Using a similarity measure for credible classification

M. Anthony[*] P. L. Hammer[†] E. Subasi[‡] M. Subasi[§]

CDAM Research Report LSE-CDAM-2005-22

Abstract

This paper concerns classification by Boolean functions. We investigate the classification accuracy obtained by standard classification techniques on unseen points (elements of the domain, $\{0,1\}^n$, for some n) that are similar, in particular senses, to the points that have been observed as training observations. Explicitly, we use a new measure of how similar a point $x \in \{0,1\}^n$ is to a set of such points to restrict the domain of points on which we offer a classification. For points sufficiently dissimilar, no classification is given. We report on experimental results which indicate that the classification accuracies obtained on the resulting restricted domains are better than those obtained without restriction. These experiments involve a number of standard data-sets and classification techniques. We also compare the classification accuracies with those obtained by restricting the domain on which classification is given by using the Hamming distance.

[*] Department of Mathematics,London School of Economics, Houghton Street, London WC2A 2AE, UK, m.anthony@lse.ac.uk

[†] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA, hammer@rutcor.rutgers.edu

[‡] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA, esub@rutcor.rutgers.edu

[§] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA, msub@rutcor.rutgers.edu

1 Introduction

In [4], the authors proposed a way of measuring the similarity $s(x, A)$ of a Boolean vector x to a set A of such vectors. The measure is based on the *absence* of certain substrings of x from the set of vectors in A. In the context of machine learning classification problems, we may think of A as a training data-set, a set of observations on which we know the correct classifications. For example, each observation in the data set might arise from a set of medical tests on a patient and may represent, suitably encoded, the absence or presence—or degree of presence—of a number of symptoms the patient may have. In this context, the similarity measure provides a plausible way of deciding which unseen possible observations it would be credible to classify with some confidence once a classifier has been found that correctly classifies all (or most of) the observations in the training data-set.

Elegant and useful theories of classification error and confidence have been developed, but these usually make probabilistic assumptions about the way in which the observations have been generated. Specifically, the PAC model of learning and its variants (see, for instance [19, 21, 5, 2, 10]) assume that each observation in the data set has been chosen independently of the others, at random, according to a fixed probability distribution on $\{0, 1\}^n$, the set of all conceivable observations. Vovk *et al.* [23, 24, 20] have studied on-line learning in which one wants not only to predict classifications, but to give some indication of how 'credible' such predictions are, or not to predict if the predictions are not to be credible; and this is similar to the type of application we have in mind for the similarity measure. But in these papers, it is also assumed that the observations are generated independently according to the same probability distribution. In practice, what can one do without such probabilistic assumptions? It may be hard to prove anything sensible about classification accuracy in this case. Nonetheless, it might be at least useful not only to determine a classifier and to classify unseen observations with it, but also to attach to such predicted classifications the indication $s(x, A)$ of how similar the observation x is to those in the training data-set. Equally, one may decide not to classify at all those unseen observations that have a low similarity with the training data-set. This paper reports on empirical investigations that suggest that a higher classification accuracy is then achieved on the region of the domain $\{0, 1\}^n$ on which we *do* decide to classify.

2 A Measure of Similarity

2.1 Definitions

Suppose $x \in \{0,1\}^n$, $I \subseteq [n] = \{1, 2, \ldots, n\}$, and $|I| = k$. Then the projection of x onto I is the k-vector obtained from x by considering only the coordinates in I. For example, if $n = 5$, $I = \{2, 4\}$ and $x = 01001$ then $x|_I = 10$.

By a *positional substring* of $x \in \{0,1\}^n$, we mean a pair (z, I) where $z = x|_I$. The key point here is that the coordinates in I are specified: we will want, as part of our later definitions, to indicate that two vectors x and y have the same entries *in exactly the same places*, as specified by some $I \subseteq [n]$. For instance, although both $x = 10101$ and $y = 01010$ have substrings equal to 00, there is no I such that $x|_I = y|_I = 00$.

We now give the definition of similarity from [4].

Definition 2.1 *For $A \subseteq \{0,1\}^n$ and $x \in \{0,1\}^n$, the* similarity *of x to A, $s(x, A)$, is defined to be the largest s such that every positional substring (x, I) of length s appears also as a positional substring (y, I) of some observation $y \in A$. That is,*

$$s(x, A) = \max\{s : \forall I \subseteq [n], |I| \leq s, \exists y \in A, y|_I = x|_I\}.$$

Here $x|_I$ denotes the projection of x onto the coordinates indicated by I.

