{ESCA,sum}$ Bonferroni p-value = 0.0001849 $<$ .05 with 6 paired comparisons.

7.6.2 Experts' opinion whether MLR is usable instead of ESCA

Dr. David Hunter, our collaborating expert, reviewed the performance of MLR and ESCA on the B3 and BR datasets. The performance of MLR is slightly poorer than ESCA on the

M →	PSL	PSH	PL	PH	$P(SUM)$
$\hat{V}(MLR,M) - \hat{V}(ESCA,M)$	.0070	.1163	.3945	-.0108	.5071
$SE(V(MLR,M) - V(ESCA,M))$	.0165	.0214	.1129	.0490	.1208
$z(V(MLR,M) - V(ESCA,M))$	.4257	5.414	3.493	-0.2203	4.198
$p(V(MLR,M) - V(ESCA,M))$	.6705	8.7e-8	.0005	.8250	3.0e-05
$6 \times p(V(MLR,M) - V(ESCA,M))$	> 0.5	5.2e-7	0.0061	> 0.5	0.0001849

Table 7.72: Bonferroni values for differences in perturbation scores $\hat{V}$ on the B3 dataset. The difference in perturbation scores between MLR and ESCA is shown. The maximum absolute values of bias and kurtosis are .03 and .73 respectively. There are $4 \times 3/2 = 6$ possible comparisons since there are 4 classifiers viz. ESCA, MLR, BP, and C4. Therefore a Bonferroni multiplier of 6 is used.

BR dataset but Dr. Hunter stated that this is not a cause for concern. He reviewed the results of perturbation-testing. He stated that MLR can be used to evaluate cases for the AEPOSCH application.

A second expert, Dr. Carl White[2], was asked his opinion on the system and the results. He said that he agreed with Dr. David Hunter's opinion. He further stated that the evaluation of the expert system and the classifiers characterized the algorithm that Dr. David Hunter uses very well against a test he provides.

Expert systems enable one to characterize one's thinking/intuition processes in an executable model. The model can be studied by other researchers, and the results of different models compared. This can lead to a better definition and understanding of changing evaluation procedures.

7.7 Summary

In this chapter we show how to compute values and differences of values for $\hat{L}, SE(L), \hat{\kappa}_{b,w}$, and $SE(\kappa_{b,w})$ for real-world data. Various classifiers are evaluated using the RRH technique, the multinomial technique, the bootstrap, and the grouped jackknife when training of the classifier is not needed. When training of the classifier is needed the $SM, SGJK, SB$, and RRH techniques can be used. The $SM, SGJK$, and the SB techniques are based on the SR conjecture introduced in Section 4.6.3. The SR conjecture is needed because the procedure suggested in the literature [17, 19] to estimate Expected Excess Error in the bootstrap is proved incorrect (Section 4.5.2). The SR conjecture is shown plausible in Chapter 4 but its correctness is not proven. The SM, SB, and SGJK techniques compute $\hat{L}$, $\hat{\kappa}_{b,w}$ for classifiers, and differences of $\hat{L}$, $\hat{\kappa}_{b,w}$ for classifiers within one standard error of each other. The standard errors computed by the SM, SB, and SGJK methods for $\hat{L}$, $\hat{\kappa}_{b,w}$, differences of $\hat{L}$, $\hat{\kappa}_{b,w}$ between classifiers, are within 10% of each other. The bias values reported are

[2] Dr. Carl White is the principal investigator of the angiographic reading center of the post coronary artery bypass surgery study, another large National Heart, Blood and Lung Institute (NHLBI) sponsored angiographic trial. He is also the director of clinical cardiology at the University of Minnesota.

negligible and are less than 10% of the reported values.

