Efficient training of Support Vector Machines and their hyperparameters

By

Charl J. van Heerden

Thesis submitted for the degree

Philosophiae Doctor in Computer Engineering

at the

Potchefstroom campus

of the

NORTH-WEST UNIVERSITY

Advisor: Professor Etienne Barnard

September 2012

SUMMARY

Efficient training of Support Vector Machines and their hyperparameters

by

Charl J. van Heerden

Advisor: Professor Etienne Barnard

North-West University

Philosophiae Doctor in Computer Engineering

As digital computers become increasingly powerful and ubiquitous, there is a growing need for pattern-recognition algorithms that can handle very large data sets. Support vector machines (SVMs), which are generally viewed as the most accurate classifiers for general-purpose pattern recognition, are somewhat problematic in this respect: as for all classifiers which employ hyperparameters, the behavior of SVMs depends strongly on the particular choice of hyperparameter values, and popular approaches to training SVMs require computationally expensive grid searches to choose these parameters appropriately [1, 2]. Our main objective is therefore to find more efficient ways to train SVM hyperparameters. We also show that for non-separable datasets, SVMs do not behave like large margin classifiers. This observation in turn leads us to explore algorithms which do not employ a margin term. Since one of the hyperparameters of SVMs is a regularization parameter that controls the relative contribution of the margin term and the sum of misclassifications, dropping the margin term means that there is one less hyperparameter to be trained.

Grid searches are an expensive yet widely used technique to train the SVM hyperparameters. We therefore investigate ways in which the hyperparameters can be trained more efficiently, since the traditional grid search approach to finding good parameters takes very long. We also investigate alternative algorithms which are similar to SVMs, but which have fewer hyperparameters to find.

With this goal in mind, we first investigate the scaling and asymptotic behaviors of popular SVM hyperparameters on non-separable datasets. We find that the scale factor of the radial basis function (RBF) kernel depends only weakly on the size of the training set and that the regularization parameter C must assume relatively large values for accurate classi-

fication to be achieved. The observation with regard to C is true for all datasets considered in the thesis when a linear kernel is employed, while for RBF kernels the evidence is not as strong.

The preference for large C casts doubt on the large margin classifier (LMC) tag often associated with SVMs, especially with linear kernels. Further investigation confirms our suspicion that minimization of an error term, rather than maximization of the inter-class margin, is responsible for the widely acknowledged excellence of SVM classifiers.

These insights suggest two different approaches to reducing overall SVM training time: SVM hyperparameter training on reduced training sets and stochastic optimization of a simplified criterion function. The SVM hyperparameter training on reduced training sets is further enhanced by a heuristic for the choice of the RBF scale factor. This enables us to propose a hyperparameter selection algorithm that performs as well as the conventional SVM approach on all classification problems considered in this thesis, while reducing the required training time by several orders of magnitude. Our second approach, stochastic optimization of a simplified criterion, is slightly less accurate on some problems, but reduces the overall training time even further. With training sets consisting of tens of thousands of samples, efficient hyperparameter selection for standard SVMs is the method of choice. Looking to the future where training-set sizes will inevitably continue to increase, methods such as our stochastic approach will become preferable for a growing proportion of practical problems.

OPSOMMING

Efficient training of Support Vector Machines and their hyperparameters

deur

Charl J. van Heerden

Adviseur: Professor Etienne Barnard

Noordwes-Universiteit

Philosophiae Doctor in Rekenaaringenieurswese

Soos rekenaarkrag toeneem en data-bronne toenemend groei word die behoefte aan patroonherkenningsalgoritmes wat baie groot datastelle kan verwerk diensooreenkomstig groter. Steunvektorstelsels (SVSe) word in die algemeen as die mees akkurate veeldoelige klassifiseerders beskou, maar hul afrigting mag problematies wees: soos vir alle klassifiseerders wat hiperparameters gebruik, hang die vertoning van SVSe baie sterk af van die spesifieke keuse van een of meer hiperparameterwaardes, en die beste huidige benaderings verg 'n roostersoektog om geskikte waardes vir hierdie parameters te vind, wat rekenaarmatig baie duur kan wees [1, 2].