Equivalently, if r is the smallest length of a positional substring possessed by x that does *not* appear (in the same positions) anywhere in A, then $s(x, A) = r - 1$.

Notice that $s(x, A)$ is a measure of how similar x is to a *set* of vectors. It is not a metric or distance function. It can immediately be seen, indeed, that if A consists solely of one vector y, not equal to x, then $s(x, A) = 0$, since there must be some coordinate on which x and y differ (and hence a positional substring of length 1 of x that is absent from A).

Informally, the similarity of x to A is low if x has a short positional substring absent from A; and the similarity is high if all positional substrings of x of a fairly large length can be found in the same positions in some $y \in A$. To use the medical analogy discussed earlier, if x has a small combination of symptoms (that is, a simple syndrome) that does not appear in any of the patients in the set A then x has low similarity to A. Conversely, if $x \notin A$ then, certainly, it has some positional substring absent from A (as this is trivially true for the case $I = [n]$), but if the smallest such substring is long, then all simple syndromes indicated in x can be found among the patients of A. In this sense, x is similar to previously observed patients. One might expect that the presence or absence of a medical condition in a patient would be indicated by the patient having certain syndromes, and that short syndromes might carry more weight in such an explanation. For this reason, if a patient has a small syndrome not previously seen, one may want to be cautious in diagnosing the patient; whereas if all short syndromes possessed by the patient appear somewhere in the previously observed patients, one might have more confidence in a diagnosis on that patient.

This definition of similarity requires the elements of A to be binary vectors. However, in many applications, the raw data that we work with in a particular classification problem might be more naturally encoded as a real-valued vector. In such cases, the data may be transformed into binary data through a process known as *binarization* (see [6] for example). The transformed data set may then be simplified or cleaned in a variety of ways, by the removal of repeated points, for instance, and the deletion of coordinates found to be statistically insignificant in determining the classification.

2.2 A Boolean function formulation

Any Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be expressed by a *disjunctive normal formula* (or DNF), using *literals* $u_1, u_2, \ldots, u_n, \bar{u}_1, \ldots, \bar{u}_n$, where the $\bar{u}_i$ are known as *negated literals*. A disjunctive normal formula is one of the form

$$T_1 \vee T_2 \vee \cdots \vee T_k,$$

where each T_l is a *term* of the form

$$T_l = \left(\bigwedge_{i \in P} u_i \right) \bigwedge \left(\bigwedge_{j \in N} \bar{u}_j \right),$$

for some disjoint subsets P, N of $\{1, 2, \ldots, n\}$. A Boolean function is said to be a k-DNF if it has a disjunctive normal formula in which, for each term, the number of literals ($|P \cup N|$) is at most k. Such a function is said to be an l-term k-DNF if, additionally, it has a k-DNF formula in which the number of terms is at most l. For two Boolean functions f and g, we write $f \leq g$ if $f(x) \leq g(x)$ for all x; that is, if $f(x) = 1$ implies $g(x) = 1$. Similarly, for two Boolean formulae ϕ, ψ, we shall write $\phi \leq \psi$ if, when f and g are the functions represented by ϕ and ψ, then $f \leq g$. A term T of a DNF is said to *absorb* another term T' if $T' \leq T$. A term T is an *implicant* of f if $T \leq f$; in other words, if T true implies f true. The terms in any DNF representation of a function f are implicants of f. The most important type of implicants are the *prime implicants*. These are implicants with the additional property that there is no other implicant of f absorbing T. Thus, a term is a prime implicant of f if it is an implicant, and if the deletion of any literal from T results in a non-implicant T' of f (meaning that there is some x such that $T'(x) = 1$ but $f(x) = 0$). If we form the disjunction of all prime implicants of f, we have a DNF representation of f.

Given A, we can define $n+1$ Boolean functions $g_0, g_1, \ldots, g_n$, as follows. The function g_0 is taken to be the identically-0 function and, for $1 \leq k \leq n$, g_k is the 'largest' k-DNF function that is 0 on every member of A, in the sense that if f is a k-DNF function and $f(x) = 0$ for all $x \in A$ then $f \leq g_k$. It can be seen that g_k is the disjunction of all terms corresponding to positional substrings of length at most k that are not present in any element of A. For example, if the positional substring $(10, \{2, 4\})$ is not in A (that is, there is no $y \in A$ with $y_{\{2,4\}} = 10$) then, for $k \geq 2$, g_k will have as a term $u_2\bar{u}_4$.

