Cost-Conscious Comparison of Supervised Learning Algorithms over Multiple Data Sets

Mehmet Aydın Ulaş, Olcay Taner Yıldız, Ethem Alpaydın

Technical Report, FBE/CMPE-01/2008-04

Institute of Graduate Studies in Science and Engineering

Department of Computer Engineering

Boğaziçi University

ulasmehm@boun.edu.tr

olcaytaner@isikun.edu.tr

alpaydin@boun.edu.tr

June 30, 2008

ABSTRACT

We propose Multi^{2}Test for ordering multiple learning algorithms on multiple data sets from "best" to "worst." Our goodness measure uses a prior cost term additional to generalization error. Our simulations show that Multi^{2}Test generates orderings using pairwise tests on error and different types of cost.

1. Introduction

In choosing among multiple algorithms, one can either select according to past experience, choose the one that is currently the most popular, or resort to some kind of objective measure. In classification, there is no single algorithm which is always the most accurate and the user is faced with the question of which one to favor. We also note that generalization error, though the most important, is rarely the sole criterion in choosing among algorithms and other criteria, such as training/testing time/space complexity, interpretability of results, ease of programming, etc. may also play an important role.

When a researcher proposes a new learning algorithm or a variant, he/she compares its performance with a number of existing algorithms on a number of data sets. These data sets may come from a variety of applications (such as those in the UCI repository [1]) or may be from some particular domain (for example, a set of face recognition data sets). In either case, the aim is to see how this new algorithm/variant ranks with respect to the existing algorithms either in general, or for the particular domain at hand, and this is where a method to compare algorithms on multiple data sets will be useful. Especially in data mining applications where users are not necessarily experts in machine learning, an automated method for choosing the best of a number of candidate learning algorithms is useful. When designing a general purpose data mining tool, one may need a methodology to compare multiple algorithms over multiple data sets automatically without user intervention.

To compare the generalization error of learning algorithms, statistical tests have been proposed [2, 3]. In choosing between two, one can use a pairwise test to compare their generalization error and select the one that has lower error. Typically, cross-validation is used to generate a set of training, validation folds, and we compare expected error on the validation folds. Examples of such tests are parametric tests, such as k-fold paired t test, 5×2 cv t test [2], 5×2 cv F test [4], nonparametric tests, such as the sign test and Friedman's test, or range tests, such as Wilcoxon signed rank

test. Bouckeart [5] showed that the widely used t test showed superior performance compared to the sign test in terms of replicability. On the other hand, he found the 5×2 cv t test dissatisfactory and suggested the corrected resampled t test. Resampling still has the problem of high Type I error, and this issue has been theoretically investigated by Nadeau and Bengio [6]. They propose variance correction to take into account not only the variability due to test sets, but also the variability due to training examples.

Although these tests are good for comparing the means of two populations (that is, the expected error rate of two algorithms), they can not be used to compare multiple populations (algorithms). In our previous work [7], we proposed the MultiTest method to order multiple algorithms in terms of "goodness" where goodness takes into account both generalization error and a prior term of cost. This cost term accounts for what we try to minimize additional to error and allows us to choose between algorithms when they have equal expected error.

A further need is to be able to compare algorithms over not a single data set but over multiple data sets. Demsar [3] examines various methods, such as the sign test and Friedman's test together with its post-hoc Nemenyi's test, for comparing multiple algorithms over multiple data sets. These methods can make pairwise comparisons, or find subsets of equal error, but lack a mechanism of ordering and therefore, for example, cannot always tell which algorithm is the best.

In this report, we generalize the MultiTest method so that it can work on multiple data sets and hence is able to choose the best, or in the general case, order an arbitrary number of learning algorithms on an arbitrary number of data sets from best to worst. Our simulation results using eight classification algorithms on 38 data sets indicate the utility of this novel Multi^{2}Test method. We also show the effect of different cost terms on the final ordering.

Tests come with approximations (of normality), assumptions (of independence) that do not always hold, limited power on small sets, or artificially set significance

levels, but in the absence of a more reliable alternative remain the best way to assess or compare the accuracy of learning algorithms.

This report is organized as follows: In Chapter 2, we review the statistical tests for comparing multiple algorithms. We propose the Multi^{2}Test method in Chapter 3. Our experimental results are given in Chapter 4 and Chapter 5 concludes.

2. Comparing Multiple Algorithms over Multiple Data sets

Two nonparametric methods have been discussed in [3] to compare multiple algorithms over multiple data sets, namely, the sign test and Friedman's test together with its post-hoc Nemenyi's test.

2.1. Sign Test

Given S data sets, we compare two algorithms by using a pairwise test (over the validation folds) and we let the number of wins of one algorithm over the other be w, and we let the number of losses be l where $S' = w + l$ (we ignore the ties). The sign test assumes that the wins/losses are binomially distributed and tests the null hypothesis that $w = l$. We calculate $p = B(w, S')$ of the binomial distribution and if $p > \alpha$, we fail to reject the hypothesis that the two have equal error with significance α. Otherwise, we say that the first one is more accurate if $w > l$, and the second one is more accurate otherwise. For large values of S', we can use an approximation for $p = (w - 0.5S')/\sqrt{0.25S'}$; we fail to reject the test if $p \in (-z_{\alpha/2}, z_{\alpha/2})$.

Note that sign test results cannot be used to find an ordering: For three algorithms A, B, C, if A is more accurate than C and B is also more accurate than C and if A and B have equal error, we do not know which to pick as the first, A or B. This is where the concept of cost and the methodology of MultiTest comes into play.

2.2. Friedman's Test and Nemenyi's Test

Friedman's test is the nonparametric equivalent of ANOVA. First, all algorithms are ranked on each data set using the average error on the validation folds, giving the best one rank 1. If the algorithms have no difference between their expected errors, then their ranks should not be different either, which is what is tested by Friedman's test. Let r_{ij} be the rank of algorithm $j = 1, \ldots, L$ on data set $i = 1, \ldots, S$ and

$R_j = \frac{1}{S}\sum_i r_{ij}$ be the average rank of algorithm j. The Friedman test statistic is:

$$X^2 = \frac{12S}{L(L+1)}\left[\sum_j R_j^2 - \frac{L(L+1)^2}{4}\right] \tag{2.1}$$

which is chi-square distributed with $L-1$ degrees of freedom. If the test fails to reject, we will say that we cannot find any difference between the means of the L algorithms and we do no further processing; if the test rejects, we will use the post-hoc Nemenyi's test.

According to Nemenyi's test, two algorithms have different error rates if their average ranks differ by at least a critical difference, $CD = q_\alpha\sqrt{\frac{L(L+1)}{6S}}$, where values for q_α are based on the Studentized range statistic divided by $\sqrt{2}$. The subsets of algorithms which have equal error are denoted by underlining them. An example result is:

$$\underline{A\ \underline{B}\ \underline{C}\ D}$$

where algorithms are sorted in ascending average error. We see that there is no difference between A and B, no difference between B and C but there is difference between A and C.

Note that Nemenyi's test can find subsets of algorithms which have comparable error on S data sets but cannot order $L > 2$ algorithms, e.g., to be able to find the best one. A range test checks for equality and a rejection, that is, the absence of an underline, does not imply an ordering. For example, we know that A and C have significantly different errors and that the average errors of A is less than the average errors of C but this does not necessarily mean that A has significantly less error than C; a range test does not check for this. Nor does it provide us a mechanism to choose between two algorithms which have no significant difference between them, for example A and B. Note also that Nemenyi's test is too conservative, has low power, and may

not detect existing differences, even if Friedman's test rejects; this is expected to occur very rarely [3].

3. Multi^{2}Test

Our proposed method is based on MultiTest [7] which we review first. We then discuss how it can be generalized to work on multiple data sets.

3.1. MultiTest

MultiTest [7] is a cost-conscious method which orders algorithms according to their expected error and uses their costs for breaking ties. We assume that we have a prior ordering of algorithms in terms of some cost measure. Various types of cost can be used [8], for example, the space or time complexity during training or testing, interpretability, ease of programming, etc. The actual cost measure is dependent on the application and different costs may induce different orderings.

The effect of this cost measure is that, given any two algorithms with the same expected error, we favor the simpler one in terms of the used cost measure. The result of the pairwise test overrides this prior preference, that is, we choose the more costly only if it has significantly less error.

Let us assume that we index the algorithms according to this prior order as $1, 2 \ldots L$ such that 1 is the simplest (most preferred) and L is the most costly (least preferred). A graph is formed with vertices M_j corresponding to algorithms and we place directed edges as follows: $\forall i, j, i < j$, we test if algorithm i has less or comparable expected error to j:

$$H_0 : \mu_i \leq \mu_j$$

Actually, we test if the prior preference holds. If this test rejects, we say that M_j, the costlier algorithm, is statistically significantly more accurate than M_i, and a directed edge is placed from i to j, indicating that we override the prior order. After

$L(L-1)/2$ pairwise tests, the graph has edges where the test is rejected. The number of incoming edges to a node j is the number of algorithms that are preferred over j but have higher expected error. The number of outgoing edges from a node i is the number of algorithms that are less preferred than i but have less expected error. The resulting graph need not be connected. Once this graph is constructed, a topological sort gives us the order of the algorithms.

Note that these graph operations do not significantly increase the computational and/or space complexity because the real cost is the training of the algorithms. Once the algorithms are trained and validated over data sets and these validation errors are recorded, applying the graph operations is simple in comparison.

As an example, we show the application of MultiTest to one of our example data sets, *optdigits*. The result of the pairwise tests is shown in Table 3.1. Figure 3.1 shows the directed graph when the prior ordering is based on training time (increasing from left to right). The sample execution of topological sort is shown in Figure 3.2. The resulting order after topological sort is 1: *svr*, 2: *svl*, 3: *sv2*, 4: *5nn*, 5: *mlp*, 6: *lnp*, 7: *mdt*, 8: *c45*.

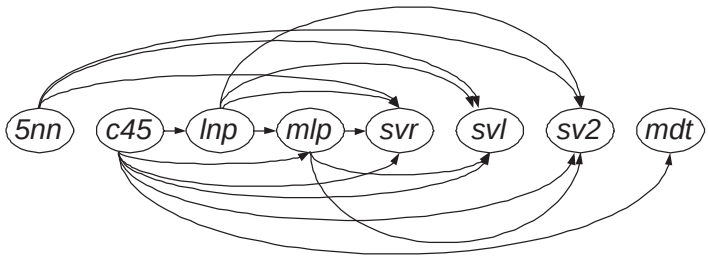


Figure 3.1. The directed graph constructed by MultiTest on *optdigits* using the pairwise test results in Table 3.1 and prior ordering based on training time (*5nn* is the fastest, *mdt* is the slowest to train.).