The RRH technique computes standard errors that are slightly different from the SB, SM, and SGJK techniques. This difference is explainable because the size of the sample set used for testing in the RRH technique is half that used in the SM, SB, and SGJK techniques. The variability in the training set for the RRH technique is greater than the variability in the training set for the SM, SB, and SGJK techniques. This is because the training set consists of half the cases drawn at random each time for the RRH technique, while for the SM, SB, and SGJK techniques only $2 \times h$ cases are changed each time (recall here that $n - h$ is the number of cases used for training in the SM, SB, and SGJK techniques when the sample is of size n). This higher variability of the training set in the RRH technique contributes to a higher standard error for the RRH technique. The smaller size of the test set further contributes to a higher standard error for the RRH technique. Partly because of this higher standard error, the RRH technique cannot detect the difference in $\hat{L}$ between ESCA and the COLMS at the 5% significance level, but the SGJK method and the SB method can.

Perturbation scores for the methods showed that MLR is significantly better at the 5% significance level for the perturbation tests, perturb low (PL) and perturb high (PH) even after Bonferronizing with a multiplier of 6. For the perturbation tests, perturb slightly low (PSL) and sum of perturbation tests $P(SUM)$, MLR is better than ESCA but not at the 5% significance level.

The experts reviewed the cross-classification tables, the values of $\hat{L}, SE(L), \hat{\kappa}_{b,w}, SE(\kappa_{b,w})$ and the perturbation tests. They concluded that MLR can be used in the study instead of ESCA. In Dr. Carl White's opinion, the evaluated criteria and the perturbation measures did a very good job of characterizing the various classifiers.

Chapter 8

Summary and Future Work

Classifiers are essential in all walks of life. Without classification almost all real-world tasks cannot be performed, as it is not possible to recognize whether one has what one desires. Classification is necessary whether it is the acquisition of food for an amoeba/human being, or rejecting/accepting an item in an assembly line by a machine/industrial robot, or the recognition of disease by a doctor/expert system. Classification is a central task in all intelligent systems and is needed for most artificial intelligence systems. Classification algorithms and systems are therefore likely to be used in almost all walks of human life.

Estimation of appropriate performance criteria and their standard errors for classifiers is necessary so that competing classifiers can be evaluated and compared. These classifiers can be expert systems, algorithms, or a hybrid of the two (for instance, using a combination strategy such as the COLMS). A rigorous evaluation is necessary so that the prejudice against algorithmic classifiers can be overcome and algorithms used with or instead of expert systems. The use of rigorous testing and evaluating strategies can permit the use of algorithms and classifiers in fields where they previously were thought inappropriate, because such testing can help convince the expert that a classifier is appropriate. Perturbation techniques and the use of statistical techniques for evaluating the results of perturbation techniques are an important contribution of this thesis.

Two performance criteria are introduced for classifiers, whether human, expert system, algorithmic, or something else. These criteria use an expert-defined loss function to assign different losses to reflect different degrees of disagreement. The first criterion, L, is a weighted loss criterion giving a weighted index of mis-classification that takes a value between 0 and 1 indicating the seriousness of disagreement (with 1 indicating maximum disagreement) between the classifiers and the gold standard. The second criterion, $\kappa_{b,w}$, is an index of validity of agreement between the classifier and the gold standard, that is, agreement adjusted for agreement because of chance. $\kappa_{b,w}$ is an important criterion, especially in fields where there is little prior knowledge of what acceptable values of L are. By estimating $\kappa_{b,w}$, the experimenter knows whether the classifier is doing better than a best strategy guessing based on a priori knowledge of the probabilities of the classes.