Ons hoofdoel is dus om meer doeltreffende maniere te vind om die SVS-hiperparameters te vind. Ons wys ook dat SVSe se gedrag nie soos dié van grootgrensklassifiseerders vir nieskeibare datastelle blyk te wees nie. Hierdie waarneming lei ons om algoritmes te verken wat nie 'n grensterm bevat nie. Aangesien een van die hiperparameters van SVSe 'n regulariseringsparameter is wat die relatiewe bydrae van die grensterm en die som van die foutiewe klassifikasies beheer, beteken dit dat daar een minder hiperparameter is om te skat indien die grensterm wegval.

Roostersoektogte is 'n duur dog gewilde tegniek om SVS-hiperparameters af te rig. Ons ondersoek dus maniere waarop die hiperparameters meer effektief afgerig kan word, aangesien die tradisionele roostersoektoeg baie lank neem. Ons ondersoek ook alternatiewe algoritmes wat soortgelyk is aan SVSe, maar wat een minder hiperparameter bevat.

Met hierdie doel in gedagte ondersoek ons eerstens die skalering en asimptotiese gedrag van die mees tipiese SVS hiperparameters. Ons vind dat die skaleringsfaktor van die radiale basisfunksie (RBF) kern baie min geaffekteer word deur die grootte van die leerstel, en dat

die regulariseringsparameter (C) relatief groot moet wees vir akkurate klassifisering. Hierdie waarneming geld vir alle datastelle wat in hierdie studie ondersoek is wanneer 'n lineêre kern gebruik word. Vir 'n RBF-kern is die bewyse egter nie so sterk nie.

Hierdie waarneming lei tot die bevraagtekening van die algemene beskouing van SVSe as grootmargeklassifiseerders. Verdere analise bevestig dat die minimering van die fout-term eerder as die maksimering van die interklasmarge die hoofdryfveer is vir die uitnemende vertoning van SVSe.

Hierdie insig lei tot twee verskillende benaderings tot die vermindering van die tyd wat dit neem om die SVS-hiperparameters af te rig: SVS-hiperparameterafrigting op kleiner datastelle en stogastiese optimering van 'n soortgelyke doelfunksie. Ons verbeter eersgenoemde metode verder deur 'n heuristiese keuse vir die RBF-skaalfaktor voor te stel, en skep so 'n algoritme vir hiperparameterseleksie wat op 'n wye reeks klassifikasieprobleme so goed soos meer konvensionele SVSe vertoon, maar waarvan die afrigtyd met verskeie grootte-ordes verkort. Die tweede benadering op sigself is ietwat minder akkuraat vir die oplos van sommige probleme, maar verminder tog die algehele afrigtyd van die hiperparameters. Vir leerstelle wat bestaan uit tienduisende items word effektiewe hiperparameterseleksie saam met standaard-SVSe die verkose metode. Aangesien ons weet dat datastelle in die toekoms slegs gaan vergroot, glo ons dat stogastiese metodes soos ons nuut-voorgestelde stogastiese benadering 'n belowende opsie is.

TABLE OF CONTENTS

CHAPTER ONE - INTRODUCTION 2

1.1 Support vector machines . . . 3

1.2 Perceptron kernel criterion . . . 4

1.3 Objectives, hypotheses and outline . . . 5

CHAPTER TWO - BACKGROUND 6

2.1 SVM error function . . . 7

2.1.1 Linear SVMs and separable data . . . 7

2.1.2 Non-separable data . . . 9

2.1.3 Non-linear extension . . . 10

2.2 SVM hyperparameters . . . 11

2.2.1 Estimating the generalization error . . . 12

2.2.2 Minimizing the generalization error . . . 14

2.2.2.1 Exhaustive search . . . 14

2.2.2.2 Focused exhaustive search . . . 15

2.2.2.3 Gradient descent . . . 15

2.2.2.4 Evolutionary algorithms . . . 16

2.3 Approaches to solve the SVM primal/dual optimization problem . . . 17

CHAPTER THREE - EMPIRICAL METHODS 19

3.1 Estimating the generalization error . . . 20

3.2 Minimizing the generalization error . . . 21

3.3 Statistical hypothesis testing . . . 21

3.3.1 Independent two-sample t-tests . . . 22

3.3.2 The Wilcoxon signed-rank test . . . 22

3.4 Datasets . . . 23

3.4.1 Artificial dataset . . . 24

5.2.3 The perceptron kernel approach 63
5.2.3.1 PK with a linear kernel 64
5.2.3.2 PK with an RBF kernel 65
5.3 Conclusion 65