Note that $s(x, A) \geq r$ if and only if $g_r(x) = 0$. For a subset B of $\{0, 1\}^n$ we denote by $\mathbb{I}_B$ the characteristic function of B, satisfying $\mathbb{I}_B(x) = 1 \iff x \in B$. Then, as noted in [4], if $\bar{A}$ denotes the complement $\{0, 1\}^n \setminus A$ of A, we have

$$0 \equiv g_0 \leq g_1 \leq g_2 \leq \cdots \leq g_{n-1} \leq g_n = \mathbb{I}_{\bar{A}}.$$

2.3 Computing similarity

One approach to computing the similarity is to compute the functions g_k and use the fact that, for a given x, $s(x, A) \geq k$ precisely if $g_k(x) = 0$. For fixed k, a k-DNF formula for g_k can be computed in time $O(|A|n^k)$ by using what is essentially Valiant's k-DNF learning algorithm [19, 3]. This proceeds as follows. Start with all terms of degree at most k and run through each observation in A in turn, deleting from the current set of terms those that are true on the current observation. Then, the disjunction of the remaining terms is g_k. Given any x, one can now determine whether $s(x, A) \geq k$ by establishing whether $g_k(x) = 0$. Of course, this algorithm is only efficient for (small) fixed k, not depending on n.

The problem of determining similarity can also be posed as a set covering problem. Note first that if we can determine the shortest positional substring possessed by x and absent from A, then $s(x, A)$ is one less than the length of this string. Now, fix $x \in \{0, 1\}^n$, and suppose $x \notin A$ (it being easy to check quickly whether $x \in A$). For $i = 1, 2, \ldots, n$, let $S_i = \{y \in A : y_i \neq x_i\}$. Then the smallest I such that for all $y \in A$, $y|_I \neq x|_I$ is exactly the smallest number of sets S_i needed to cover A. The standard greedy set-covering heuristic will therefore provide an efficient way of determining a number $s'(x, A)$ such that $s'(x, A) \leq s(x, A) \ln |A|$, enabling us at least to lower-bound the similarity.

2.4 Example

Example Suppose the set A consists of the following 10 points of $\{0, 1\}^5$.

$$\begin{array}{ccccc}
1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 0
\end{array}$$

Note, first, that no x can have $s(x, A) = 0$, since this could only happen if, on one of the five coordinates, all elements of A had a fixed value, either 0 or 1. Consider any x of the form $x = 01x_3x_4x_5$. Since there is no $y \in A$ with $y|_{\{1,2\}} = x|_{\{1,2\}} = 01$, we have $s(x, A) = 1$. Consider, however, $x = 10101$. For this x, we have $s(x, A) = 3$, because all (positional) substrings of x of length 3 belong to A, but there is no $y \in A$

such that $y|_{\{1,2,4,5\}} = x|_{\{1,2,4,5\}} = 1001$. Suppose now that $x = 00001$. Then, since all (positional) substrings of x of length 2 appear in A, $s(x, A) \geq 2$. However, there are substrings of length 3 missing from A: for example, there is no $y \in A$ with $y|_{\{1,3,4\}} = x|_{\{1,3,4\}} = 000$. So $s(x, A) = 2$.

3 Hierarchies based on similarity and relationship with Hamming distance

The similarity measure provides a way of filtering, or grading, $\{0, 1\}^n$ according to similarity to a given set A. For $0 \leq k \leq n$, let

$$A_k = \{x \in \{0, 1\}^n : s(x, A) \geq k\}$$

be the set of Boolean vectors which have similarity at least k to A. Then we have the following *hierarchy*:

$$\{0, 1\}^n = A_0 \supseteq A_1 \supseteq \cdots \supseteq A_{n-1} \supseteq A_n = A.$$

So, for large k, A_k is the set of vectors highly similar to A. Suppose that, in a machine learning problem, A is a training data-set. We might then decide to form a classifier of a particular type, using a particular learning algorithm, on the basis of A, but not to use it to predict classifications outside A_k for a particular choice of k. The rationale for this would be that vectors in $\{0, 1\}^n \setminus A_k$ are judged to be too dissimilar to those in A. In this paper we explore empirically whether this is a good strategy.