Results based on the multinomial distribution are preferred to estimate $\hat{L}$, $\hat{\sigma}_{\hat{L}}$, and $\hat{\kappa}_{b,w}$ when the classifier does not require training. The SR conjecture is introduced for use when the classifier requires training and its plausibility is demonstrated by a simple experiment. The SM, SB, and SGJK techniques are developed using the SR conjecture. The RRH technique is developed independently of the SR conjecture. When the classifier

does not require training, the multinomial results are preferred to estimate $\hat{L}$, $\hat{\sigma}_{\hat{L}}$, and $\hat{\kappa}_{b,w}$. When the classifier does not require training, the bootstrap method can be used to estimate $SE(\kappa_{b,w})$. The results of the bootstrap technique approach those of the multinomial results as the number of replications is increased. When the classifier requires training, the SM method is preferred to estimate $\hat{L}$, $\hat{\sigma}_{\hat{L}}$, and $\hat{\kappa}_{b,w}$. The SM method cannot be used to estimate $SE(\kappa_{b,w})$ as the multinomial results are not known. When the classifier requires training, the RRH technique, the SGJK technique, and the SB technique can be used to estimate $SE(\kappa_{b,w})$. The SB technique can be used to estimate $\hat{L}$, $SE(L)$, $\hat{\kappa}_{b,w}$, and $SE(\hat{\kappa}_{b,w})$ and it approaches the multinomial results (if they were known). The RRH technique does not use all the available samples as thoroughly as the SM, SB, and SGJK techniques and therefore computes larger standard errors than the SM, SB, and SGJK techniques. The SM technique is the preferred technique if the multinomial results are available, as they are for $L, SE(L)$, and $\kappa_{b,w}$. The results of the grouped jackknife (and hence the SGJK) technique are not guaranteed to approach the multinomial results for an arbitrary non-linear criterion such as $\kappa_{b,w}$. For these reasons the SB technique is preferred over the RRH technique and the SGJK technique if the multinomial results are not available.

The bootstrap, the grouped jackknife, the RRH technique, the SGJK technique, and the SB technique can be used for computing numerical values (and their standard errors) for functions including sums and differences of criteria, including cases where the criteria are not computed from independent samples. In such cases it is difficult to derive the expression from the multinomial approximation.

The Classifier-based Overall Loss Minimization Strategy (COLMS) is introduced to combine assessments of classifiers to reduce the weighted loss L. The COLMS is compared with the individual constituent classifiers (ESCA and MLR) and the Expert-Defined Domain-Specific Strategy (EDSS) which also combines classifiers. The COLMS outperforms one of its constituent classifiers (ESCA) in terms of lower $\hat{L}$ and higher $\hat{\kappa}_{b,w}$ at the 5% significance level during pairwise comparisons and even after Bonferroni comparisons using a multiplier of 6. It outperforms the other classifier (MLR) in terms of lower $\hat{L}$ and higher $\hat{\kappa}_{b,w}$ but the difference is not significant at the 5% significance level during pairwise comparisons. The COLMS outperforms the EDSS at the 5% significance level. A variant of the COLMS, the OLMS is introduced. The OLMS may be important, as it enables partitioning (clustering) algorithms to be used where the clusters need not be the output classes alone. This is useful for using clustering and partitioning algorithms from fields such as neural networks. The use of the OLMS is demonstrated and the results compared with the COLMS. The COLMS has a lower value of L than the OLMS although the difference is not significant at the 5% significance level. In general, this is not surprising because unless all of the classifiers are poor, it is likely that at least one of them should select the class with the minimum risk. In our case, we have two good classifiers being combined by the COLMS or the OLMS. This shows that the OLMS need not do better than the COLMS in real-world situations. The EDSS is worse than all but one of its constituent classifiers. This failure of the EDSS can be explained when considering that unlike the COLMS, the combination scheme in the EDSS fails to consider each classifier's accuracy in classifying cases from different classes. Another possible reason is that Dr. David Hunter designed the combination system with a human user in mind, who could change his assessments. In our case, a computer based classifier is the most trusted classifier (with Dr. David Hunter's approval) that cannot change its assessments and therefore the EDSS must reject the case.

A perturbation-testing strategy is introduced as an example of an Expert-Defined Test (EDT). The perturbation-testing strategy tests the classifier using domain-specific perturbation tests. A good performance under an EDT can further increase confidence of the expert(s) in a classifier.