3.2. Multi^{2}Test

We now discuss how MultiTest can be generalized to run over multiple data sets. The pseudocode of the method is given in Table 3.2. What we do is we first apply MultiTest separately on each data set, using a pairwise test and a prior ordering based

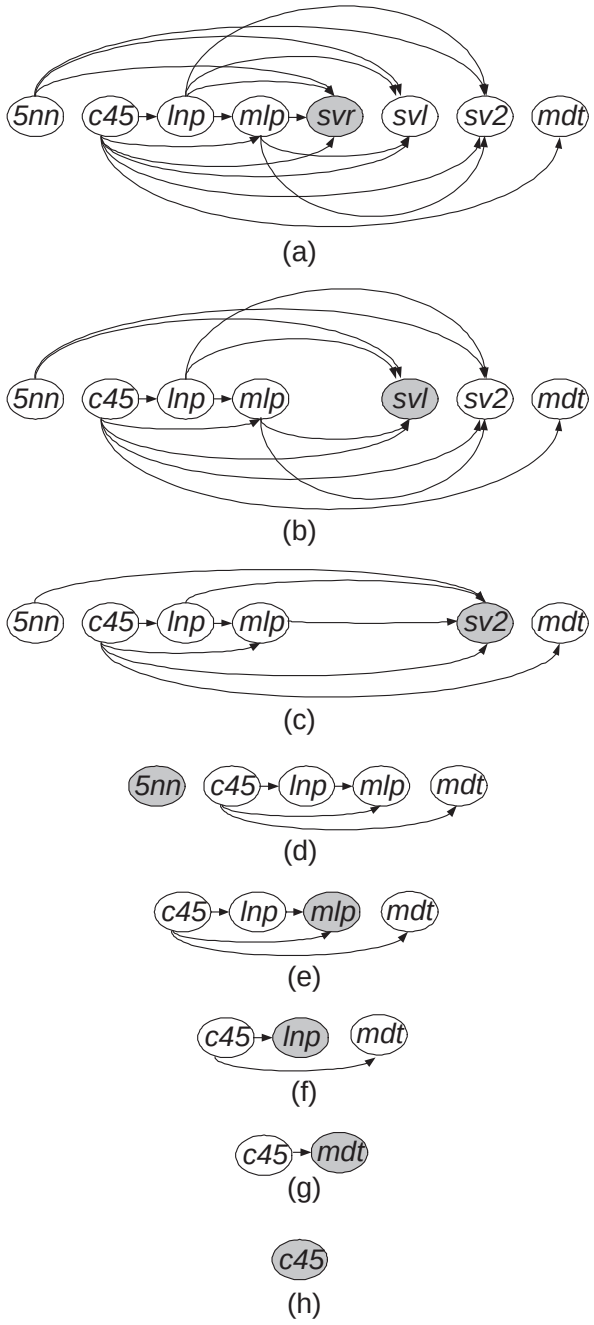


Figure 3.2. Sample execution of topological sort on the directed graph generated by MultiTest on *optdigits* using training time as prior cost. The node chosen at each iteration is shaded. (a) There are no algorithms better than *svl*, *sv2*, *svr*, and *mdt* (they do not have outgoing edges), and of them, *svr* is the simplest and is taken first. (b) After *svr* and its incident edges are removed, *svl* is the simplest. (c) Then comes *sv2*. (d) *5nn*, *mlp* and *mdt* are nodes without any outgoing edges and of the three, *5nn* is the simplest. (e) Then we choose *mlp*. (f) *lnp* is more accurate than *c45* and is simpler than *mdt*. (g) *mdt* is more accurate than *c45*, so *mdt* is selected and (h) *c45* is taken last. The resulting ranking after topological sort is 1: *svr*, 2: *svl*, 3: *sv2*, 4: *5nn*, 5: *mlp*, 6: *lnp*, 7: *mdt*, 8: *c45*.

Table 3.1. The result of pairwise tests on *optdigits*. If the entry is 1, the algorithm on the row is statistically significantly more accurate than the algorithm on the column. The algorithms are: *c45*: C4.5 decision tree, *mdt*: multivariate decision tree, *mlp*: multilayer perceptron, *lnp*: linear perceptron, *svl*: support vector machine with linear kernel, *sv2*: support vector machine with quadratic kernel, *svr*: support vector machine with radial (Gaussian) kernel, *5nn*: 5-nearest neighbor.

	c45	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
c45	0	0	0	0	0	0	0	0
mdt	**1**	0	0	0	0	0	0	0
mlp	**1**	**1**	0	**1**	0	0	0	0
lnp	**1**	0	0	0	0	0	0	0
svl	**1**	**1**	**1**	**1**	0	0	0	**1**
sv2	**1**	**1**	**1**	**1**	0	0	0	**1**
svr	**1**	**1**	**1**	**1**	0	0	0	**1**
5nn	**1**	**1**	0	**1**	0	0	0	0

on cost. We then convert the order found for each data set into ranks such that 1 is the best and L is the worst. These set of ranks are then given to Nemenyi's test which does not order the algorithms, but it gives us pairwise statistical differences which we use in MultiTest once more (thus the name Multi^{2}Test), again using the same prior ordering (this time averaged over all data sets after normalization). That is, in the outer MultiTest, the directed graph has edges provided by Nemenyi's test (which accumulates the ranks found by MultiTest separately, over all the data sets).

As an example for the second pass of Multi^{2}Test, let us assume that we have four algorithms A, B, C, D, according to the prior order of $C < A < D < B$, and the result of Nemenyi's test after the first pass of MultiTest is $\underline{A\ B}\ \underline{C\ D}$. We then convert the results of Nemenyi's test to pairwise statistically significant differences (Table 3.3) and together with the prior ordering, the formed directed graph is shown in Figure 3.3. Doing a topological sort, we find the final order as 1: A, 2: C, 3: B, 4: D.

Table 3.2. Multi^{2}Test to rank L supervised algorithms on S data sets.

Input: Cost function C, Pairwise test T, Data sets: $D_i, i = 1, \ldots, S$,
Algorithms: $M_j, j = 1, \ldots, L$, Errors: $Err_{i,j}$

Output: Ranks of algorithms

```
foreach data set D_i, i = 1 to S do
    order algorithms according to C ;
    rank algorithms using C and pairwise test T on Err with MultiTest ;
    foreach algorithm M_j, j = 1 to L do
        record rank r_ij of algorithm j for data set i ;
    end
end
apply Nemenyi's test on R_j = (1/S) Σ_i r_ij ;
calculate C̄_j, which is the average normalized cost for each algorithm over all
data sets ;
rank algorithms using C̄_j and pairwise results of Nemenyi's test with
MultiTest ;
```

Table 3.3. Tabular representation of Nemenyi's test results

	A	B	C	D
A	0	0	**1**	**1**
B	0	0	0	**1**
C	0	0	0	0
D	0	0	0	0

Figure 3.3. The directed graph constructed by MultiTest on the example problem using the pairwise test results in Table 3.3 and the prior ordering: $C < A < D < B$.

4. Results

4.1. Experimental Setup

4.1.1. Data sets

We use a total of 38 data sets where 35 of them (*zoo, iris, tae, hepatitis, wine, flags, glass, heart, haberman, flare, ecoli, bupa, ionosphere, dermatology, horse, monks, vote, cylinder, balance, australian, credit, breast, pima, tictactoe, cmc, yeast, car, segment, thyroid, optdigits, spambase, pageblock, pendigits, mushroom, and nursery*) are from UCI [1] and 3 (*titanic, ringnorm, and twonorm*) are from Delve [9] repositories.

4.1.2. Learning algorithms

We use eight algorithms:

1) *c45*: C4.5 decision tree algorithm.
2) *mdt*: Multivariate decision tree algorithm where the decision at each node is not univariate as in C4.5 but uses a linear combination of all inputs [10].
3) *mlp*: Multilayer perceptron where with D inputs and K classes, the number of hidden units is taken as $(D + K)/2$.
4) *lnp*: Linear perceptron with softmax outputs trained by gradient-descent to minimize cross-entropy.
5) *svl*: Support vector machine (SVM) with a linear kernel. We use the LIBSVM 2.82 library [11].
6) *svr*: SVM with a radial (Gaussian) kernel.
7) *sv2*: SVM with a polynomial kernel of degree 2.
8) *5nn*: k-nearest neighbor with $k = 5$.

4.1.3. Division of training, validation, and test sets

Our methodology is as follows: A data set is first divided into two parts, with 1/3 as the test set, *test*, and 2/3 as the training set, *train-all*. The training set, *train-all*, is then resampled using 5×2 cross-validation (cv) [2] where 2-fold cv is done five times (with stratification) and the roles swapped at each fold to generate ten training and validation folds, tra_i, val_i, $i = 1, \ldots, 10$. tra_i are used to train the base classifiers and the tests are run on the val_i results. We do not use the test sets in this study; in a real-world setting where we use training and validation sets to assess and compare algorithms and choose the best one (for example, using Multi^{2}Test) to estimate the generalization error of the best algorithm, the test set will be used once only after the best one is chosen. The results of our simulations are given in more detail in [12]. This processed data of base classifier outputs is publicly available [13].

4.2. The Sign Test

Table 4.1 shows the number of wins and number of losses of each algorithm over each algorithm if we use the average validation fold accuracies only (without any check for significance). The number of wins that are statistically significantly different using the sign test over 38 runs are shown in bold. We see that for example, *svl* and *svr* are significantly more accurate than the other algorithms, and *mlp* is significantly more accurate than *mdt*.

4.3. Friedman's and Nemenyi's Test

Table 4.2 shows average rankings by Friedman's and Nemenyi's tests using validation fold accuracies. The table also shows the graphical representation of post-hoc Nemenyi's test results of compared algorithms with ranks as proposed in [3]. The numbers on the line represent the average ranks, CD is the critical difference for statistical significance, and bold lines connect the algorithms which have no significant difference. We see that with respect to average accuracies, *svl* and *svr* form one group and are statistically significantly different from all other algorithms except *mlp*. *mlp* is

Table 4.1. Number of wins of all algorithms using average accuracies. The bold face entries show statistically significant difference using the sign test.

	c45	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
c45	0	19	16	16	11	17	5	15
mdt	19	0	11	16	9	18	6	18
mlp	22	**27**	0	22	9	19	7	24
lnp	22	22	16	0	8	22	8	21
svl	**26**	**29**	**29**	**30**	0	**25**	17	**31**
sv2	20	20	18	16	12	0	7	17
svr	**33**	**32**	**31**	**30**	21	**31**	0	**33**
5nn	23	20	14	17	7	21	5	0

not different from *svl*, is different from *svr*, and is not different from the other group of algorithms, namely, *mdt, c45, 5nn, sv2* and *lnp*. Nemenyi's test results shown as a table of pairwise comparisons are shown in Table 4.2 (c).

4.4. Pairwise Test

If instead of using the average accuracies (as we did in Section 4.2), we use the 5×2 cv F test for pairwise comparison ($\alpha = 0.05$), we get Table 4.3. Here, there are less wins than in Table 4.1 because to have a win, the difference should be significant. Again, wins that are significant using the sign test over 38 runs are shown in bold. We again see that *svl* and *svr* are significantly more accurate than *mdt*, *lnp*, and *5nn*; and *mlp* is significantly more accurate than *mdt*, but for example *svl* is not more accurate than *sv2* anymore.

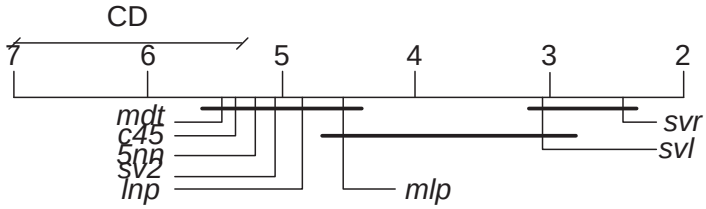
4.5. Applying Multi^{2}Test

Although the above methods give us pairwise comparisons, they cannot be used to order the given algorithms. For this, we use Multi^{2}Test. We show how it is used with two different cost functions, training time and space complexity.

