Table 9.20: Results for the diabetes dataset (2 classes, 8 attributes, 768 observations, 12-fold cross-validation).

Algorithm	Max. Storage	Time (sec.) Train	Time (sec.) Test	Error Rate Train	Error Rate Test	Rank
Discrim	338	27.4	6.5	0.220	0.225	3
Quadisc	327	24.4	6.6	0.237	0.262	11
Logdisc	311	30.8	6.6	0.219	0.223	1
SMART	780	3762.0	*	0.177	0.232	4
ALLOC80	152	1374.1	*	0.288	0.301	21
k-NN	226	1.0	2.0	0.000	0.324	22
CASTLE	82	35.3	4.7	0.260	0.258	10
CART	144	29.6	0.8	0.227	0.255	9
IndCART	596	215.6	209.4	0.079	0.271	14
NewID	87	9.6	10.2	0.000	0.289	19
AC^2	373	4377.0	241.0	0.000	0.276	18
Baytree	68	10.4	0.3	0.008	0.271	14
NaiveBay	431	25.0	7.2	0.239	0.262	11
CN2	190	38.4	2.8	0.010	0.289	19
C4.5	61	11.5	0.9	0.131	0.270	13
ITrule	60	31.2	1.5	0.223	0.245	6
Cal5	137	236.7	0.1	0.232	0.250	8
Kohonen	62	1966.4	2.5	0.134	0.273	17
DIPOL92	52	35.8	0.8	0.220	0.224	2
Backprop	147	7171.0	0.1	0.198	0.248	7
RBF	179	4.8	0.1	0.218	0.243	5
LVQ	69	139.5	1.2	0.101	0.272	16
Default	*	*	*	0.350	0.350	23

9.5.3 DNA

This classification problem is drawn from the field of molecular biology. Splice junctions are points on a DNA sequence at which "superfluous" DNA is removed during protein creation. The problem posed here is to recognise, given a sequence of DNA, the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA that are spliced out). The dataset used in the project is a processed version of the Irvine Primate splice-junction database. Each of the 3186 examples in the database consists of a window of 60 nucleotides, each represented by one of four symbolic values (*a*,*c*,*g*,*t*), and the classification of the middle point in the window as one of; intron–extron boundary, extron–intron boundary or neither of these. Processing involved the removal of a small number of ambiguous examples (4), conversion of the original 60 symbolic attributes to 180 or 240 binary attributes and the conversion of symbolic class labels to numeric labels (see Section 7.4.3). The training set of 2000 was chosen randomly from the dataset and the remaining 1186 examples were used as the test set.

This is basically a partitioning problem and so we might expect, in advance, that Decision Tree algorithms should do well. The classes in this problem have a heirarchical

Table 9.21: Results for the DNA dataset (3 classes, 60/180/240 attributes, (train, test) = (2000, 1186) observations).

Algorithm	Max. Storage	Time (sec.) Train	Time (sec.) Test	Error Rate Train	Error Rate Test	Rank
Discrim	215	928.5	31.1	0.034	0.059	4
Quadisc	262	1581.1	808.6	0.000	0.059	4
Logdisc	1661	5057.4	76.2	0.008	0.061	6
SMART	247	79676.0	16.0	0.034	0.115	17
ALLOC80	188	14393.5	*	0.063	0.057	3
k-NN	247	2427.5	882.0	0.000	0.146	20
CASTLE	86	396.7	225.0	0.061	0.072	8
CART	283	615.0	8.6	0.075	0.085	11
IndCART	729	523.0	515.8	0.040	0.073	9
NewID	729	698.4	1.0	0.000	0.100	15
AC^2	9385	12378.0	87.0	0.000	0.100	15
Baytree	727	81.7	10.5	0.001	0.095	13
NaiveBay	727	51.8	14.8	0.052	0.068	7
CN2	10732	869.0	74.0	0.002	0.095	13
C4.5	1280	9.0	2.0	0.040	0.076	10
ITrule	282	2211.6	5.9	0.131	0.135	19
Cal5	755	1616.0	7.5	0.104	0.131	18
Kohonen	2592	*	*	0.104	0.339	21
DIPOL92	518	213.4	10.1	0.007	0.048	2
Backprop	161	4094.0	9.0	0.014	0.088	12
RBF	1129	*	*	0.015	0.041	1
LVQ	FD	FD	FD	FD	FD	
Default	*	*	*	0.475	0.492	22

structure; the primary decision is whether the centre point in the window is a splice–junction or not. If it is a splice–junction then the secondary classification is as to its type; intron–extron or extron–intron.

Unfortunately comparisons between algorithms are more difficult than usual with this dataset as a number of methods were tested with a restricted number of attributes; some were tested with attribute values converted to 180 binary values, and some to 240 binary values. CASTLE and CART only used the middle 90 binary variables. NewID, CN2 and C4.5 used the original 60 categorical variables and k-NN, Kohonen, LVQ, Backprop and RBF used the one–of–four coding. The classical statistical algorithms perform reasonable well achieving roughly 6% error rate. k-NN is probably hampered by the large number of binary attributes, but Naive Bayes does rather well helped by the fact that the attributes are independent.

Surprisingly, machine learning algorithms do not outperform classical statistical algorithms on this problem. CASTLE and CART were at a disadvantage using a smaller window although performing reasonably. IndCART used 180 attributes and improved on the CART error rate by around 1%. ITrule and Cal5 are the poorest performers in this

group. ITrule, using only uni–variate and bi–variate tests, is too restricted and Cal5 is probably confused by the large number of attributes.