One of the algorithmic classifiers, multiple linear regression (MLR), had values of $\hat{L}$ as computed by the preferred SB method that were lower than that of ESCA, but the difference was not significant at the 5% significance level. MLR was better than ESCA on the results of the perturbation-testing at the 5% significance level. MLR was worse than ESCA on a different population of cases than the one it was trained on but the difference was not significant at the 5% significance level. This can be argued as a case for choosing an expert system over an algorithmic classifier because it appears possible (as in our case) that an expert system may generalize better to different populations. In fact, an expert system might have in built conditions to handle different populations. With these results, the domain expert was sufficiently confident with MLR to consider it suitable for use.

MLR is simpler than ESCA and easier to design. A classifier based on multiple linear regression can be obtained by simply training an off-the-shelf algorithm available in a package on the given dataset. Purely on the basis of statistical comparisons, the effort to develop ESCA was unnecessary, since the purely algorithmic MLR performed well on statistical comparisons and in the view of the domain expert. The domain expert was confident that MLR could be used.

Without the rigorous evaluation of MLR and ESCA with respect to the experts' own agreement with each other as demonstrated by an analysis of the quality-control data there was little way to demonstrate MLR's (or ESCA's) desirability. It should be noted here that the scope of the methods based on the split recombination procedure is restricted to the classifiers studied here, i.e. the idealistic classifier separating the cases into one of two normal populations and the more complex classifiers used on the B3 dataset. Future work may lead others to other techniques.

Future work should focus on determining the validity of the SR conjecture. Extracting domain-specific commonalities across EDTs is another possible area of future research. An effort to determine performance criteria and techniques to evaluate them for tasks other than classification in the same way that L and $\kappa_{b,w}$ are defined for classifiers should be fruitful. An example of such a task is the obtaining of economic forecasts. We can obtain economic forecasts using simple and widely used algorithms such as in time-series analysis, complex, specialized economic models, or experts. A comparison of these methods can demonstrate the effectiveness of simple methods such as time series analysis and lead to their greater use. Different methods can again be combined by defining a strategy such as the COLMS.

Bibliography

[1] R. C. Dubes A. K. Jain and C. Chen. Bootstrap Techniques for Error Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):628–623, 1987.

[2] A. Agresti. A Model for Agreement Between Ratings on an Ordinal Scale. *Biometrics*, 44:539–548, June 1988.

[3] A. Barr, P. R. Cohen, and E. A. Feigenbaum. *The Handbook of Artificial Intelligence*, volume 4. Addison-Wesley Publishing Company, Reading, MA, 1989.

[4] Y. M. M. Bishop, S. E. Feinberg, and P. W. Holland. *Discrete Multivariate Analysis: Theory and Practice*. M.I.T. Press, Cambridge, MA., 1975.

[5] R. L. Brennan and D. J. Prediger. Coefficient kappa: Some uses, misuses, and alternatives. *Educational and Psychological Measurement*, 41:687–699, 1981.

[6] M. R. Chernick, V. K. Murthy, and C. D. Nealy. Application of bootstrap and other resampling techniques: Evaluation of classifier performance. *Pattern Recognition Letters*, 3:167–178, May 1985.

[7] W. G. Cochran. *Sampling Techniques*. Wiley, New York, 2nd edition, 1963.

[8] J. Cohen. A coefficient for agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[9] P. R. Cohen. A Survey of the Eigth National Conference on Artificial Intelligence: Pulling Together or Pulling Apart? *AI Magazine*, pages 17–41, Spring 1991.

[10] R. Davis. Expert Systems How Far Can They Go (Part 1)? *AI Magazine*, pages 61–67, Spring 1989.

[11] R. Davis. Expert Systems How Far Can They Go (Part 2)? *AI Magazine*, pages 65–76, Summer 1989.

[12] R. M. Dawes and B. Corrigan. Linear modes in decision making. *Psychological Bulletin*, 81:95–106, 1974.