For a particular A, the hierarchy will typically look as follows:

$$\{0, 1\}^n = A_0 = \cdots = A_p \supset A_{p+1} \supseteq \cdots \supseteq A_e \supset A_{e+1} = \cdots = A_{n-1} = A_n = A,$$

where '$\supset$' denotes strict containment. (This is modified in the obvious way if $p = e$. Here, $p = p(A)$ is the 'pervasiveness' of A and $e = e(A)$ is the 'extent' of A, as defined in [4].) In terms of the Boolean functions g_k, we can see that A_k has characteristic function $\bar{g}_k$, the complement of g_k. The set A_k can also be thought of geometrically: if B_k is the union of all $(n - k)$-dimensional cubes that are contained entirely in the complement of A, then $A_k = \bar{B}_k$ is the complement of B_k. That is, A_k is obtained by deleting from $\{0, 1\}^n$ all cubes of co-dimension k that lie entirely outside A.

Another very natural way to measure how 'similar' a given $x \in \{0,1\}^n$ is to $A \subseteq \{0,1\}^n$ is to consider its Hamming distance. Recall that the Hamming distance $d(x, y)$ between x, y in $\{0,1\}^n$ is the number of entries on which they differ; and that, for $A \subseteq \{0,1\}^n$, the Hamming distance of x to the set A is defined by $d(x, A) = \min\{d(x, y) : y \in A\}$. This leads, in a similar way, to a hierarchy of subsets of $\{0,1\}^n$: if for $0 \leq k \leq n$, we let $D_k = \{x \in \{0,1\}^n : d(x, A) \leq n - k\}$, then we have the hierarchy

$$\{0,1\}^n = D_0 \supseteq D_1 \supseteq \cdots \supseteq D_{n-1} \supseteq D_n = A.$$

It can be shown [4] that, for all k, $A_k \subseteq D_k$. So, in this sense, the hierarchy resulting from the use of similarity is a refinement of that resulting from Hamming distance. However, the two approaches are quite different. For example, as shown in [4], if $A_k \neq \{0,1\}^n$, then $\{0,1\}^n \setminus A_k$ contains an element of $\{0,1\}^n$ that is at Hamming distance only 1 from A.

4 Classification accuracy and similarity

In this paper we explore, experimentally, the extent to which it appears that, on standard data-sets, standard learning algorithms produce more accurate classifications on unseen instances that have high similarity to those in a training set. We assume, therefore, that there is some underlying *target concept* $c : \{0,1\}^n \to \{0,1\}$ that represents the 'true' classifications of all $x \in \{0,1\}^n$. What we see when we learn is a subset $A \subseteq \{0,1\}^n$ together with the corresponding values of $c(y)$ for $y \in A$. On the basis of the training data-set and its classifications, we then produce a *hypothesis* $h : \{0,1\}^n \to \{0,1\}$ that we hope to be a good approximation to c. Typically, we might aim to produce, using one of a standard range of learning algorithms, a function h such that $h(y) = c(y)$ for all $y \in A$. Such a hypothesis is said to be *consistent* with the target concept on A (so that h is an *extension* of c). Ideally, we would hope that for many other points of $\{0,1\}^n$ (not in A), we would also have $h(x) = c(x)$. This has been thoroughly modelled and investigated within computational (or statistical) learning theory (see [19, 21, 5, 2, 10] for instance). However, as mentioned earlier, the theoretical results of computational learning theory require probabilistic assumptions about the way in which the data set is generated. Therefore, rather than require, as there, that *highly probable* instances be classified correctly, we might ask whether *highly similar* instances will be classified correctly by our hypothesis. That is, can we be sure that if the similarity of x to A is sufficiently high, then $h(x)$ will indeed be correct?