Table 4.2. Average ranks and graphical representation of post-hoc Nemenyi's test results of compared algorithms with ranks using average accuracy.

c45	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
5.37	5.45	4.57	4.87	3.05	5.07	2.45	5.18

(a) Average ranks of compared algorithms



(b) Graphical representation of Nemenyi's test

	c45	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
c45	0	0	0	0	**1**	0	**1**	0
mdt	0	0	0	0	**1**	0	**1**	0
mlp	0	0	0	0	0	0	**1**	0
lnp	0	0	0	0	**1**	0	**1**	0
svl	**1**	**1**	0	**1**	0	**1**	0	**1**
sv2	0	0	0	0	**1**	0	**1**	0
svr	**1**	**1**	**1**	**1**	0	**1**	0	**1**
5nn	0	0	0	0	**1**	0	**1**	0

(c) Tabular representation of Nemenyi's test results

Table 4.3. Number of wins of all algorithms using 5×2 cv F test. The bold face entries show statistically significant difference using sign test.

	c45	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
c45	0	5	3	4	2	5	0	4
mdt	5	0	0	2	0	10	0	7
mlp	11	**7**	0	6	3	10	3	9
lnp	7	3	1	0	0	9	0	5
svl	9	**6**	4	**6**	0	13	2	**12**
sv2	7	9	8	6	5	0	1	8
svr	**14**	**14**	10	**10**	8	**16**	0	**16**
5nn	6	4	4	3	1	10	1	0

4.5.1. Training time as cost

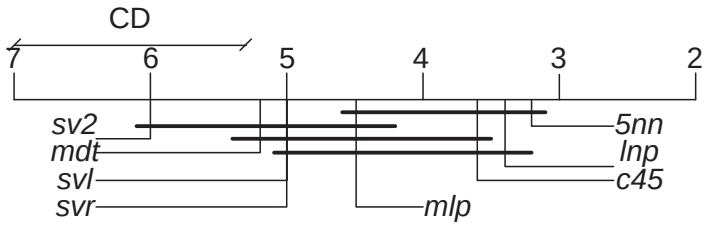
When we use the training time to define prior preferences with MultiTest, we see that the costlier support vector machine variants and *mdt* become significantly different from *5nn* (see Table 4.4 (a) and (b)). Note that this prior ordering is different for different data sets because it depends on the sample. Now, *lnp* which learns fast is significantly different from the slow *sv2* and *mdt* is significantly different from *lnp* and *c45*. There is no significant difference between *mlp, c45, lnp* and *5nn* (Table 4.4 (c)).

We still do not have an order, so we apply the second pass of Multi^{2}Test using average cost values to define the prior preference. According to the average training time, the prior order is: *5nn* $<$ *c45* $<$ *lnp* $<$ *mlp* $<$ *mdt* $<$ *svl* $<$ *sv2* $<$ *svr*. Using this prior order and the pairwise test results using Nemenyi's test results of Table 4.4 (c), gives us the graph of Table 4.4 (d), where we see that no test result overrides prior order; that is, the second MultiTest pass conforms with the accumulated first MultiTest pass on data sets separately. And therefore, the ranking is: 1: *5nn*, 2: *c45*, 3: *lnp*, 4: *mlp*, 5: *mdt*, 6: *svl*, 7: *sv2*, 8: *svr*.

Table 4.4. Average ranks and graphical representation of post-hoc Nemenyi's test results of compared algorithms used by Multi^{2}Test with training time as the cost measure.

c45	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
3.66	5.21	4.53	3.39	5.05	5.95	5.00	3.21

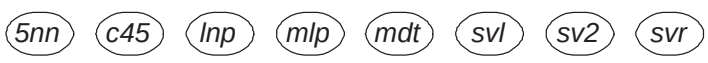
(a) Average ranks of compared algorithms



(b) Graphical representation of Nemenyi's Test

	c45	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
c45	0	0	0	0	0	**1**	0	0
mdt	0	0	0	**1**	0	0	0	**1**
mlp	0	0	0	0	0	0	0	0
lnp	0	**1**	0	0	0	**1**	0	0
svl	0	0	0	0	0	0	0	**1**
sv2	**1**	0	0	**1**	0	0	0	**1**
svr	0	0	0	0	0	0	0	**1**
5nn	0	**1**	0	0	**1**	**1**	**1**	0

(c) Tabular representation of Nemenyi's test results



(d) MultiTest graph for the second pass of Multi^{2}Test

4.5.2. Space complexity as cost

When we use the space complexity with the same validation errors, we see that, this time, *5nn* has the highest rank, and it is significantly different from *svr, lnp, c45, mdt*, and *mlp* (see Table 4.5 (a) and (b)). Also, *sv2* is significantly different from *lnp, c45, mdt*, and *mlp*; *svl* is significantly different from *c45, mdt*, and *mlp*; and there is no difference between *svr, lnp, c45, mdt*, and *mlp* (Table 4.5 (c)).

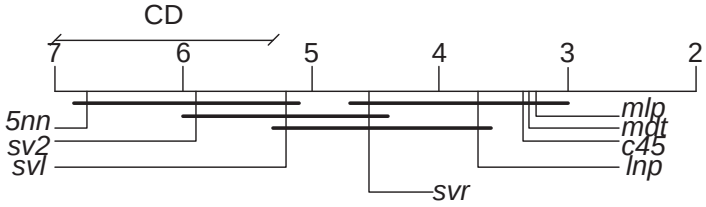
When we apply the second pass of Multi^{2}Test according to average space complexity, the prior order is: $c45 < mdt < mlp < lnp < svl < svr < sv2 < 5nn$. Using this and the pairwise test results using Nemenyi's test results of Table 4.5 (c), we get the graph of Table 4.5 (d). So the ranking is: 1: *c45*, 2: *mdt*, 3: *mlp*, 4: *lnp*, 5: *svl*, 6: *svr*, 7: *sv2*, 8: *5nn*. We see that *5nn*, which is the best when training time is critical becomes the worst when space complexity is used.

One may argue that it is useless to apply the second pass of MultiTest, but this is not always the case. We have relatively accurate classifiers and the classifiers do not span a large range of accuracy and the diversity is small. We would expect different orderings going from one data set to another if the classifiers were more diverse and the range spanned by the accuracies of the classifiers were larger. We can construct an example where this is the case: Suppose that we have three algorithms A, B, C according to the prior order of $A < B < C$ and suppose that the result of the range test is $C\underline{A\ B}$. The final order will be 1: C, 2: A, 3: B, which is different from the prior order which is A, B, C. If we choose A as *c45*, B as *mdt* and C as *svr*, and use *breast, car, nursery, optdigits, pendigits, ringnorm, spambase, tictactoe*, and *titanic* data sets only, this is what we get using real data. We see that the average prior order is $c45 < mdt < svr$, but the result of Multi^{2}Test is 1: *svr*, 2: *c45*, 3: *mdt* which is different from the prior order.

Table 4.5. Average ranks and graphical representation of post-hoc Nemenyi's test results of compared algorithms used by Multi^{2}Test with space complexity as the cost measure.

c45	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
3.34	3.32	3.29	3.71	5.18	5.87	4.55	6.74

(a) Average ranks of compared algorithms



(b) Graphical representation of Nemenyi's Test

	c45	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
c45	0	0	0	0	**1**	**1**	0	**1**
mdt	0	0	0	0	**1**	**1**	0	**1**
mlp	0	0	0	0	**1**	**1**	0	**1**
lnp	0	0	0	0	0	**1**	0	**1**
svl	**1**	**1**	**1**	0	0	0	0	0
sv2	**1**	**1**	**1**	**1**	0	0	0	0
svr	0	0	0	0	0	0	0	**1**
5nn	**1**	**1**	**1**	**1**	0	0	**1**	0

(c) Tabular representation of Nemenyi's test results

(d) MultiTest graph for the second pass of Multi^{2}Test

5. Conclusions

We propose a statistical method, Multi^{2}Test, a generalization of our previous work, which compares and orders multiple supervised learning algorithms over multiple data sets. Existing methods in the literature can find statistical differences between two algorithms, or find subset of algorithms with comparable error but our proposed method compares and orders the given algorithms, and allows, for example, to choose the best of an arbitrary number of algorithms over an arbitrary number of data sets.

The ranking produced by Multi^{2}Test uses the generalization error of algorithms and a prior cost measure. This prior preference allows ordering algorithms whose expected errors do not have a statistically significant difference between them. This is a realistic setting in that in real life, in many applications, error is not the sole criterion but some measure of cost is also critical. We see that the ranking produced by Multi^{2}Test uses this cost measure effectively.

Our implementation uses the 5×2 cv F test and is for comparing classification algorithms. One could use other pairwise tests or other resampling schemes. For example, 10×10 folding [5] will have the advantage of decreasing Type I and II errors but will increase the computational complexity. MultiTest and Multi^{2}Test are statistical frameworks, and any resampling method and pairwise statistical test could be used. For example, the same methodology can be used to compare regression algorithms over a single or multiple data sets.

There are also statisticians [14, 15] who are of the opinion that too much emphasis is given to significance testing; we believe that when used carefully and results interpreted cautiously, statistical testing, in showing that differences are not due to random chance, is useful and informative.

Note that MultiTest or Multi^{2}Test do not significantly increase the overall computational and/or space complexity because the real cost is the training and validation

of the algorithms. Once the algorithms are trained and validated over data sets and these validation errors are recorded, applying the calculations necessary for MultiTest or Multi^{2}Test is simple in comparison.

Acknowledgments

We would like to thank Mehmet Gönen for discussions. This work has been supported by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (EA-TÜBA-GEBİP/2001-1-1), Boğaziçi University Scientific Research Project 07HA101, Turkish Scientific Technical Research Council (TÜBİTAK) EEEAG 107E127 and 107E222.

APPENDIX A: EXPERIMENTAL DETAILS

A.1. 5×2 cv F test

We use the 5×2 cv F test as the pairwise test to compare the error rates of two algorithms [4]. The test uses 5×2 cross-validation [2]. Let $p_i^{(j)}$ is the difference between the errors on replication $i = 1, 2$ of fold $j = 1, \ldots, 5$: $p_i^{(j)} = e1_i^{(j)} - e2_i^{(j)}$ where $e1$ and $e2$ are the errors of two algorithms. s_i^2 is the estimated variance of the replication i: $s_i^2 = (p_i^{(1)} - \overline{p_i})^2 + (p_i^{(2)} - \overline{p_i})^2$, where $\overline{p_i}$ is the average of the differences on replication i: $\overline{p_i} = (p_i^{(1)} + p_i^{(2)})/2$. Under the null hypothesis that two algorithms have equal error rate, $p_i^{(j)}/\sigma$ is unit normal and assuming that $p_i^{(1)}$ and $p_i^{(2)}$ are independent, s_i^2 chi-square distributed with one degree of freedom. Assuming also that the s_i^2 are independent,

$$M = \frac{\sum_{i=1}^{5} s_i^2}{\sigma^2}$$

is chi-square with five degrees of freedom. Similarly,

$$N = \frac{\sum_{i=1}^{5} \sum_{j=1}^{2} p_{ij}^2}{\sigma^2}$$

is chi-square with ten degrees of freedom. Then,

$$f = \frac{N/10}{M/5} = \frac{\sum_{i=1}^{5} \sum_{j=1}^{2} p_{ij}^2}{2 \sum_{i=1}^{5} s_i^2}$$

is F distributed with ten and five degrees of freedom. We reject the hypothesis that the algorithms have the same error rate with significance α if $f > F_{\alpha,10,5}$.