[13] R. M. Dawes, D. Faust, and P. E. Meehl. Clinical Versus Actuarial Judgment. *Science*, 243:1668–1674, 1989.

[14] J. Devore and R. Peck. *Statistics: the exploration and analysis of data*. West Publishing Company, St. Paul, 1986.

[15] N. R. Draper and H. Smith. *Applied Regression Analysis*. Wiley, New York, 2nd edition, 1981.

[16] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[17] B. Efron. *The Jackknife, the Bootstrap and Other Resampling Plans.* Society for Industrial and Applied Mathematics, Philadelphia, 1982.

[18] B. Efron. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *Journal of the American Statistical Association*, 78(382):316–331, June 1983.

[19] B. Efron and G. Gong. A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation. *The American Statistician*, 37:36–48, February 1983.

[20] B. Efron and R. Tibshirani. Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy. *Statistical Science*, 1(1):54–77, 1986.

[21] A. S. Elstein et al. Comparison of physicians' decisions regarding estrogen replacement therapy for menopausal women and decisions derived from a decision analytic model. *American Journal of Medicine*, 80:246–258, 1986.

[22] A. S. Elstein et al. Comparison of physicians' decisions regarding estrogen replacement therapy for menopausal women and decisions derived from a decision analytic model. In J. Dowie and A. Elstein, editors, *Professional judgment: A reader in clinical decision making*, chapter 15, pages 298–322. Cambridge University Press, New York, 1988.

[23] J. L. Fleiss. *Statistical Methods for Rates and Proportions*. John Wiley and Sons, New York, 2nd edition, 1981.

[24] S. J. Frank. What AI Practitioners Should Know about the Law. *AI Magazine*, pages 109–114, Summer 1988.

[25] D. H. Freeman, Jr. *Applied Categorical Data Analysis*. Marcel Dekker, Inc., New York, 1987.

[26] T. W. Froman and M. M. Llabre. The equivalence of kappa and del. *Perceptual and Motor Skills*, 60:3–9, 1985.

[27] K. Fukunaga and R. R. Hayes. Effects of sample size in classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):873 – 885, August 1989.

[28] K. Fukunaga and R. R. Hayes. Estimation of classifier performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1087–1101, October 1989.

[29] S. Geisser. The Predictive Sample Reuse Methd with Applications. *Journal of the American Statistical Association*, 70(350):320–328, 1975.

[30] J. Giarratano and G. Riley. *Expert Systems: Principles and Programming*. PWS-Kent Publishing Company, Boston, 1989.

[31] L. A. Goodman and W. H. Kruskal. Measures of association for cross classifications. *Journal of the American Statistical Association*, 49:732–764, 1954.

[32] L. A. Goodman and W. H. Kruskal. *Measures of Association for Cross Classifications*. Springer-Verlag, New York, 1979.

[33] L. C. Grebstein. Relative accuracy of actuarial prediction, experienced clinicians, and graduate students in a clinical judgement task. *Journal of Consulting Psychology*, 27(2):127–132, 1963.

[34] D. J. Hand. Recent advances in error rate estimation. *Pattern Recognition Letters*, 4:335–346, October 1986.

[35] P. Harmon and D. King. *Expert Systems: Artificial Intelligence in Business*. John Wiley and Sons, Inc., New York, 1985.

[36] D. K. Hildebrand, J. D. Liang, and H. Rosenthal. *Prediction analysis of cross classifications*. John Wiley and Sons, New York, 1977.

[37] W. Hoeffding. Asymptotic Normality. In S. Kotz, N. Johnson, and C. Read, editors, *Encyclopedia of statistical sciences*, pages 139–147. John Wiley and Sons, 1982.

[38] P. J. Hoffman, P. Slovic, and L. G. Rorer. An analysis-of-variance model for the assessment of configural cue utilization in clinical judgment. *Psychological Bulletin*, 69(5):338–349, 1968.

[39] Intellicorp, SunnyVale, CA. *KEE 3.0 Core Reference Manual*.