Of the neural network algorithms, Kohonen performs very poorly not helped by unequal class proportions in the dataset. DIPOL92 constructs an effective set of piecewise linear decision boundaries but overall, RBF is the most accurate algorithm using 720 centres. It is rather worrying here, that LVQ claimed an error rate of 0, and this result was unchanged when the test data had the classes permuted. No reason could be found for this phenomenon – presumably it was caused by the excessive number of attributes – but that the algorithm should "lie" with no explanation or warning is still a mystery. This problem did not occur with any other dataset.

In order to assess the importance of the window size in this problem, we can examine in a little more detail the performance of one of the machine learning algorithms. CN2 classified the training set using 113 rules involving tests on from 2 to 6 attributes and misclassifying 4 examples. Table 9.22 shows how frequently attributes in different ranges appeared in those 113 rules. From the table it appears that a window of size 20 contains the most important variables. Attributes just after the middle of the window are most important in determining class 1 and those just before the middle are most important in determining class 2. For class 3, variables close to the middle on either side are equally important. Overall though, variables throughout the 60 attribute window do seem to contribute. The question of how many attributes to use in the window is vitally important for procedures that include many parameters - Quadisc gets much better results (error rate of 3.6% on the test set) if it is restricted to the middle 20 categorical attributes.

Table 9.22: Frequency of occurrence of attributes in rules generated by CN2 for the DNA training set.

	1–10	11–20	21–30	31–40	41–50	51–60
class 1	17	10	12	59	7	2
class 2	17	28	78	21	13	11
class 3	6	8	57	55	4	3
total	40	46	147	135	24	16

It is therefore of interest to note that decision tree procedures get almost the same accuracies on the original categorical data and the processed binary data. NewID, obtained an error rate of 9.95% on the preprocessed data (180 variables) and 9.20% on the original data (with categorical attributes). These accuracies are probably within what could be called experimental error, so it seems that NewID does about as well on either form of the dataset. There is a little more to the story however, as the University of Wisconsin ran several algorithms on this dataset. In Table 9.23 we quote their results alongside ours for nearest neighbour. In this problem, ID3 and NewID are probably equivalent, and the slight discrepancies in error rates achieved by ID3 at Wisconsin (10.5%) compared to NewID (9.95%) in this study are attributable to the different random samples used. This cannot be the explanation for the differences between the two nearest neighbour results: there appears to be an irreconcilable difference, perhaps due to preprocessing, perhaps due to "distance" being measured in a conditional (class dependent) manner.

Certainly, the Kohonen algorithm used here encountered a problem when defining dis-

tances in the attribute space. When using the coding of 180 attributes, the Euclidean distances between pairs were not the same (the squared distances were 2.0 for pairs $(A,C),(A,G),(C,G)$ but only 1.0 for the pairs involving T: $(A,T),(C,T),(G,T)$). Therefore Kohonen needs the coding of 240 attributes. This coding was also adopted by other algorithms using distance measures (k-NN, LVQ).

Table 9.23: DNA dataset error rates for each of the three classes: splice–junction is Intron–Extron (IE), Extron–Intron (EI) or Neither. All trials except the last were carried out by the University of Wisconsin, sometimes with local implementations of published algorithms, using ten-fold cross-validation on 1000 examples randomly selected from the complete set of 3190. The last trial was conducted with a training set of 2000 examples and a test set of 1186 examples.

Algorithm	Neither	EI	IE	Overall
KBANN	4.62	7.56	8.47	6.28
Backprop	5.29	5.74	10.75	6.69
PEBLS	6.86	8.18	7.55	7.36
PERCEPTRON	3.99	16.32	17.41	10.31
ID3	8.84	10.58	13.99	10.50
Cobweb	11.80	15.04	9.46	12.08
N Neighbour (Wisconsin)	31.11	11.65	9.09	20.94
N Neighbour (Leeds)	0.50	25.74	36.79	14.60

9.5.4 Technical (Tech)

Table 9.24: The four most common classes in the technical data, classified by the value of attribute **X52**.

Range of **X52**	A_{69}	A_{72}	A_{77}	A_{78}
$<$ -0.085	0	1	0	180
-0.085, -0.055	260	0	0	0
-0.055, +0.055	0	324	0	0
+0.055, +0.085	0	0	1036	0
$>$ +0.085	0	0	0	392

Very little is known about this dataset as the nature of the problem domain is secret. It is of commercial interest to Daimler-Benz AG, Germany. The dataset shows indications of some sort of preprocessing, probably by some decision-tree type process, before it was received. To give only one instance, consider only the four most common classes ($A_{69}, A_{72}, A_{77}, A_{78}$), and consider only one attribute (**X52**). By simply tabulating the values of attribute **X52** it becomes obvious that the classifications are being made according to symmetrically placed boundaries on **X52**, specifically the two boundaries at -0.055 and +0.055, and also the boundaries at -0.085 and +0.085. These boundaries divide the range of **X52** into five regions, and if we look at the classes contained in these regions we get the frequency table in Table 9.24. The symmetric nature of the boundaries suggests strongly that the classes have been *defined* by their attributes, and that the class definitions are only concerned with *inequalities* on the attributes. Needless to say, such a system is perfectly suited to decision trees , and we may remark, in passing, that the above table was discovered