A.2. Learning algorithms

The posterior probabilities of the classifiers are calculated as follows: For SVM, since we use LibSVM, we use the posterior probabilities provided by the program which

are calculated as follows: Given the estimated pairwise class probabilities r_{ij}, the goal is to estimate multiclass posterior probabilities p_i such that $\frac{p_i}{p_i+p_j} \approx r_{ij}$ or $\sum_{j:j\neq i} r_{ji}p_i \approx \sum_{j:j\neq i} r_{ij}p_j$. These equations are reformulated as an optimization problem and solved by a simple iterative method [16]. For k-nn and trees, the posterior probability is the number of instances with that label divided by the total number of instances (k or in that leaf). For *mlp* and *lnp*, the output units have softmax nonlinearity. The parameters of *svm* were found by a simple search on the validation set. For margin tradeoff (C) of *svm*, we used the range 10^{-5} to 10^5 and for width we used 10^{-3} to 10^3. For *mlp*, we used softmax and classic backpropogation with no momentum, learning rate = 0.01 and 50 epochs and learning rate is decreased by 0.95 at every epoch using online learning.

A.3. Accuracies And Complexities Of Classifiers

In this chapter we present the 5×2 validation accuracies of all classifiers over all data sets. We can see the accuracies of all classifiers using *c45*, *mdt*, *mlp*, *lnp*, *svl*, *sv2*, *svr*, and *5nn* in Tables A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8.

Table A.9 and Table A.10 show the normalized training time and normalized space complexities of algorithms on all data sets.

A.4. Results Of Statistical Tests

Table A.11 shows rankings by Friedman and Nemenyi tests using validation fold accuracies.

We see the ranks of classification algorithms using training time in Table A.12.

We see the ranks of classification algorithms using space complexity in Table A.13.

Table A.1. 5×2 validation accuracies for *c45*

Data set	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	5-1	5-2	Mean	Std
australian	85.22	86.90	90.00	82.10	90.43	81.66	87.39	84.72	84.78	84.28	85.75	2.95
balance	69.23	46.15	59.62	67.31	62.50	65.87	46.15	60.10	65.38	59.13	60.14	8.11
breast	96.57	95.69	91.85	92.67	92.70	94.83	88.84	93.10	90.13	93.10	92.95	2.37
bupa	58.26	57.89	58.26	57.89	67.83	63.16	58.26	57.89	58.26	63.16	60.09	3.44
car	81.25	83.83	83.68	81.22	83.68	81.39	85.07	84.87	83.85	82.78	83.16	1.44
cmc	51.12	45.31	44.60	44.29	42.77	52.04	46.44	55.10	53.56	42.65	47.79	4.69
credit	80.87	87.34	85.65	85.59	83.48	87.77	81.74	87.34	85.22	85.15	85.01	2.34
cylinder	67.78	72.22	65.00	67.22	62.78	71.67	70.00	66.67	56.11	62.78	66.22	4.84
dermatology	98.35	86.67	89.26	91.67	95.04	92.50	90.91	93.33	92.56	90.00	92.03	3.22
ecoli	81.58	82.24	77.19	76.64	72.81	83.18	72.81	79.44	72.81	79.44	77.81	4.01
flags	60.94	58.73	60.94	65.08	64.06	60.32	64.06	61.90	54.69	61.90	61.26	3.02
flare	88.89	89.62	88.89	89.62	88.89	89.62	88.89	89.62	88.89	89.62	89.26	0.39
glass	61.97	76.81	53.52	55.07	60.56	69.57	70.42	63.77	66.20	71.01	64.89	7.36
haberman	73.53	73.53	73.53	73.53	73.53	73.53	59.80	73.53	75.49	73.53	72.35	4.45
heart	80.00	74.44	67.78	71.11	72.22	76.67	77.78	72.22	76.67	74.44	74.33	3.61
hepatitis	75.00	70.59	78.85	80.39	78.85	80.39	73.08	80.39	78.85	80.39	77.68	3.53
horse	63.11	77.05	72.13	63.11	72.13	63.11	67.21	81.15	70.49	69.67	69.92	6.09
iris	88.89	93.16	92.31	85.47	92.31	94.02	88.03	84.62	82.91	92.31	89.40	3.99
ionosphere	98.04	81.25	94.12	95.83	96.08	93.75	90.20	93.75	94.12	95.83	93.30	4.71
monks	79.86	73.61	83.33	100.00	90.28	80.56	91.67	74.31	74.31	93.06	84.10	9.20
mushroom	99.93	99.70	99.67	99.96	99.70	99.93	99.67	99.89	99.59	99.78	99.78	0.13
nursery	92.48	92.92	92.82	93.17	92.45	92.98	93.68	92.41	92.43	92.57	92.79	0.41
optdigits	82.59	80.63	81.73	80.63	82.04	83.94	82.59	83.39	81.80	82.76	82.21	1.07
pageblock	94.68	96.49	95.01	96.43	94.96	95.72	96.05	95.61	96.82	95.01	95.68	0.75
pendigits	90.08	89.82	91.08	91.98	90.36	89.90	90.36	90.18	89.56	89.30	90.26	0.78
pima	73.44	77.65	72.27	68.24	75.39	74.12	70.31	65.10	74.22	65.10	71.58	4.29
ringnorm	86.70	87.59	85.44	86.78	84.63	85.77	85.36	84.55	85.97	86.17	85.90	0.96
segment	93.64	90.91	91.30	93.77	92.21	91.56	92.21	91.69	92.21	93.12	92.26	0.97
spambase	88.65	89.30	90.41	89.82	87.15	88.71	89.82	88.06	87.74	87.28	88.70	1.13
tae	34.00	42.86	34.00	32.65	44.00	46.94	34.00	36.73	46.00	32.65	38.38	5.86
thyroid	98.50	98.50	98.50	98.17	98.39	98.60	97.11	98.71	98.50	98.39	98.34	0.45
tictactoe	77.19	76.10	75.94	75.47	78.75	68.24	69.06	69.50	73.44	73.58	73.73	3.66
titanic	76.43	77.63	76.16	78.58	78.61	75.31	78.34	75.31	78.07	78.04	77.25	1.32
twonorm	79.72	79.89	80.17	79.72	81.02	81.67	82.32	80.45	81.55	80.74	80.73	0.90
vote	95.17	95.17	95.86	94.48	95.17	95.86	93.10	97.24	94.48	95.86	95.24	1.10
wine	88.33	87.93	86.67	82.76	88.33	77.59	85.00	89.66	91.67	79.31	85.72	4.56
yeast	42.42	50.31	52.93	45.21	46.46	55.80	47.47	47.25	53.74	53.56	49.52	4.39
zoo	76.47	76.67	73.53	86.67	82.35	86.67	82.35	86.67	64.71	76.67	79.27	7.07

Table A.2. 5×2 validation accuracies for *mdt*

Data set	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	5-1	5-2	Mean	Std
australian	83.48	85.15	90.43	83.41	89.57	82.53	87.39	83.84	86.09	84.28	85.62	2.71
balance	86.06	85.10	85.10	88.94	83.65	85.58	91.83	87.98	87.50	82.69	86.44	2.69
breast	97.00	95.69	96.14	96.55	97.42	96.12	97.00	96.55	97.85	96.55	96.69	0.65
bupa	58.26	57.89	58.26	67.54	63.48	57.89	72.17	63.16	68.70	61.40	62.88	5.13
car	90.63	89.39	91.84	88.87	94.27	91.48	88.37	91.30	90.28	90.78	90.72	1.69
cmc	47.25	42.65	43.38	46.12	43.99	48.16	42.77	47.96	45.82	50.00	45.81	2.55
credit	83.04	87.34	86.96	81.22	84.35	86.90	81.30	86.90	86.52	82.53	84.71	2.50
cylinder	70.00	66.11	67.78	67.78	73.89	71.11	67.22	68.33	57.78	57.78	66.78	5.24
dermatology	95.87	96.67	86.78	95.83	91.74	96.67	90.91	95.83	95.87	91.67	93.78	3.33
ecoli	81.58	85.98	81.58	79.44	77.19	71.96	82.46	84.11	78.95	85.05	80.83	4.17
flags	40.63	31.75	39.06	44.44	48.44	46.03	46.88	47.62	50.00	46.03	44.09	5.49
flare	88.89	89.62	88.89	83.02	88.89	89.62	88.89	89.62	88.89	89.62	88.60	1.99
glass	63.38	69.57	67.61	57.97	61.97	66.67	47.89	57.97	67.61	55.07	61.57	6.84
haberman	78.43	74.51	72.55	73.53	73.53	72.55	67.65	73.53	73.53	73.53	73.33	2.61
heart	87.78	75.56	81.11	76.67	77.78	81.11	76.67	86.67	85.56	75.56	80.44	4.75
hepatitis	78.85	80.39	78.85	80.39	78.85	76.47	78.85	80.39	78.85	80.39	79.23	1.24
horse	80.33	75.41	87.70	75.41	83.61	77.87	84.43	76.23	79.51	81.97	80.25	4.17
iris	82.05	84.62	88.03	85.47	83.76	87.18	83.76	85.47	81.20	92.31	85.38	3.21
ionosphere	96.08	97.92	94.12	100.00	94.12	97.92	96.08	97.92	94.12	97.92	96.62	2.04
monks	72.22	75.00	77.08	73.61	78.47	73.61	71.53	62.50	74.31	73.61	73.19	4.30
mushroom	97.67	98.30	96.97	97.52	98.08	97.01	98.19	98.41	98.49	97.60	97.82	0.55
nursery	92.66	92.54	92.45	93.08	93.42	92.94	91.92	93.08	92.10	92.96	92.72	0.47
optdigits	92.00	93.78	92.08	92.91	93.49	93.23	92.78	94.72	93.57	94.41	93.30	0.89
pageblock	93.31	94.57	95.50	95.17	94.41	95.50	93.86	93.91	94.35	95.01	94.56	0.74
pendigits	96.92	93.75	95.44	95.19	95.12	96.07	95.88	94.23	95.20	96.31	95.41	0.95
pima	76.95	76.08	73.44	81.57	65.23	74.12	73.44	80.78	80.08	72.55	75.42	4.86
ringnorm	77.17	74.82	75.22	75.91	75.71	76.07	76.32	75.95	75.02	76.48	75.87	0.71
segment	87.66	90.26	90.00	88.70	89.61	87.27	90.26	90.52	89.48	90.91	89.47	1.22
spambase	88.52	88.91	90.08	89.95	90.28	89.30	89.56	89.04	89.82	89.69	89.52	0.56
tae	34.00	42.86	34.00	46.94	36.00	36.73	34.00	32.65	38.00	32.65	36.78	4.71
thyroid	97.22	97.96	97.22	97.85	97.54	98.17	98.07	98.28	98.50	97.85	97.87	0.43
tictactoe	63.75	71.07	70.63	65.41	69.69	66.98	66.25	64.47	65.31	68.55	67.21	2.62
titanic	76.29	77.63	75.20	77.49	78.20	75.58	78.61	76.67	77.38	77.90	77.10	1.12
twonorm	98.05	97.53	98.05	97.73	97.65	98.09	97.65	97.89	97.81	97.97	97.84	0.20
vote	96.55	94.48	95.17	94.48	93.10	92.41	92.41	95.86	87.59	91.03	93.31	2.64
wine	95.00	89.66	98.33	94.83	95.00	96.55	95.00	91.38	96.67	93.10	94.55	2.57
yeast	48.69	50.10	52.73	46.44	50.91	53.56	49.29	54.99	48.48	50.31	50.55	2.59
zoo	58.82	83.33	73.53	63.33	61.76	70.00	55.88	80.00	58.82	73.33	67.88	9.53