[40] Intellicorp, SunnyVale, CA. *KEE pictures Reference Manual*.

[41] Intellicorp, SunnyVale, CA. *KEE Reference Manual*.

[42] E. A Irani, J. P. Matts, J. M. Long, J. R. Slagle, and The POSCH group. Using Artificial Neural Nets for Statistical Discovery: Observations after using Back-propagation, Expert Systems, and Multiple-Linear Regression on Clinical Trial Data. *Complex Systems*, 3(3):295–311, 1989.

[43] D. Kahneman, P. Slovic, and A. Tversky, eds. *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge University Press, New York, 1982.

[44] L. Kanal and B. Chandrasekaran. On Dimensionality and Sample Size in Statistical Pattern Classification. *Pattern Recognition*, 3:225–234, 1971.

[45] L. H. Koopmans. *An Introduction to Contemporary Statistics*. Duxbury Press, Boston, MA, 1981.

[46] P. A. Lachenbruch and R. M. Mickey. Estimation of error rates in discriminant analysis. *Technometrics*, 10(1):1–11, 1968.

[47] K. Levi. Expert Systems Should be More Accurate than Human Experts: Evaluation Procedures from Human Judgment and Decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(3):647–657, May/June 1989.

[48] J. M. Long, E. A. Irani, D. W. Hunter, J. R. Slagle, J. P. Matts, et al. Using A Symbiotic Man/Machine Approach to Evaluating Visual Clinical Research Data. *Journal of Medical Systems*, 12(5):327–339, 1988.

[49] J. M. Long, J. R. Slagle, A. S. Leon, M. W. Wick, J. P. Matts, J. N. Karnegis, J. K. Bissett, H. S. Sawin, and J. P. Stevenson. An Example of Expert Systems Applied to Clinical Trials: Analysis of Serial Graded Exercise ECG Test Data. *Controlled Clinical Trials*, 8:136–145, 1987.

[50] J M Long, J R Slagle, M R Wick, E A Irani, P R Weisman, J P Matts, P F Clarkson, and The POSCH group. Lessons learned while implementing expert systems in the real world of clinical trial data analyses: The POSCH AI Project. In *Proceedings of the IEEE Symposium on the Engineering of Computer-Based Medical Systems*, pages 167–173. IEEE Computer Society Press, 1988.

[51] L. L. Lopes. Decision making in the short run. *Journal of Experimental Psychology [Human Learning and Memory]*, 7:377–385, 1981.

[52] N. Horace. Mann III. Artificial Intelligence in the Diagnosis of Low Back Pain. *Orthopedic Clinics of North America*, 22(2):303–314, April 1991.

[53] R. G. Miller. The jackknife - a review. *Biometrika*, 61(1):1–15, 1974.

[54] R. G. Miller., Jr. *Simultaneous Statistical Inference*. Springer-Verlag, New York, 2nd edition, 1981.

[55] T. M. Mullin. Experts' Estimation of Uncertain Quantities and its Implications for Knowledge Acquisition. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(3):616–625, May/June 1989.

[56] N. H. Nies et al. *Statistical Packages for the Social Sciences*. McGraw Hill, New York, 1975.

[57] R. M. O'Keefe and E. P. Smith. Validating Expert System Performance. *IEEE Expert*, pages 81–90, Winter 1987.

[58] D. E. O'Leary. Validation of expert systems - with applications to auditing and accounting expert systems. *Decision Sciences*, 18:468–486, 1987.

[59] T. J. O'Leary, M. Goul, K. E. Moffitt, and A. E. Radwan. Validating Expert Systems. *IEEE Expert*, pages 51–58, June 1990.

[60] J. R. Quinlan. *Online documentation for C4.5*. University of Sydney, Sydney, Australia.

[61] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1:81–106, 1986.

[62] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume I and II. MIT Press, 1986.

[63] L. Sachs. *Applied Statistics: A Handbook of Techniques*. Springer series in statistics. Springer-Verlag, New York, 2nd edition, 1984.