There is some theoretical evidence that such an approach might work. Veal [22] has shown that if there is a 'simple' underlying target concept, and if we use an algorithm that produces a simple classifier, then the classifications given to instances with sufficiently high similarity to the training data-set will be correct. More precisely, suppose the target concept, c, is an l-term k-DNF function and that the data-set is A. Suppose also that we have a hypothesis h which is an l'-term k'-DNF function and is consistent with the target concept on A. Then, for any $x \in \{0,1\}^n$, if $s(x, A) \geq \max\{l' + k, l + k'\}$, then $h(x) = c(x)$. Of course, we don't necessarily know *a priori* bounds on k and l, so this is not in practice necessarily very useful. However, it does show that if the similarity is sufficiently high, we will classify correctly. One might be tempted to think that, generally, an instance with a higher similarity to A is more likely to be correctly classified that one with a lower similarity. In the notation used above, this would mean that if $r > s$ then the proportion of points in A_r misclassified by h would be smaller than the proportion of points in A_s that are misclassified by h. We investigate experimentally, on standard data sets and using standard learning algorithms, whether this might be the case, and it does generally appear to be, at least for such standard data-sets. However, as shown in [22], it is possible to construct examples in which such a relationship does not hold: there is a target concept c and a training data-set A and hypothesis h such that h is consistent with c on A (that is, in an extension of c), but such that all the instances misclassified by h are of *higher* similarity that those correctly classified. It will not, therefore, be true in general that higher similarity necessarily implies higher classification accuracy, but this might, at least often, be the case for 'real', natural data-sets and target concepts.

5 Empirical results on classification accuracy for different data-sets

5.1 The data-sets

In our experiments we used the following nine real life data-sets, taken from the UCI Machine Learning Repository [18].

- Cleveland heart disease (hea)
- Pima Indian Diabetes (pid)
- German credit (nominal data from Statlog, made numeric and then binarized)
- Hepatitis
- Ionosphere
- Mushroom
- Tic-Tac-Toe
- House Votes (voting)
- Wisconsin breast cancer (bcw).

The data-sets were pre-processed in several ways before we ran our experiments. First, any observations in the data-set that had any missing attribute values were deleted. Next, the data-sets were binarized, according to the method described in [6], so that any numerical or nominal attribute values were changed to binary values. Next, techniques from [8, 9] were used to determine that some attributes (of the binarized data) could be deemed irrelevant and therefore deleted. (Set covering was used to find a small 'support set'.) The binarized data was then projected onto the remaining binary attributes. If this process resulted in any repetition, these were deleted, and if any of the processed observations appeared once with each class label, all its occurrences were deleted. After pre-processing in this manner, the data-sets consisted of binary vectors, generally in a higher-dimensional space than the original data. The following table describes the characteristics of the data-sets before and after this pre-processing.

Dataset	# observations		# attributes		After preprocessing		
	Positive	Negative	Numeric	Nominal	# observations		# binary attributes
					Positive	Negative	
Cleveland Heart Disease	139	164	10	3	137	158	63
Pima Indian Diabetes	130	262	8	0	130	262	47
German credit	700	300	7	13	697	300	66
Hepatitis	123	32	6	13	92	19	28
Ionosphere	225	126	34	0	216	125	49
Mushroom	3916	4208	0	22	2188	2047	50
Tic-Tac-Toe	626	332	0	9	626	332	27
Voting	267	168	16	0	96	64	16
Wisconsin Breast Cancer	458	241	9	0	203	182	48

5.2 Cross validation, error and accuracy

5.3 The learning algorithms

The classification methods, or learning algorithms, used in this experiment were taken from commonly used packages. These included See5 [17] and LAD [9] (see [12, 13] for background), the specific implementations used being Datascope [7] and Ladoscope [15]. We also used WEKA (see [25] and http://www.cs.waikato.ac.nz/ ml/weka/index.html), which consists of many algorithms. Those we used in our experiments are:

- J48, which generates a pruned or unpruned C4.5 decision tree. (See [17]).
- IBk K-nearest neighbours classifier. This normalizes the attributes by default and can select appropriate value of K based on cross-validation. For more information, see [1].

- Simple Logistic Regression Classifier for building linear logistic regression models. LogitBoost with simple regression functions as base learners is used for fitting the logistic models. The optimal number of LogitBoost iterations to perform is cross-validated, which leads to automatic attribute selection. For more information see [14].

- SMO implements John Platt's sequential minimal optimization algorithm for training a support vector classifier [16, 11, 25].

- Multilayer Perceptron, using back-propagation to train.

6 Accuracy on similarity hierarchy

The first set of experiments we conducted was intended to investigate whether the classification accuracy improved as we restricted the domain on which we predict, according to similarity.