Table A.3. 5×2 validation accuracies for *mlp*

Data set	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	5-1	5-2	Mean	Std
australian	83.91	80.79	87.39	82.53	90.00	83.84	89.13	85.15	81.74	85.59	85.01	3.08
balance	90.87	88.94	91.35	92.79	91.35	88.94	91.35	96.63	90.87	89.90	91.30	2.21
breast	98.28	96.98	97.00	97.84	97.85	97.84	97.42	98.28	98.28	97.84	97.76	0.49
bupa	58.26	57.89	58.26	55.26	58.26	57.89	58.26	57.89	58.26	57.89	57.81	0.91
car	93.40	91.65	91.15	89.57	92.01	89.04	93.40	92.52	92.53	92.70	91.80	1.49
cmc	51.53	52.45	49.08	51.22	47.25	49.59	47.66	51.02	47.25	47.96	49.50	1.95
credit	82.61	86.03	84.78	78.17	83.91	86.03	83.91	85.59	87.39	83.41	84.18	2.56
cylinder	68.33	67.22	63.89	66.11	71.67	69.44	68.89	70.00	68.89	66.11	68.06	2.26
dermatology	98.35	94.17	95.87	97.50	90.91	96.67	96.69	94.17	96.69	92.50	95.35	2.35
ecoli	72.81	78.50	74.56	75.70	74.56	77.57	76.32	77.57	72.81	80.37	76.08	2.47
flags	59.38	65.08	59.38	47.62	57.81	52.38	59.38	49.21	60.94	60.32	57.15	5.57
flare	88.89	89.62	88.89	85.85	88.89	89.62	88.89	89.62	89.81	89.62	88.97	1.16
glass	46.48	66.67	53.52	59.42	47.89	56.52	53.52	56.52	56.34	46.38	54.33	6.32
haberman	74.51	73.53	74.51	73.53	73.53	73.53	74.51	73.53	73.53	74.51	73.92	0.51
heart	83.33	77.78	90.00	78.89	78.89	76.67	78.89	77.78	78.89	80.00	80.11	3.90
hepatitis	82.69	78.43	78.85	88.24	78.85	80.39	80.77	84.31	78.85	78.43	80.98	3.23
horse	75.41	77.05	84.43	81.97	86.07	77.87	80.33	84.43	83.61	83.61	81.48	3.63
iris	88.03	82.05	88.89	84.62	88.03	82.91	86.32	87.18	87.18	90.60	86.58	2.67
ionosphere	94.12	93.75	90.20	87.50	94.12	93.75	92.16	85.42	68.63	93.75	89.34	7.89
monks	75.69	72.22	74.31	68.75	72.22	66.67	76.39	64.58	65.28	71.53	70.76	4.25
mushroom	100.00	99.89	99.78	100.00	99.93	99.93	99.85	100.00	99.67	99.96	99.90	0.11
nursery	98.33	99.42	98.89	99.28	99.51	99.47	99.77	99.77	99.35	99.33	99.31	0.43
optdigits	95.84	95.83	96.47	96.69	96.63	95.83	95.76	97.09	96.39	96.69	96.32	0.47
pageblock	94.52	96.05	95.23	95.66	96.71	94.68	95.12	94.46	96.22	95.99	95.46	0.78
pendigits	97.96	97.35	97.16	97.80	96.88	96.91	97.68	97.39	97.64	97.07	97.38	0.38
pima	78.13	78.04	73.05	82.35	73.05	75.69	76.17	76.08	77.73	75.29	76.56	2.73
ringnorm	89.62	90.31	89.58	88.77	90.43	88.48	91.28	90.19	91.20	89.78	89.96	0.92
segment	87.53	86.23	84.81	91.43	89.35	89.35	90.39	90.78	86.49	88.57	88.49	2.17
spambase	92.11	92.95	92.63	90.48	93.15	92.50	93.61	92.17	92.04	92.24	92.39	0.84
tae	32.00	38.78	34.00	44.90	38.00	44.90	40.00	36.73	38.00	28.57	37.59	5.17
thyroid	98.18	98.50	98.18	97.96	97.75	98.17	97.75	97.53	97.64	97.85	97.95	0.30
tictactoe	95.63	92.14	95.94	95.28	94.38	96.23	96.88	97.48	97.19	97.17	95.83	1.62
titanic	76.29	78.85	76.16	78.04	78.61	77.49	77.25	74.90	77.93	77.35	77.29	1.21
twonorm	97.61	97.32	97.73	97.04	97.69	97.93	97.28	97.93	97.73	97.65	97.59	0.29
vote	94.48	92.41	90.34	95.86	95.17	97.24	93.10	95.86	95.17	91.72	94.14	2.16
wine	98.33	91.38	86.67	94.83	91.67	94.83	100.00	94.83	95.00	98.28	94.58	3.93
yeast	51.11	56.01	53.54	57.03	55.15	52.34	51.92	54.79	50.10	51.32	53.33	2.33
zoo	55.88	60.00	58.82	60.00	61.76	60.00	55.88	63.33	50.00	63.33	58.90	4.06

Table A.4. 5×2 validation accuracies for *lnp*

Data set	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	5-1	5-2	Mean	Std
australian	82.17	82.97	83.48	77.73	84.35	79.04	86.52	82.97	84.35	82.10	82.57	2.57
balance	88.94	88.46	91.83	85.58	91.83	91.35	87.98	92.31	85.58	89.42	89.33	2.50
breast	97.85	97.41	98.28	97.41	98.28	97.84	98.28	97.41	98.28	97.84	97.89	0.38
bupa	63.48	64.91	70.43	63.16	60.00	62.28	69.57	66.67	51.30	61.40	63.32	5.41
car	89.06	83.83	89.41	87.65	86.98	86.43	88.72	87.48	86.81	88.52	87.49	1.63
cmc	46.84	46.73	45.62	43.47	44.81	43.67	45.62	48.16	47.66	47.14	45.97	1.62
credit	78.26	85.15	82.61	76.42	81.30	78.17	82.61	83.84	85.65	80.35	81.44	3.11
cylinder	73.89	67.22	69.44	69.44	76.11	66.11	70.00	69.44	70.56	67.78	70.00	3.01
dermatology	98.35	92.50	98.35	98.33	93.39	98.33	95.87	95.83	99.17	97.50	96.76	2.30
ecoli	82.46	82.24	84.21	85.05	79.82	86.92	79.82	89.72	76.32	85.98	83.25	3.95
flags	56.25	46.03	67.19	65.08	56.25	44.44	59.38	42.86	53.13	58.73	54.93	8.36
flare	88.89	85.85	86.11	86.79	87.96	78.30	84.26	85.85	87.04	83.96	85.50	2.94
glass	66.20	69.57	60.56	62.32	59.15	57.97	60.56	63.77	73.24	50.72	62.41	6.31
haberman	76.47	75.49	73.53	77.45	78.43	71.57	72.55	75.49	75.49	76.47	75.29	2.16
heart	88.89	72.22	85.56	74.44	81.11	78.89	76.67	76.67	83.33	81.11	79.89	5.14
hepatitis	78.85	72.55	80.77	82.35	78.85	86.27	78.85	88.24	80.77	76.47	80.40	4.53
horse	80.33	70.49	78.69	74.59	85.25	84.43	76.23	84.43	79.51	80.33	79.43	4.71
iris	87.18	84.62	87.18	84.62	81.20	90.60	84.62	89.74	88.03	88.03	86.58	2.82
ionosphere	90.20	91.67	88.24	100.00	90.20	87.50	86.27	95.83	82.35	91.67	90.39	4.94
monks	68.75	61.11	68.06	72.92	65.97	60.42	68.06	61.11	65.97	64.58	65.69	4.00
mushroom	100.00	99.89	99.93	100.00	99.89	99.93	99.85	100.00	99.89	100.00	99.94	0.06
nursery	90.30	90.88	90.39	91.02	91.66	90.55	91.64	90.00	91.85	90.65	90.89	0.64
optdigits	95.37	93.94	93.57	93.46	94.27	93.07	94.35	94.65	94.27	94.96	94.19	0.70
pageblock	94.30	96.21	96.05	95.28	95.67	95.06	95.67	96.38	96.00	95.01	95.56	0.65
pendigits	94.72	94.59	94.80	95.55	94.96	95.35	95.12	94.75	95.28	94.99	95.01	0.31
pima	74.22	74.90	67.58	79.22	74.61	75.29	74.22	78.04	72.66	74.51	74.52	3.11
ringnorm	71.78	71.13	72.26	74.37	69.83	69.51	71.57	74.66	72.99	72.95	72.10	1.71
segment	88.96	88.31	90.52	85.45	91.17	85.71	90.00	88.44	89.87	87.53	88.60	1.93
spambase	90.48	91.45	89.95	91.45	90.28	88.13	91.59	89.56	90.74	91.78	90.54	1.13
tae	56.00	40.82	42.00	38.78	48.00	38.78	34.00	53.06	44.00	38.78	43.42	6.95
thyroid	97.11	98.17	97.75	97.85	97.75	97.42	97.97	97.21	97.43	97.85	97.65	0.34
tictactoe	97.81	97.48	97.50	96.86	96.88	97.80	97.81	92.45	98.13	97.48	97.02	1.66
titanic	75.75	74.76	76.29	78.04	77.38	73.12	77.25	75.03	76.43	76.53	76.06	1.45
twonorm	98.13	97.49	98.01	97.53	97.81	97.24	97.40	97.40	97.57	97.45	97.60	0.29
vote	94.48	96.55	92.41	95.17	91.72	94.48	93.10	93.79	95.17	92.41	93.93	1.52
wine	95.00	91.38	98.33	94.83	93.33	93.10	98.33	94.83	96.67	94.83	95.06	2.22
yeast	51.11	54.79	53.33	53.16	54.14	53.97	51.92	50.51	53.74	51.93	52.86	1.41
zoo	88.24	86.67	88.24	86.67	88.24	90.00	85.29	90.00	85.29	90.00	87.86	1.83