[64] S. Schwartz and T. Griffin. *Medical Thinking: The Psychology of Medical Judgment and Decision Making.* Contributions to Psychology and Medicine. Springer-Verlag, New York, 1986.

[65] R. J. Serfling. *Approximation Theorems of Mathematical Statistics.* John Wiley and Sons, New York, 1980.

[66] E. H. Shortliffe. *MYCIN: A Rule-based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection.* PhD thesis, Stanford University, California, October 1974.

[67] E. H. Shortliffe. *Computer-Based Medical Consultations: MYCIN.* Elsevier, New York, 1976.

[68] E. H. Shortliffe. *Rule-based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, Mass., 1984.

[69] E. H. Shortliffe, R. Davis, S. G. Axline, B. G. Buchanan, C. C. Green, and S. N. Cohen. Computer-based Consultations in Clinical Therapeutics: Explanation and Rule Acquisition Capabilities of the MYCIN System. *Computers and Biomedical Research*, 8(4):303–320, 1975.

[70] J. R. Slagle, J. M. Long, M. R. Wick, J. P. Matts, and A. S. Leon. Expert Systems in Medical Studies - A New Twist. In *Proceedings of the Symposium of the International Association of Optical Engineers, Vol. 635*, pages 25–29, 1986.

[71] J. R. Slagle, J. M. Long, M. R. Wick, J. P. Matts, and A. S. Leon. The ETA Project: A Case Study of Expert Systems for Analysis of Serial Clinical Trial Data. In *Proceedings of MEDINFO '86*, pages 155–159, 1986.

[72] J. R. Slagle, M. R. Wick, and M. O. Poliac. AGNESS: A Generalized Network-Based Expert System Shell. In *Proceedings of the NCAI*, pages 996–1002. AAAI, 1986.

[73] S. M. Snapinn and J. D. Knoke. Bootstrapped and smoothed classification error rate estimators. *Communications in statistics: Simulation and computation*, 17(4):1135–1153, 1988.

[74] E. L. Spitznagel and J. E. Helzer. A Proposed Solution to the Base Rate Problem in the Kappa Statistic. *Archives of General Psychiatry*, 42:725–728, 1985.

[75] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society - Series B*, 36(2):111–147, 1974.

[76] J. A. Swets, D. J. Getty, R. M. Pickett, C. J. D'Orsi, S. E. Seltzer, and B. J. McNeil. Enhancing and Evaluating Diagnostic Accuracy. *Medical decision making*, 11(1):9–18, January-March 1991.

[77] M. A. Tanner and M. A. Young. Modeling Agreement Among Raters. *Journal of the American Statistical Association*, 80(389):175–180, 1985.

[78] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles.* Addison-Wesley Publishing Company, Reading, MA, 1974.

[79] G. T. Toussaint. Bibliography on Estimation of Misclassification. *IEEE Transactions on Information Theory*, IT-20(4), July 1974.

[80] A. Tversky and D. Kahneman. The framing of decisions and the psychology of choice. *Science*, 211:453–458, 1981.

[81] D. A. Waterman. *A Guide To Expert Systems*. Addison-Wesley, Reading, MA, 1986.

[82] S. Weiss. Small Sample Error Rate Estimation for k-NN Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):285–289, March 1991.

[83] K. Wernecke. A Coupling Procedure for the Discrimination of Mixed Data. *Biometrics*, 48:497–506, June 1992.

[84] S. S. Wilks. *Mathematical Statistics*. John Wiley and Sons, New York, 1962.

[85] J. S. Zeide and J. Liebowitz. Using Expert Systems: The Legal Perspective. *IEEE Expert*, pages 19–21, Spring 1987.

[86] R. Zwick. Another Look at Interrater Agreement. *Psychological Bulletin*, 103(3):374–378, 1988.