CHAPTER SIX - STOCHASTIC GRADIENT DESCENT 69
6.1 Introduction and background 69
6.2 Step size update algorithm 71
6.3 Block size 75
6.4 Initial step size 84
6.5 Stopping criteria 93
6.6 Conclusion 98

CHAPTER SEVEN - COMPARATIVE RESULTS 99
7.1 Approaches compared 99
7.2 Results 102
7.3 Analysis of comparative performance and efficiency 111
7.3.1 Comparing the algorithms 113
7.3.2 Choosing γ 113
7.3.3 Amount of data for hyperparameter search 114
7.3.4 Statistical significance tests 114
7.4 Conclusion 129

CHAPTER EIGHT - CONCLUSION 130
8.1 Summary 130
8.2 Contribution 131
8.3 Unresolved issues 132

REFERENCES 134

APPENDIX A - LIST OF ACRONYMS 142

LIST OF FIGURES

4.1 *10-fold cross-validation accuracy for linear SVMs against* $log(C)$. *All functions converge after a sufficiently high value of* C. 32

4.2 *Number of SVs vs* $log(C)$. *The number of SVs are indicated as* $\frac{true\ num\ SVs}{total\ num\ SVs}$ *as the dataset sizes (and hence the number of SVs) vary significantly, making presentation on a single graph difficult. It is clear that for small* C, *the algorithm does not learn much and assigns almost all points as SVs.* 33

4.3 *An artificial two-class dataset is shown (class one samples are shown in red and class two samples are shown in blue). Support vectors for class one are highlighted in black, while SVs from class two are highlighted in green. The dataset was generated by randomly sampling points from a Gaussian mixture model (for details see Section 3.4.1). The data points that are retained as SVs after training an SVM, and the corresponding decrease of the number of such data points as* C *is increased from* 10^{-2} *to* 10^{4} *are shown. (*γ *is kept constant at* $10^{-0.5}$ *in all cases.) It is clear that as* C *is increased, the SVM starts to approximate the true boundary between the classes, as the SVs become more concentrated on that boundary.* 34

4.4 *Contour plots depicting the CV accuracy over a wide range of* $log(C)$ *and* $log(\gamma)$ *for the UCI Diabetes, Thyroid, Heart and German datasets. Accuracy is indicated by color (see color bar next to each figure), with dark blue corresponding to the lowest accuracy achieved and dark red to the highest accuracy achieved.* 35

4.5 *Contour plots showing the results of a grid search with varying amounts of data. Fig. 4.5(a) shows a contour plot for all of the data, with every subsequent figure generated with half the amount of data of the previous plot. In this fashion, Fig. 4.5(d) is generated with an eighth of the amount of data used for Fig. 4.5(a). Accuracy is indicated by color (see color bar next to each figure), with dark blue corresponding to the lowest accuracy achieved and dark red to the highest accuracy achieved.* 37

4.6 *Density estimates for the distance to the nearest neighbor when randomly sampling N points from a five- and 10 dimensional normal distribution with zero mean and unit variance. Note the weak implied relationship between γ and N.* 43

4.7 *German: histograms depicting squared euclidean distances to all samples in the other 4.7(a) and the same 4.7(b) class, nn in the other 4.7(c) and same class 4.7(d) and to the other class mean 4.7(e), as well as to the same class mean 4.7(f).* 45

4.8 *Image: histograms depicting squared euclidean distances to all samples in the other 4.8(a) and the same 4.8(b) class, nn in the other 4.8(c) and same class 4.8(d) and to the other class mean 4.8(e), as well as to the same class mean 4.8(f).* 46

4.9 *Cross-section of the contour plot of hyperparameter values vs accuracy for both 10-fold cross-validation and LOO cross-validation. This particular cross-section was taken from one of the folds of the 12.5% Splice subset and depicts varying C vs classification accuracy with γ fixed at 0.01. It is interesting to note that C for the 10-fold cross-validation estimate has an apparent best accuracy at $C \approx 10^0$, whereas the LOO CV estimate has no peak in accuracy; rather, the accuracy reaches an asymptote, after which further increases in C have no further visible effect on accuracy.* 55

5.1 *Contour plots showing the results of a grid search (10-fold cross-validation accuracy) for C and γ in an RBF kernel. Note that in all cases, a very large C can provide competitive (if not the best) results for some γ. Accuracy is indicated by color (see color bar next to each figure), with dark blue corresponding to the lowest accuracy achieved and dark red to the highest accuracy achieved.* 67