Table A.5. 5×2 validation accuracies for *svl*

Data set	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	5-1	5-2	Mean	Std
australian	85.65	86.46	90.00	84.28	91.30	81.66	87.83	84.72	88.26	86.03	86.62	2.84
balance	93.75	90.38	92.79	90.87	90.38	95.19	92.31	91.83	87.98	91.83	91.73	2.00
breast	98.28	98.28	97.85	98.28	98.28	97.84	97.42	98.28	98.28	97.41	98.02	0.36
bupa	58.26	64.91	58.26	67.54	66.96	61.40	65.22	61.40	58.26	65.79	62.80	3.72
car	92.36	92.35	91.67	89.74	92.19	91.30	93.06	92.00	91.84	92.52	91.90	0.90
cmc	46.03	44.29	45.01	45.31	47.05	43.27	43.99	46.53	46.64	44.69	45.28	1.26
credit	83.04	87.77	87.39	84.28	86.09	87.34	83.91	91.27	87.39	84.28	86.28	2.47
cylinder	70.00	72.78	71.67	71.67	73.89	73.89	79.44	72.78	73.33	68.33	72.78	2.93
dermatology	99.17	96.67	97.52	98.33	93.39	98.33	97.52	99.17	100.00	97.50	97.76	1.83
ecoli	62.28	87.85	65.79	85.05	66.67	86.92	64.91	82.24	64.04	87.85	75.36	11.36
flags	64.06	60.32	64.06	71.43	68.75	60.32	65.63	65.08	64.06	61.90	64.56	3.49
flare	88.89	89.62	88.89	89.62	88.89	89.62	88.89	89.62	88.89	89.62	89.26	0.39
glass	61.97	63.77	57.75	71.01	59.15	60.87	70.42	60.87	66.20	57.97	63.00	4.81
haberman	74.51	73.53	74.51	73.53	73.53	74.51	76.47	73.53	75.49	73.53	74.31	1.01
heart	86.67	85.56	87.78	80.00	83.33	82.22	80.00	87.78	86.67	83.33	84.33	2.98
hepatitis	84.62	80.39	84.62	84.31	82.69	86.27	80.77	84.31	84.62	82.35	83.50	1.88
horse	86.89	86.89	88.52	86.07	88.52	86.07	86.89	86.07	88.52	86.89	87.13	1.03
iris	91.45	88.03	94.02	84.62	92.31	86.32	88.89	88.03	87.18	90.60	89.15	2.91
ionosphere	96.08	95.83	94.12	100.00	96.08	97.92	96.08	95.83	94.12	97.92	96.40	1.79
monks	74.31	62.50	67.36	68.06	66.67	68.75	69.44	65.28	65.97	68.06	67.64	3.07
mushroom	100.00	99.89	99.89	100.00	99.93	100.00	99.85	100.00	100.00	99.96	99.95	0.06
nursery	92.29	92.68	92.87	92.61	92.31	92.50	92.17	92.75	92.54	92.52	92.53	0.22
optdigits	97.73	98.11	97.88	97.72	97.41	97.32	97.73	98.27	97.65	97.72	97.75	0.28
pageblock	94.96	95.83	96.27	95.44	95.39	94.68	95.39	95.55	96.60	95.17	95.53	0.58
pendigits	95.56	95.11	94.76	96.15	97.08	95.15	95.48	95.63	95.80	94.87	95.56	0.68
pima	76.95	79.22	74.61	81.96	78.13	77.25	76.17	81.57	79.69	75.69	78.12	2.46
ringnorm	75.83	75.59	74.86	76.07	75.14	76.28	75.79	76.12	74.49	76.80	75.70	0.70
segment	86.23	73.64	75.19	87.66	75.71	70.00	87.53	75.19	81.82	70.78	78.38	6.84
spambase	90.93	91.19	90.54	91.59	91.85	90.08	89.69	91.32	91.00	90.15	90.83	0.70
tae	44.00	44.90	42.00	51.02	46.00	46.94	38.00	44.90	40.00	32.65	43.04	5.15
thyroid	98.18	98.28	98.39	98.28	97.75	98.60	98.18	98.28	98.39	97.85	98.22	0.25
tictactoe	98.13	97.48	97.19	98.43	96.88	98.74	98.13	97.48	97.81	97.80	97.81	0.57
titanic	75.89	76.81	75.48	77.49	78.07	74.62	78.20	74.76	76.29	76.40	76.40	1.26
twonorm	98.22	97.77	98.05	97.93	97.85	98.13	97.57	98.13	97.81	98.05	97.95	0.20
vote	95.86	97.93	93.79	96.55	95.86	95.86	93.79	96.55	95.86	96.55	95.86	1.26
wine	98.33	93.10	98.33	94.83	96.67	96.55	100.00	96.55	98.33	98.28	97.10	2.01
yeast	54.14	58.66	54.95	58.04	54.55	54.79	55.96	56.42	57.37	56.62	56.15	1.55
zoo	88.24	90.00	82.35	86.67	82.35	90.00	82.35	86.67	88.24	86.67	86.35	3.02

Table A.6. 5×2 validation accuracies for *sv2*

Data set	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	5-1	5-2	Mean	Std
australian	70.00	73.80	75.22	70.74	72.17	68.12	73.04	72.49	72.17	72.49	72.03	2.00
balance	74.04	76.44	76.44	77.88	76.44	76.92	75.96	74.52	79.33	77.88	76.59	1.57
breast	93.13	94.40	94.85	94.83	96.14	94.40	93.99	94.40	95.28	92.67	94.41	1.00
bupa	58.26	58.77	58.26	57.89	58.26	57.89	58.26	57.89	59.13	57.89	58.25	0.42
car	96.88	96.00	97.74	93.57	97.22	96.17	96.01	96.52	94.79	96.52	96.14	1.20
cmc	45.82	43.88	44.81	43.06	42.77	42.65	44.60	43.27	44.60	45.31	44.08	1.11
credit	68.70	74.24	69.57	70.74	73.48	75.11	73.91	75.11	72.61	71.62	72.51	2.27
cylinder	75.56	73.33	71.67	77.22	77.22	75.56	73.33	75.00	71.11	72.78	74.28	2.16
dermatology	98.35	94.17	95.04	95.00	94.21	95.83	93.39	95.83	95.04	96.67	95.35	1.42
ecoli	62.28	83.18	64.91	82.24	64.91	86.92	59.65	81.31	63.16	84.11	73.27	11.03
flags	60.94	60.32	57.81	55.56	64.06	53.97	54.69	55.56	53.13	60.32	57.63	3.62
flare	88.89	89.62	88.89	89.62	88.89	89.62	88.89	89.62	88.89	89.62	89.26	0.39
glass	63.38	72.46	67.61	75.36	59.15	69.57	67.61	63.77	70.42	68.12	67.74	4.71
haberman	73.53	73.53	73.53	76.47	73.53	74.51	73.53	73.53	73.53	73.53	73.92	0.95
heart	64.44	63.33	71.11	66.67	71.11	65.56	64.44	71.11	63.33	64.44	66.56	3.29
hepatitis	86.54	80.39	86.54	88.24	84.62	88.24	84.62	90.20	86.54	84.31	86.02	2.72
horse	86.89	85.25	85.25	85.25	81.97	89.34	82.79	82.79	85.25	82.79	84.75	2.26
iris	91.45	93.16	94.87	94.87	94.02	91.45	94.87	94.87	95.73	95.73	94.10	1.58
ionosphere	86.27	81.25	72.55	87.50	86.27	83.33	86.27	72.92	82.35	83.33	82.21	5.38
monks	73.61	66.67	67.36	64.58	72.22	63.89	74.31	58.33	68.75	75.00	68.47	5.38
mushroom	100.00	99.89	99.89	100.00	99.89	100.00	99.85	100.00	100.00	99.96	99.95	0.06
nursery	98.80	98.52	97.64	99.07	99.42	98.96	98.56	99.10	97.78	98.73	98.66	0.57
optdigits	97.80	98.50	98.12	98.03	97.73	97.80	97.41	97.72	98.35	97.95	97.94	0.32
pageblock	95.23	94.46	95.56	94.68	93.91	94.07	93.53	94.35	94.90	94.68	94.54	0.61
pendigits	98.84	98.80	98.84	98.96	98.92	98.76	98.96	98.84	98.24	99.04	98.82	0.22
pima	68.36	67.84	69.53	68.63	70.31	70.20	70.70	65.88	70.31	68.24	69.00	1.50
ringnorm	98.18	98.30	98.05	98.05	97.97	97.77	97.93	97.93	97.89	98.18	98.03	0.16
segment	86.49	85.06	87.01	72.60	79.35	84.16	85.19	79.35	87.14	85.84	83.22	4.69
spambase	84.54	80.76	82.97	83.50	86.04	78.67	79.06	82.39	82.13	81.93	82.20	2.29
tae	34.00	44.90	52.00	38.78	48.00	40.82	46.00	42.86	44.00	32.65	42.40	6.02
thyroid	97.64	98.17	97.86	97.85	97.64	98.39	98.39	97.96	98.18	97.64	97.97	0.30
tictactoe	75.31	71.70	75.63	76.73	76.25	74.21	70.94	73.27	74.69	74.53	74.33	1.88
titanic	77.66	78.58	77.52	78.72	78.61	77.49	79.43	77.08	78.07	78.04	78.12	0.71
twonorm	50.73	49.96	50.93	49.96	50.57	49.96	50.49	49.96	49.96	49.96	50.25	0.39
vote	91.03	90.34	91.72	91.03	91.72	91.03	88.97	92.41	88.28	90.34	90.69	1.27
wine	91.67	84.48	90.00	86.21	90.00	87.93	88.33	84.48	86.67	87.93	87.77	2.38
yeast	45.66	49.08	47.27	47.45	45.25	51.32	45.05	47.25	47.27	49.69	47.53	2.02
zoo	88.24	86.67	82.35	86.67	85.29	90.00	82.35	83.33	70.59	90.00	84.55	5.66

Table A.7. 5×2 validation accuracies for *svr*

Data set	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	5-1	5-2	Mean	Std
australian	83.04	83.41	88.70	84.28	86.52	83.84	89.13	82.97	84.78	83.84	85.05	2.28
balance	81.25	83.17	85.10	83.65	85.58	85.10	87.02	84.62	83.65	83.65	84.28	1.57
breast	97.85	97.41	97.42	98.28	98.28	97.84	97.85	98.28	98.28	96.98	97.85	0.45
bupa	71.30	67.54	65.22	67.54	62.61	65.79	64.35	70.18	68.70	68.42	67.16	2.68
car	97.57	97.04	97.92	96.00	96.70	97.74	97.40	97.74	96.53	97.04	97.17	0.62
cmc	51.32	50.20	51.12	47.14	51.12	52.04	46.44	47.14	52.55	48.37	49.74	2.26
credit	81.74	86.90	87.39	81.66	84.78	89.08	83.48	87.34	86.96	83.41	85.27	2.61
cylinder	74.44	72.78	70.56	75.56	79.44	75.00	79.44	75.00	72.22	73.33	74.78	2.88
dermatology	99.17	96.67	97.52	97.50	94.21	98.33	96.69	97.50	99.17	98.33	97.51	1.46
ecoli	63.16	91.59	67.54	88.79	66.67	85.98	65.79	84.11	65.79	88.79	76.82	11.84
flags	62.50	60.32	62.50	68.25	71.88	61.90	62.50	65.08	62.50	60.32	63.77	3.67
flare	88.89	89.62	88.89	89.62	88.89	89.62	88.89	89.62	88.89	90.57	89.35	0.56
glass	70.42	68.12	64.79	78.26	66.20	68.12	73.24	68.12	74.65	66.67	69.86	4.26
haberman	77.45	76.47	74.51	74.51	75.49	72.55	74.51	73.53	75.49	72.55	74.71	1.59
heart	85.56	83.33	87.78	80.00	82.22	81.11	82.22	86.67	86.67	82.22	83.78	2.68
hepatitis	86.54	80.39	86.54	84.31	84.62	88.24	80.77	82.35	84.62	88.24	84.66	2.82
horse	85.25	84.43	88.52	83.61	86.07	87.70	85.25	84.43	86.89	84.43	85.66	1.60
iris	94.87	96.58	97.44	93.16	96.58	96.58	94.02	98.29	97.44	94.02	95.90	1.75
ionosphere	96.08	91.67	94.12	100.00	96.08	97.92	96.08	95.83	94.12	97.92	95.98	2.34
monks	90.97	84.03	86.81	88.19	82.64	80.56	87.50	83.33	76.39	90.28	85.07	4.55
mushroom	99.96	99.82	99.89	99.96	99.93	100.00	99.85	100.00	100.00	99.96	99.94	0.07
nursery	95.39	95.60	94.70	96.04	95.90	94.77	95.53	95.44	95.42	95.88	95.47	0.45
optdigits	97.80	97.64	97.88	98.03	97.96	97.72	97.96	97.95	98.20	97.80	97.89	0.16
pageblock	94.24	95.61	95.39	95.12	95.61	94.57	94.90	94.84	95.67	95.17	95.11	0.48
pendigits	99.60	99.56	99.44	99.56	99.56	99.40	99.52	99.72	99.40	99.12	99.49	0.16
pima	75.39	77.25	73.05	81.57	75.78	75.29	74.22	79.61	76.17	74.90	76.32	2.55
ringnorm	98.99	98.66	98.95	98.78	98.78	98.78	98.82	98.74	98.70	98.91	98.81	0.10
segment	87.92	91.43	91.30	92.21	87.66	89.74	86.49	88.57	88.18	86.49	89.00	2.07
spambase	91.19	90.61	91.32	90.67	91.26	90.80	91.45	90.74	90.67	91.26	91.00	0.33
tae	48.00	44.90	44.00	46.94	46.00	44.90	42.00	53.06	46.00	36.73	45.25	4.18
thyroid	97.75	98.28	97.54	97.53	97.64	98.17	97.86	98.07	97.43	97.96	97.82	0.29
tictactoe	93.13	95.60	95.63	97.17	93.44	94.03	93.44	94.97	95.31	94.97	94.77	1.26
titanic	76.98	78.44	76.16	78.72	78.61	75.99	79.43	77.08	78.07	78.44	77.79	1.17
twonorm	98.26	97.69	98.09	97.93	97.85	98.01	97.81	98.13	97.89	98.13	97.98	0.18
vote	97.24	95.17	94.48	95.17	95.86	96.55	94.48	96.55	94.48	93.10	95.31	1.25
wine	96.67	94.83	96.67	94.83	95.00	96.55	98.33	94.83	98.33	96.55	96.26	1.37
yeast	52.93	60.69	56.16	57.84	58.59	56.42	57.17	59.67	58.99	55.19	57.37	2.30
zoo	79.41	93.33	82.35	86.67	82.35	86.67	79.41	83.33	82.35	86.67	84.25	4.20