Index

L, 17
$P(A)$, 10
PH, 171, 177
PL, 171, 175
PSH, 171, 175
PSL, 171, 173
$SE(\theta)$, 10
S_{diff}, 80
U method, 60
W, 17
χ^2, 14
$\bigtriangledown$, 18
$\hat{\pi}_i$, 12
$\hat{p}_{ij}$, 12
$\kappa_{b,w}$, 8, 20
κ, 18
κ_b, 8, 18
p_{ij}, 12
p_j, 12
π_i, 12
$SE(L)$, 17
$f_{i.}$, 12
f_{ij}, 12
p-value, 7
repeated random halving (RRH) technique, 75

AEPOSCH
    application, 26
    dataset, 26
AEPOSCH-B3 dataset, 183
AEPOSCH-B3-QA dataset, 183
AEPOSCH-BR dataset, 183
AEPOSCH-BR-QA dataset, 183
agreement, 13, 14
agreement function, 16
AN, 55
association, 14, 15
Asymptotic Normality (AN), 55

B3 dataset, 183
B3-QA dataset, 183
back-propagation, 40
    hidden units, 40
    intermediate layer, 42
    layered feed-forward networks, 40
Bonferroni p-values, 89, 181
bootstrap, 54, 62
    estimate of variance, 60
    replication sample, 62
BR dataset, 183
BR-QA dataset, 183
Brennan's kappa (κ_b), 18
bst-diff method, 81

C4, 46
Central Limit Theorem, 55
chi-square (χ^2), 14
classifier consistent with itself, 172
Classifier-based Overall Loss Minimization Strategy (COLMS), 9, 154
coefficient of variation (CV), 64
Cohen's kappa (κ_c), 18
collateral
    arteries, 35
    filling, 35
COLMS, 9, 154
Consensus A form, 28
Consensus B form, 28
contingency table, 12
    bivariate, 12
    rectangular, 12
cross-classification, 12
    upper triangular matrix, 182
cross-validation, 59

data shuffling method, 59
diff dataset, 80
diff method, 80

distal segments, 35

EDSS, 160
EDT, 169
    perturbation-testing, 169
eight-point scale, 28
ESCA, 2, 29
ESCA (Evaluator of Serial Coronary Angiograms), 29
ETA, 2
evaluation
    quality-control, 182
    regular, 182
expert-defined test (EDT), 169

gjk-diff method, 81
grouped jackknife, 54, 61

Hildebrand's $\bigtriangledown$, 18
holdout method, 58

ID3, 46
Independent Film A form, 27
Independent Film B, 28

jackknife, 61

kappa, 18
kappa base, 18
kurtosis, 65

leave-1-out method, 60
    estimate of variance, 60
leave-k-out, 60
loss function, 15

M14 form, 28, 29
marginals, 13
    homogeneous, 22
mean squared error (MSE), 64
multiple comparisons, 89
multiple linear regression, 39

normative decision theory, 5

OLMS, 9, 157
over-training, 72
Overall Loss Minimization Strategy (OLMS), 9, 157

perturb
    high (PH), 171, 177
    low (PL), 171, 175
    slightly high (PSH), 171, 175
    slightly low (PSL), 171, 173
perturbation scheme
    arbitrary, 171
    flexible, 170
    rigid, 170
perturbation-testing, 169
POSCH, 2, 26
Program On the Surgical Control of the Hyperlipidemias (POSCH), 2, 26
proportionate reduction of error (PRE) measures, 18

quality-control evaluation, 182

regular evaluation, 182
repeated random halving (RRH) technique, 73
resubstitution method, 58
rotation method, 59

SB, 79
SGJK, 79
SM, 78
split bootstrap method (SB), 79
split grouped jackknife method (SGJK), 79
split methods, 80
split multinomial method (SM), 78
split recombination (SR) conjecture, 78
split recombination (SR) procedure, 76
SR conjecture, 60, 78
standard error, 10
stenosis, 27

weight function, 16

Yule's Q, 15

zero-one loss function, 154