To describe this in more detail, we must first explain cross-validation estimates. Suppose we randomly partition the data-set into two equally-sized parts, S and R. Suppose, further that we then use S as input to the learning (or classification) algorithm and measure the accuracy of the output hypothesis, h_S, of the algorithm on R, by which is meant the proportion of observations in R that are correctly classified by h_S. Then, suppose we instead use R as input to the learning algorithm and measure the accuracy of the output hypothesis, h_R, of the algorithm on S. If these two accuracy rates are then averaged, we obtain what is known as a 2-fold cross-validation estimate of accuracy for that partitioning of the data-set. If we repeat this procedure ten times, each time with a different randomly chosen partitioning of the data into two parts, then, for our purposes, we refer to the average accuracy of the ten cross-validation estimates as the 10-*times* 2-*fold CV (cross-validation) estimate* of the accuracy. We shall sometimes find it more convenient to consider *error* rather than *accuracy.* Error measures the proportion of observations incorrectly classified, and so it is just 1 minus the accuracy.

Now, we are interested in the performance of a classifier on observations that have at least a given similarity to the observations that were used as input to the learning algorithm that produced the classifier (or hypothesis). Suppose that k is some positive integer. We might then adapt the cross-validation procedure outlined above as follows: instead of finding the accuracy of h_S on R and then of h_R on S, and averaging the two, we instead determine the accuracies of h_S on $R \cap S_k$ and of h_R on $S \cap R_k$, and average the two. Recall that S_k is the set of points in the data-set that have similarity at least k to S (and R_k is similarly defined). Repeating this ten times and averaging, we obtain an estimate which we call the 10-*times* 2-*fold CV estimate on observations of similarity at least* k.

For values of k between 2 and 6, and for each of the nine data-sets and each of the seven learning algorithms, we determined the 10-times 2-fold CV estimate on observations of similarity at least k. It is conceivable that any perceived improvement in the accuracy estimates as we increase the similarity might be an artefact of the use of a particular learning algorithm, so we report two types of result here. First, for each data-set, we report the average, over all seven learning algorithms, of the accuracy estimates. Secondly, we report, for each algorithm, the average of the accuracy estimates over all nine data-sets.

6.1 Performance on each data-set

Figure 1 illustrates the accuracies obtained on restricting the domain of prediction to observations of increasing similarity for the Cleveland Heart Disease data, and Figure 2 does likewise for the German Credit data. These accuracies are the average accuracies over all seven learning algorithms. The detailed results for all the data-sets are indicated in the Tables in Section A1 of the appendix (which also indicate the average number of observations having at least a given similarity). Figure 3 shows the average, over all the data-sets, of the average accuracies over all seven algorithms.