Table A.8. 5×2 validation accuracies for *5nn*

Data set	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	5-1	5-2	Mean	Std
australian	81.30	82.97	83.48	75.98	86.52	85.15	83.04	82.10	82.17	82.97	82.57	2.77
balance	69.71	75.00	77.40	72.12	73.56	76.44	73.08	73.56	76.44	73.08	74.04	2.32
breast	98.28	96.12	97.00	97.41	96.57	97.84	97.85	97.41	97.85	97.41	97.38	0.66
bupa	56.52	65.79	57.39	58.77	64.35	46.49	61.74	59.65	62.61	57.89	59.12	5.40
car	84.72	82.61	83.33	83.30	84.90	83.65	82.29	84.70	83.16	84.35	83.70	0.92
cmc	51.32	48.16	51.93	47.96	46.64	47.35	46.64	46.12	50.51	50.41	48.70	2.14
credit	82.17	83.41	83.48	80.35	84.35	82.97	82.61	85.15	83.91	81.22	82.96	1.44
cylinder	61.67	64.44	65.56	67.78	67.22	64.44	67.78	65.00	64.44	65.00	65.33	1.87
dermatology	94.21	94.17	95.87	95.00	93.39	95.00	94.21	93.33	95.87	96.67	94.77	1.11
ecoli	82.46	88.79	79.82	83.18	82.46	85.05	80.70	82.24	81.58	85.05	83.13	2.59
flags	51.56	50.79	46.88	36.51	50.00	42.86	50.00	46.03	51.56	47.62	47.38	4.73
flare	88.89	89.62	88.89	87.74	88.89	89.62	88.89	88.68	87.96	89.62	88.88	0.65
glass	63.38	69.57	70.42	69.57	52.11	66.67	67.61	63.77	66.20	65.22	65.45	5.27
haberman	74.51	75.49	70.59	65.69	75.49	72.55	70.59	73.53	72.55	70.59	72.16	2.97
heart	87.78	86.67	82.22	77.78	80.00	77.78	77.78	85.56	81.11	85.56	82.22	3.92
hepatitis	82.69	68.63	82.69	78.43	84.62	88.24	82.69	92.16	82.69	80.39	82.32	6.18
horse	77.05	82.79	81.15	81.15	79.51	75.41	81.15	86.07	86.07	74.59	80.49	3.97
iris	71.79	83.76	76.07	81.20	84.62	79.49	78.63	82.05	76.07	83.76	79.74	4.15
ionosphere	96.08	93.75	92.16	100.00	94.12	95.83	94.12	93.75	90.20	97.92	94.79	2.80
monks	72.22	72.92	70.14	69.44	62.50	59.03	71.53	69.44	70.83	71.53	68.96	4.54
mushroom	100.00	99.96	99.89	100.00	99.93	99.93	99.85	100.00	99.93	100.00	99.95	0.05
nursery	90.76	90.16	90.62	90.07	90.81	91.13	90.62	90.69	90.78	90.14	90.58	0.35
optdigits	96.86	96.06	97.18	96.61	95.69	95.75	95.92	96.77	96.55	96.14	96.35	0.51
pageblock	95.23	96.16	96.00	95.94	96.66	95.33	95.72	95.94	95.83	95.83	95.86	0.40
pendigits	99.04	98.60	98.24	99.12	98.72	98.80	99.04	99.00	99.12	98.64	98.83	0.29
pima	71.48	75.69	71.88	77.65	69.53	70.98	75.00	68.24	71.48	73.73	72.56	2.91
ringnorm	64.48	64.03	64.03	64.23	64.52	64.44	63.42	64.31	63.22	63.87	64.06	0.44
segment	93.38	91.69	90.65	93.12	92.99	91.04	91.95	91.82	92.34	91.30	92.03	0.92
spambase	89.04	89.11	89.17	88.78	90.08	90.28	89.89	88.65	89.11	89.30	89.34	0.55
tae	44.00	46.94	50.00	40.82	26.00	46.94	36.00	26.53	34.00	40.82	39.20	8.41
thyroid	97.75	98.17	97.97	97.85	97.32	98.28	98.29	98.17	98.39	97.96	98.02	0.32
tictactoe	84.69	83.33	86.56	83.33	83.75	83.96	84.38	85.85	84.38	84.59	84.48	1.04
titanic	32.29	38.47	32.29	32.33	32.29	32.33	32.29	38.20	32.29	32.33	33.51	2.54
twonorm	97.36	96.76	97.28	97.16	96.92	97.24	96.88	97.20	96.76	97.04	97.06	0.22
vote	92.41	91.72	91.72	93.10	91.03	94.48	91.72	93.10	91.03	91.03	92.14	1.14
wine	96.67	93.10	93.33	91.38	95.00	93.10	96.67	87.93	93.33	94.83	93.53	2.58
yeast	51.92	53.36	52.53	51.12	50.10	56.01	51.52	55.19	52.12	54.79	52.87	1.92
zoo	76.47	86.67	79.41	86.67	82.35	86.67	79.41	86.67	79.41	86.67	83.04	4.07

Table A.9. Normalized average training times of algorithms

Data set	*c45*	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
australian	3.05	70.14	36.59	4.57	74.71	91.49	100.64	0.64
balance	0.61	7.32	4.88	1.22	100.09	23.19	23.80	0.09
breast	1.59	2.30	13.42	6.71	63.76	83.90	100.68	0.68
bupa	10.09	3.53	20.18	7.18	60.54	70.64	100.91	0.91
car	0.09	5.07	2.44	0.85	100.05	24.33	22.73	0.05
cmc	3.23	82.58	8.62	2.15	75.04	89.76	100.17	0.17
credit	2.79	88.01	39.11	4.19	82.42	94.99	100.58	0.58
cylinder	1.87	100.36	18.20	1.40	22.87	25.67	28.01	0.36
dermatology	1.92	101.92	28.44	7.11	16.59	42.43	42.66	2.37
ecoli	4.44	12.61	18.91	11.60	63.03	94.55	100.85	0.85
flags	0.03	100.01	1.43	0.17	0.56	0.94	0.91	0.01
flare	0.11	2.16	1.08	0.25	100.02	1.29	0.18	0.02
glass	10.70	38.02	24.30	13.30	63.37	76.05	101.39	1.39
haberman	5.86	3.53	22.20	12.19	66.61	71.63	101.11	1.11
heart	10.22	14.61	29.21	11.09	58.43	79.51	102.02	2.02
hepatitis	16.93	60.00	85.58	24.77	48.74	85.03	100.50	0.50
horse	0.75	100.26	36.66	1.50	86.05	22.45	20.95	0.26
ionosphere	29.66	81.56	66.73	7.41	74.15	88.97	103.80	3.80
iris	6.49	6.19	29.43	19.28	47.53	51.55	103.09	3.09
monks	2.05	3.00	14.48	7.38	58.81	84.01	100.81	0.81
mushroom	0.00	4.13	2.97	0.14	41.84	44.79	100.00	0.01
nursery	0.05	1.03	0.83	0.24	74.61	86.82	100.02	0.02
optdigits	0.46	100.02	4.01	0.97	27.86	49.53	14.65	0.02
pageblock	5.79	15.04	8.87	4.44	77.14	96.80	100.47	0.47
pendigits	0.64	11.49	2.52	1.18	23.77	100.02	43.73	0.02
pima	8.40	3.80	11.20	5.33	61.59	81.19	100.79	0.79
ringnorm	4.16	0.35	1.53	0.39	70.82	86.36	100.02	0.02
segment	3.87	23.60	7.93	3.09	100.19	93.62	39.26	0.19
spambase	1.66	8.00	6.81	0.60	82.52	92.65	100.05	0.05
tae	0.32	100.04	4.88	0.61	2.77	3.20	3.90	0.04
thyroid	1.17	100.44	23.51	4.08	20.59	21.18	21.56	0.44
tictactoe	2.90	35.37	19.39	3.83	99.25	85.56	100.39	0.39
titanic	0.30	0.89	5.30	2.41	54.89	77.04	100.16	0.16
twonorm	2.41	0.37	1.64	0.40	12.97	100.02	98.40	0.02
vote	5.33	37.29	52.97	10.66	58.60	101.23	53.28	1.23
wine	13.87	25.70	44.46	18.15	74.41	93.90	103.47	3.47
yeast	3.23	29.10	7.68	4.04	59.41	85.68	100.23	0.23
zoo	2.10	101.07	28.20	13.30	65.60	72.06	54.90	1.07
Avg	4.45	38.97	19.38	5.74	59.80	67.74	73.45	0.75