5.2 *Contour plots showing the results of a grid search (10-fold cross-validation accuracy) for C and γ in an RBF kernel. Note that in all cases, a very large C can provide competitive (if not the best) results for some γ. Accuracy is indicated by color (see color bar next to each figure), with dark blue corresponding to the lowest accuracy achieved and dark red to the highest accuracy achieved.* 68

6.1 *10-fold cross-validation accuracy for a γ line search as a function of different constant factors between the streams in the three-stream optimization algorithm. Accuracy is indicated by color (see color bar next to each figure), with dark blue corresponding to the lowest accuracy achieved and dark red to the highest accuracy achieved.* 74

6.2 *Sum of squared distances of gradients calculated using a particular block size, to the gradient calculated using all samples. The trend is almost linear up to* $\sim 5k$ *samples, after which an apparent non-linear decrease takes place. We believe that this is due to the way in which the squared distances were generated; from* $\sim 5k$ *samples onwards, a single block was used to estimate the squared distances.* 76

6.3 *Train set error vs size of the block used for estimating the gradient when using our three-stream approach. The x axis is limited to the number of samples required by SGD using a block size of one to complete training.* 77

6.4 *Train set error vs size of the block used for estimating the gradient when using Rprop. The x-axis is limited to the number of samples required by SGD using a block size of one to complete training.* 78

6.5 10*-fold cross-validation accuracy when optimizing the PK using Rprop for different block sizes. Accuracy is indicated by color (see color bar next to each figure), with dark blue corresponding to the lowest accuracy achieved and dark red to the highest accuracy achieved.* 79

6.6 *Contour plots for several datasets, with initial step size* η *vs* γ*. While it is not true for all datasets, for the four in this figure, none of the values at the optimal* γ *are statistically significantly better than the others across initial step sizes. Accuracy is indicated by color (see color bar next to each figure), with dark blue corresponding to the lowest accuracy achieved and dark red to the highest accuracy achieved.* 85

6.7 *Objective function value vs number of epochs, with the standard error of each graph also plotted. The Heart, Image and Thyroid datasets have significantly different objective function values from the rest of the datasets and are therefore displayed separately. Fig. 6.7(d) shows objective function values for* 10 *datasets that are all in approximately the same absolute range. It is clear that most learning takes place within the first epoch and that after about five to* 10 *epochs, very little (if any) learning takes place.* 94

6.8 *Objective function value vs number of epochs, with the standard error of each graph also plotted. In contrast to figure 6.7(d), the initial error has been omitted, which shows the objective function's behavior after the first epoch more clearly.* 95

6.9 10-*fold cross-validation accuracy and number of training samples used during training, for six data sets. A grid search was performed across different thresholds: a minimum threshold on the mean running delta training error and an upper threshold on the maximum number of epochs without a significantly best stream emerging. Accuracy is indicated by color (see color bar next to each figure), with dark blue corresponding to the lowest accuracy achieved and dark red to the highest accuracy achieved.* 97
7.1 *Thyroid* 104
7.2 *Heart* 104
7.3 *Breast Cancer* 105
7.4 *Diabetes* 105
7.5 *German* 106
7.6 *Solar Flare* 106
7.7 *Titanic* 107
7.8 *Image* 107
7.9 *Splice* 108
7.10 *Banana* 108
7.11 *DFKI classes 1 & 4* 109
7.12 *DFKI classes 2 & 5* 109
7.13 *DFKI classes 5 & 7* 110
7.14 *MNist classes 0 & 3. This figure does not show error bars, since a single test set was employed.* 110

LIST OF TABLES

3.1 *Number of instances, dimensions and classes of all data sets. For those data sets marked with an asterisk, the IDA benchmark repository version of the dataset is slightly different from UCI.* 24

3.2 *Mathematical notation used throughout the thesis.* 28

4.1 *Grid search results when using all (100%) of the training samples on the Image dataset.* 38

4.2 *Grid search results when using 50% of the training samples on the Image dataset.* 39

4.3 *Grid search results when using 25% of the training samples on the Image dataset.* 40

4.4 *Grid search results when using 12.5% of the training samples on the Image dataset.* 41

4.5 *Grid search results when using 6.25% of the training samples on the Image dataset.* 42