Table A.10. Normalized average space complexities of algorithms

Data set	*c45*	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
australian	0.02	0.81	0.26	0.48	85.90	100.02	96.49	38.59
balance	0.16	0.43	0.82	1.43	38.24	91.05	100.16	47.37
breast	0.28	0.50	0.26	0.43	66.16	61.11	62.08	100.26
bupa	0.06	0.12	0.62	0.87	72.99	74.49	73.74	100.06
car	0.73	0.42	0.36	0.60	44.36	100.36	83.13	55.09
cmc	0.29	2.89	0.19	0.33	100.19	98.37	99.55	43.66
credit	0.02	0.19	0.26	0.49	87.28	100.02	98.10	38.39
cylinder	0.19	0.23	0.34	0.66	100.19	96.31	98.25	60.71
dermatology	0.26	1.52	1.50	2.50	31.52	58.59	59.00	100.26
ecoli	2.49	2.31	6.02	3.70	78.54	78.66	73.28	102.31
flags	0.08	0.97	3.42	6.36	92.20	100.08	98.50	22.91
flare	0.03	0.06	1.34	2.30	100.03	71.35	66.20	75.31
glass	1.37	1.30	3.47	4.34	72.89	81.34	80.69	101.30
haberman	0.12	0.25	1.23	0.98	44.49	40.44	41.18	100.12
heart	0.88	0.64	0.68	1.12	39.96	73.17	55.01	100.64
hepatitis	0.49	0.05	1.07	1.94	48.88	49.81	54.42	100.05
horse	0.32	0.33	0.46	0.91	86.37	100.32	82.32	30.56
ionosphere	0.20	0.39	0.46	0.86	74.03	70.29	72.78	100.20
iris	1.43	2.45	3.68	3.07	18.81	37.63	33.54	101.43
monks	2.29	0.85	0.50	0.70	63.08	63.38	58.30	100.50
mushroom	0.03	0.24	0.08	0.16	22.90	44.74	100.03	87.17
nursery	0.24	0.07	0.03	0.05	98.94	100.02	100.03	33.34
optdigits	0.13	0.47	0.23	0.39	77.94	43.08	40.13	100.13
pageblock	0.08	0.12	0.10	0.14	17.82	18.02	16.04	100.08
pendigits	0.26	0.53	0.17	0.20	45.68	14.71	43.34	100.17
pima	0.61	1.33	0.26	0.39	62.07	62.25	62.10	100.26
ringnorm	0.21	0.03	0.02	0.04	61.32	79.54	48.38	100.02
segment	0.22	0.57	0.32	0.46	61.45	51.00	93.84	100.22
spambase	0.06	0.12	0.03	0.07	77.60	78.11	77.47	100.03
tae	1.04	1.86	1.64	3.12	101.00	101.04	99.98	10.84
thyroid	0.02	0.05	0.20	0.37	12.15	11.79	9.09	100.02
tictactoe	0.54	0.47	0.19	0.36	100.19	83.86	79.57	40.61
titanic	0.22	0.36	0.16	0.24	100.05	100.16	100.07	77.43
twonorm	0.20	0.01	0.02	0.04	10.93	95.22	55.22	100.02
vote	0.08	0.37	0.70	1.28	24.86	100.08	57.19	95.78
wine	0.43	1.76	1.64	2.55	21.32	63.18	48.18	100.43
yeast	0.66	0.63	1.13	1.02	75.38	80.64	84.82	100.63
zoo	1.78	4.02	7.85	11.12	55.33	76.26	65.79	101.78
Avg	0.49	0.78	1.10	1.48	62.45	72.38	70.21	80.75

Table A.11. Ranks of compared algorithms according to average accuracies.

Data set	*c45*	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
australian	2	3	5	7	1	8	4	6
balance	8	4	2	3	1	6	5	7
breast	8	6	4	2	1	7	3	5
bupa	5	3	8	2	4	7	1	6
car	8	5	4	6	3	2	1	7
cmc	4	6	2	5	7	8	1	3
credit	3	4	5	7	1	8	2	6
cylinder	7	6	5	4	3	2	1	8
dermatology	8	7	5	3	1	4	2	6
ecoli	4	3	6	1	7	8	5	2
flags	3	8	5	6	1	4	2	7
flare	3	7	5	8	3	3	1	6
glass	4	7	8	6	5	2	1	3
haberman	7	6	4.5	1	3	4.5	2	8
heart	7	4	5	6	1	8	2	3
hepatitis	8	7	5	6	3	1	2	4
horse	8	6	4	7	1	3	2	5
ionosphere	3	7	5	6	4	2	1	8
iris	5	1	7	6	2	8	3	4
monks	2	3	4	8	7	6	1	5
mushroom	7	8	6	4	1	3	5	2
nursery	4	5	1	7	6	2	3	8
optdigits	8	7	5	6	3	1	2	4
pageblock	2	7	5	3	4	8	6	1
pendigits	8	6	4	7	5	3	1	2
pima	7	4	2	5	1	8	3	6
ringnorm	4	5	3	7	6	2	1	8
segment	1	3	6	5	8	7	4	2
spambase	7	5	1	4	3	8	2	6
tae	6	8	7	2	3	4	1	5
thyroid	1	6	5	8	2	4	7	3
tictactoe	7	8	3	2	1	6	4	5
titanic	4	5	3	7	6	1	2	8
twonorm	7	3	5	4	2	8	1	6
vote	3	6	4	5	1	8	2	7
wine	8	5	4	3	1	7	2	6
yeast	7	6	3	5	2	8	1	4
zoo	6	7	8	1	2	3	4	5
Avg	5.37	5.45	4.57	4.87	3.05	5.07	2.45	5.18

Table A.12. Ranks of compared algorithms according to training time using MultiTest.

Data set	*c45*	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
australian	2	4	3	7	5	8	6	1
balance	7	3	2	1	4	8	5	6
breast	6	7	3	2	4	8	5	1
bupa	6	2	7	3	4	8	5	1
car	6	4	3	7	5	2	1	8
cmc	2	4	3	6	7	8	5	1
credit	2	6	4	3	5	8	7	1
cylinder	2	8	7	1	3	4	5	6
dermatology	1	8	4	2	3	5	6	7
ecoli	3	4	5	2	6	7	8	1
flags	1	8	6	2	3	5	4	7
flare	2	7	5	4	8	6	3	1
glass	5	3	6	2	8	4	7	1
haberman	3	2	5	4	6	7	8	1
heart	2	4	5	3	6	8	7	1
hepatitis	8	2	6	4	5	3	7	1
horse	7	8	6	2	5	4	3	1
ionosphere	1	8	7	6	2	3	4	5
iris	3	2	5	4	6	8	7	1
monks	1	3	6	5	7	8	2	4
mushroom	1	8	4	3	5	6	7	2
nursery	4	6	1	5	7	2	3	8
optdigits	8	7	5	6	2	3	1	4
pageblock	3	5	4	2	6	7	8	1
pendigits	8	6	4	5	7	3	1	2
pima	4	2	5	3	6	7	8	1
ringnorm	4	5	3	7	6	2	1	8
segment	3	5	4	2	8	7	6	1
spambase	6	7	2	1	3	8	4	5
tae	2	8	7	3	4	5	6	1
thyroid	1	8	7	2	3	5	6	4
tictactoe	6	7	2	1	3	8	4	5
titanic	5	6	2	1	7	3	4	8
twonorm	7	1	3	2	4	8	5	6
vote	1	3	4	2	6	8	5	7
wine	2	6	3	5	4	8	7	1
yeast	2	3	7	6	4	8	5	1
zoo	2	8	7	3	5	6	4	1
Avg	3.66	5.21	4.53	3.39	5.05	5.95	5.00	3.21

Table A.13. Ranks of compared algorithms according to space complexity using MultiTest.

Data set	*c45*	*mdt*	*mlp*	*lnp*	*svl*	*sv2*	*svr*	*5nn*
australian	1	3	2	7	5	8	6	4
balance	6	1	2	3	4	8	5	7
breast	8	5	1	2	4	6	3	7
bupa	5	1	6	2	3	7	4	8
car	7	4	3	6	5	2	1	8
cmc	2	3	1	6	8	7	5	4
credit	1	2	3	4	6	8	7	5
cylinder	1	6	5	2	7	3	4	8
dermatology	1	3	2	4	5	6	7	8
ecoli	3	1	8	2	5	6	4	7
flags	1	7	2	3	4	6	5	8
flare	1	2	3	4	8	6	5	7
glass	6	1	7	2	4	5	3	8
haberman	1	2	4	3	7	5	6	8
heart	3	1	2	4	5	8	6	7
hepatitis	7	1	2	4	5	3	6	8
horse	7	1	6	2	5	8	4	3
ionosphere	1	4	5	6	7	2	3	8
iris	1	2	4	3	5	8	6	7
monks	1	5	3	4	6	7	2	8
mushroom	1	8	2	3	4	5	7	6
nursery	5	4	1	6	7	2	3	8
optdigits	8	7	4	6	3	2	1	5
pageblock	1	3	2	4	5	6	8	7
pendigits	8	6	4	5	7	1	2	3
pima	3	4	1	2	5	7	6	8
ringnorm	4	5	3	7	6	2	1	8
segment	1	3	8	2	5	4	6	7
spambase	7	4	1	2	5	8	3	6
tae	1	3	2	4	7	8	6	5
thyroid	1	2	3	4	7	6	5	8
tictactoe	7	6	1	2	3	8	4	5
titanic	4	5	1	2	6	7	3	8
twonorm	7	1	2	3	4	8	5	6
vote	1	2	3	4	5	8	6	7
wine	1	5	2	4	3	8	6	7
yeast	2	1	6	5	3	8	4	7
zoo	1	2	8	3	4	6	5	7
Avg	3.34	3.32	3.29	3.71	5.18	5.87	4.55	6.74

REFERENCES

1. Asuncion, A. and D. J. Newman, "UCI Machine Learning Repository", 2007, `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

2. Dietterich, T. G., "Approximate statistical tests for comparing supervised classification learning algorithms", *Neural Computation*, Vol. 10, No. 7, pp. 1895–1923, 1998.

3. Demsar, J., "Statistical Comparisons of Classifiers over Multiple Data Sets", *Journal of Machine Learning Research*, Vol. 7, pp. 1–30, 2006.

4. Alpaydın, E., "Combined 5×2 cv F test for comparing supervised classification learning algorithms", *Neural Computation*, Vol. 11, No. 8, pp. 1885–1892, 1999.

5. Bouckaert, R. R., "Estimating Replicability of Classifier Learning Experiments", *Proceedings of the International Conference on Machine Learning, ICML '04*, pp. 15–22, 2004.

6. Nadeau, C. and Y. Bengio, "Inference for the Generalization Error", *Machine Learning*, Vol. 52, pp. 239–281, 2003.

7. Yıldız, O. T. and E. Alpaydın, "Ordering and Finding the Best of $K > 2$ Supervised Learning Algorithms", *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 28, No. 3, pp. 392–402, 2006.

8. Turney, P. D., "Types of cost in inductive concept learning", *Proceedings of the Workshop on Cost-Sensitive Learning, ICML '00*, pp. 15–21, 2000.

9. Rasmussen, C. E., R. M. Neal, G. Hinton, D. van Camp, M. Revow, Z. Ghahramani, R. Kustra, and R. Tibshirani, "Delve Data for Evaluating Learning in Valid Experiments", 1995-1996, `http://www.cs.toronto.edu/~delve/`.

10. Yıldız, O. T. and E. Alpaydın, "Linear discriminant trees", *Proceedings of the International Conference on Machine Learning, ICML '00*, pp. 1175–1182, 2000.

11. Chang, C. C. and C. J. Lin, *LIBSVM: a library for support vector machines*, 2001, `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

12. Ulaş, A., O. T. Yıldız, and E. Alpaydın, "Cost-Conscious Comparison of Supervised Learning Algorithms over Multiple Data Sets", Technical Report FBE/CMPE-01/2008-04, Department of Computer Engineering, Boğaziçi University, İstanbul, 2008.

13. Yıldız, O. T., A. Ulaş, M. Semerci, and E. Alpaydın, "AYSU: Machine Learning Datasets for Model Combination, http://www.cmpe.boun.edu.tr/~ulas/aysu", 2007, `http://www.cmpe.boun.edu.tr/~ulas/aysu`.

14. Cohen, J., "The Earth is Round ($p < .05$)", *American Psychologist*, Vol. 49, pp. 997–1003, 1994.

15. Harlow, L. L., S. A. Mulaik, and J. H. Steiger (editors), *What If There Were No Significance Tests*, Lawrence Erlbaum Associates, 1997.

16. Wu, T.-F., C.-J. Lin, and R. C. Weng, "Probability Estimates for Multi-class Classification by Pairwise Coupling", *Journal of Machine Learninng Research*, Vol. 5, pp. 975–1005, 2004.