4.6 *Mean euclidean distances (μ) between samples for several datasets. The subscripts o and s refer to samples from the other and same classes respectively. We also show the number of dimensions if the first 95% and 80% of the variance were explained respectively if considering eigenvalues and eigenvectors calculated on the data covariance matrix. BC refers to the Breast Cancer dataset.* . . 47

4.7 *Correlation coefficients for the different measures from Table 4.6.* 48

4.8 *The 10-fold cross-validation error rates obtained using SVMs with RBF kernels. Datasets marked with an asterisk show results for a γ line search without C adaptation. The DFKI dataset results are reported on the single accompanying test set. No cross-validation was thus performed and for that reason a single error rate is reported as opposed to a mean and standard error. Also note that none of the results are statistically signficant (see Table 4.11. The results from Rätsch are excluded from the statistical analysis because of different experimental protocols).* 51

4.9 *Approximate total CPU time for performing grid searches in Table 4.8. While the cluster on which these times were measured was used exclusively for the experiments in question, the times can only be indicative of general duration, since care was not taken to optimize for cache misses, for example, which could have a significant impact on run time performance.* 51

4.10 *The* 10*-fold cross-validation mean and standard error when training SVMs with (1) a full grid search for* C *and* γ *(*$s(C, \gamma)$*) and (2) with* $\gamma = \frac{1}{d}$*, followed by a line search over* C *(*$s(C, \gamma = \frac{1}{d})$*). The hyperparameter training time is also included. Paired Wilcoxon rank sum tests were performed and it was found that none of the results are statistically significantly different at the* 0.01 *significance level.* 52

4.11 *Statistical significance test results corresponding to the* 10*-fold cross-validation results presented in Table 4.8. In Table 4.8, results are presented when SVM hyperparameters are trained using the algorithm proposed in Section 4.4.3. The percentage for each method corresponds to the percentage of the total number of available training samples used to perform the initial grid search, while a * indicates results where no* C *scaling was performed. In this table, the independent two-sample t-test is used to test whether or not a particular method performs significantly better than another at the* 0.01 *significance level. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 53

5.1 *The* 10*-fold cross-validation mean error and standard error obtained using linear SVMs, the PK method and LSQ are shown for several datasets from the IDA benchmark repository. The optimal* C*-value for each fold was obtained by performing* 10*-fold cross-validation on each fold's training set. The last column shows the contribution of the margin and misclassification terms in the SVM error function respectively. We show the median value of* $C \sum_i \xi_i$*, as calculated during cross-validation. The independent two-sample t-test is used to test whether or not a particular method performs significantly better than another at the* 0.01 *significance level. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 61

5.2 *The* 10*-fold cross-validation mean error and standard error obtained using SVMs with an RBF kernel as well as the PK method (also using an RBF kernel) are shown for several datasets from the IDA benchmark repository. The optimal* C*-value for each fold was obtained by performing* 10*-fold cross-validation on each fold's training set. The last column shows the contribution of the margin and misclassification terms in the SVM error function respectively. We show the median value of* $C \sum_i \xi_i$*, as calculated during cross-validation. The independent two-sample t-test is used to test whether or not a particular method performs significantly better than another at the* 0.01 *significance level. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 62

6.1 *Different block sizes for Rprop for the Banana dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 80

6.2 *Different block sizes for Rprop for the Breast Cancer dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 80

6.3 *Different block sizes for Rprop for the Diabetes dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* . 80

6.4 *Different block sizes for Rprop for the Solar Flare dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* . 81

6.5 *Different block sizes for Rprop for the german dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 81

6.6 *Different block sizes for Rprop for the Heart dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 81

6.7 *Different block sizes for Rprop for the image dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 82

6.8 *Different block sizes for Rprop for the Splice dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 82

6.9 *Different block sizes for Rprop for the Thyroid dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 82

6.10 *Different block sizes for Rprop for the Titanic dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 83

6.11 *Different block sizes for Rprop for the DFKI-1-4 dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* . 83

6.12 *Different block sizes for Rprop for the DFKI-2-5 dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* . 83

6.13 *Different block sizes for Rprop for the DFKI-5-7 dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular block size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a block size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* . 84

6.14 *The* 10*-fold cross-validation mean and standard error on a subset of the Heart dataset, where the initial step size* η *and kernel width* γ *are varied. None of the values at the optimal* γ *are statistically significantly better than the remainder at different initial step sizes. A log scale is used to show the results, otherwise results at small values would all be on one end of the linear scale.* 86

6.15 *Different initial step sizes (the log of the step size is shown) for the Banana dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 87

6.16 *Different initial step sizes (the log of the step size is shown) for the Breast Cancer dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 87

6.17 *Different initial step sizes (the log of the step size is shown) for the Diabetes dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 88

6.18 *Different initial step sizes (the log of the step size is shown) for the Solar Flare dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 88

6.19 *Different initial step sizes (the log of the step size is shown) for the german dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 89

6.20 *Different initial step sizes (the log of the step size is shown) for the Heart dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 89

6.21 *Different initial step sizes (the log of the step size is shown) for the image dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 90

6.22 *Different initial step sizes (the log of the step size is shown) for the Splice dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 90

6.23 *Different initial step sizes (the log of the step size is shown) for the Thyroid dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 91

6.24 *Different initial step sizes (the log of the step size is shown) for the Titanic dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 91

6.25 *Different initial step sizes (the log of the step size is shown) for the DFKI-1-4 dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 92

6.26 *Different initial step sizes (the log of the step size is shown) for the DFKI-2-5 dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 92

6.27 *Different initial step sizes (the log of the step size is shown) for the DFKI-5-7 dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular initial step size is significantly better than another at the* 0.01 *significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that an initial step size in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 93

7.1 *The cases where some algorithmic variants performed so poorly so as to be omitted from Fig. 7.1 – Fig. 7.14 (prevent excessive compression of the scale of the vertical axis).* 103

7.2 *In this table we compare the run times of the PK optimized using our three-stream approach (PK), optimized using batch Rprop (Rprop) and SVMs trained with SMO (SVM). The durations represent the time it took to train a single classifier. The durations are highly variable for several different reasons: poor hyper-parameter choices can have a big impact on training time, as do more difficult problems. For this reason, we include the fastest and slowest single training times for each algorithm and each problem, as well as the time it took to train the machine yielding the lowest and highest error rates. Durations are indicated in seconds, with the error rate in brackets. Also note that in some cases, the time it takes to train a classifier is less than* 100 *ms, in which case we represent it as 0.* 112

7.3 *The 10-fold cross-validation mean and standard error when training with the full, half and 10% sets of different realizations of 90% of the data. These numbers are slightly optimistic, since the best value in the grid was selected each time. The errors are also measured on different subsets of the data than the exact folds used to report the 10-fold cross-validation mean and standard errors and should thus be compared only with the other data points in this table. MNist was not trained with 10-fold cross-validation; rather, the standard validation set was used for evaluation.* 115

7.4 *The different model selection strategies for the Banana dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (25, 50, 75) obtained on the same 10 folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 116

7.5 *The different model selection strategies for the Breast Cancer dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (25, 50, 75) obtained on the same 10 folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 117

7.6 *The different model selection strategies for the Diabetes dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (25, 50, 75) obtained on the same 10 folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 118

7.7 *The different model selection strategies for the Solar Flare dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 119

7.8 *The different model selection strategies for the German dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 120

7.9 *The different model selection strategies for the Heart dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 121

7.10 *The different model selection strategies for the Image dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 122

7.11 *The different model selection strategies for the Splice dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 123

7.12 *The different model selection strategies for the Thyroid dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 124

7.13 *The different model selection strategies for the Titanic dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* !0 *folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 125

7.14 *The different model selection strategies for the DFKI-1-4 dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 126

7.15 *The different model selection strategies for the DFKI-2-5 dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 127

7.16 *The different model selection strategies for the DFKI-5-7 dataset are compared in the same format as in [3]. The paired Wilcoxon rank sum test is used to test whether or not a particular selection strategy performs significantly better than another at the 0.01 significance level. Quantiles of the test errors (*25, 50, 75*) obtained on the same* 10 *folds are also shown. Using the same notation as in [3], we indicate that a method in a particular row performs significantly better (<), worse (>) or statistically similar (no symbol) than the method in the corresponding column.* 128

LIST OF ALGORITHMS

1 SVM using SMO 10

2 Rprop algorithm for block size β (adapted from [4]). Δ_0 is the initial step size, Δ_{ij} is the weight-specific step size and ω_{ij} are the weights. **sign** returns $+1$ (positive argument), -1 (negative argument) or 0 otherwise. Δ_0 is not critical [4] and is set to 0.1. 72

3 Three-stream algorithm for block size β 73