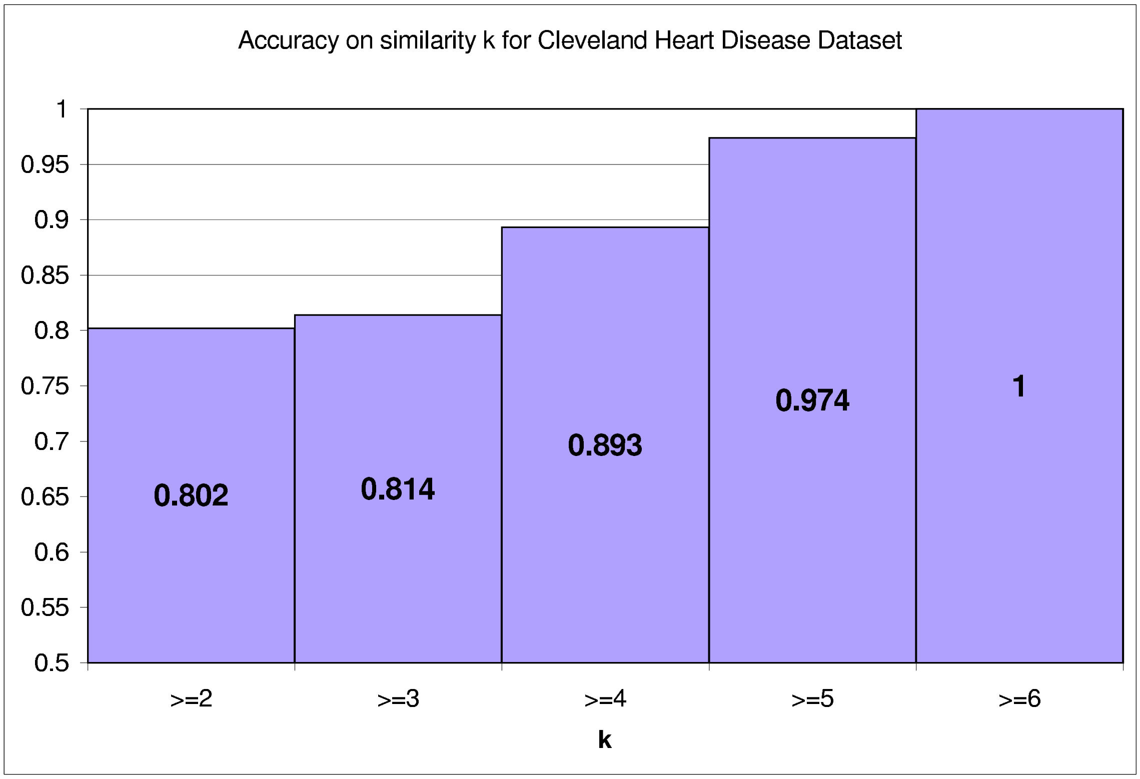

Figure 1: The average, over all seven learning algorithms, of the 10-times 2-fold CV estimates on observations of similarity at least $k = 2, 3, 4, 5, 6$ for the Cleveland Heart Disease Data

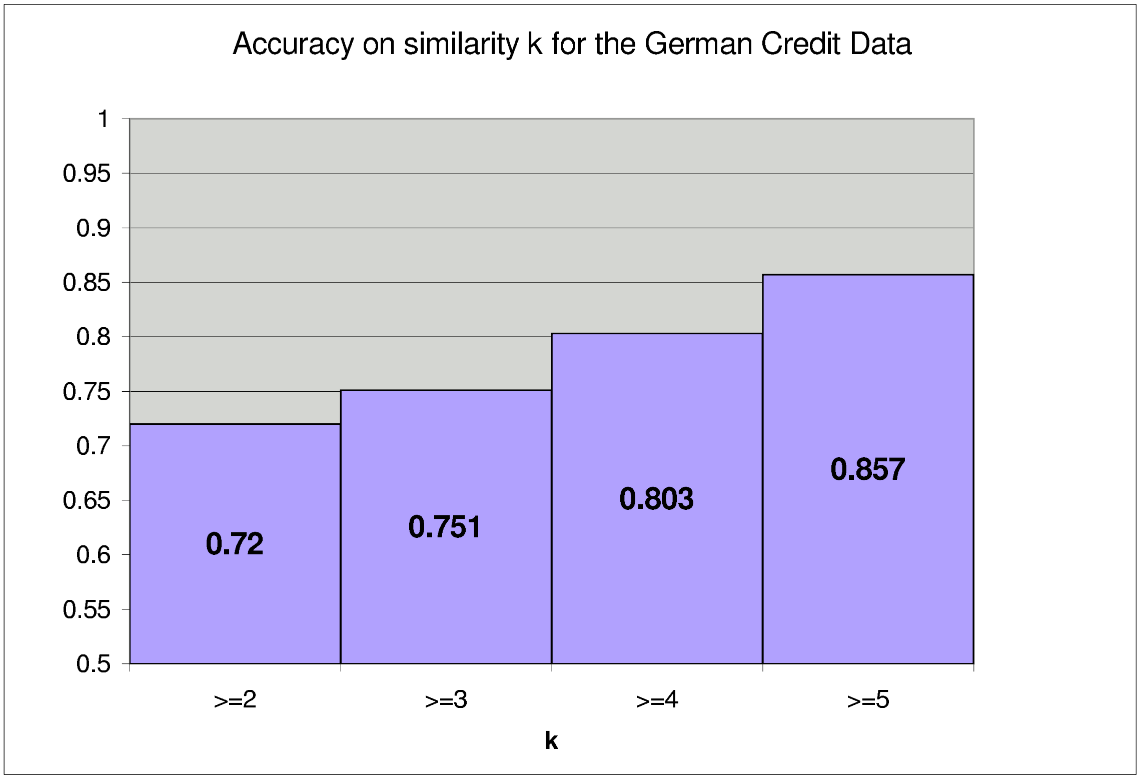

Figure 2: The average, over all seven learning algorithms, of the 10-times 2-fold CV estimates on observations of similarity at least $k = 2, 3, 4, 5$ for the German Credit Data

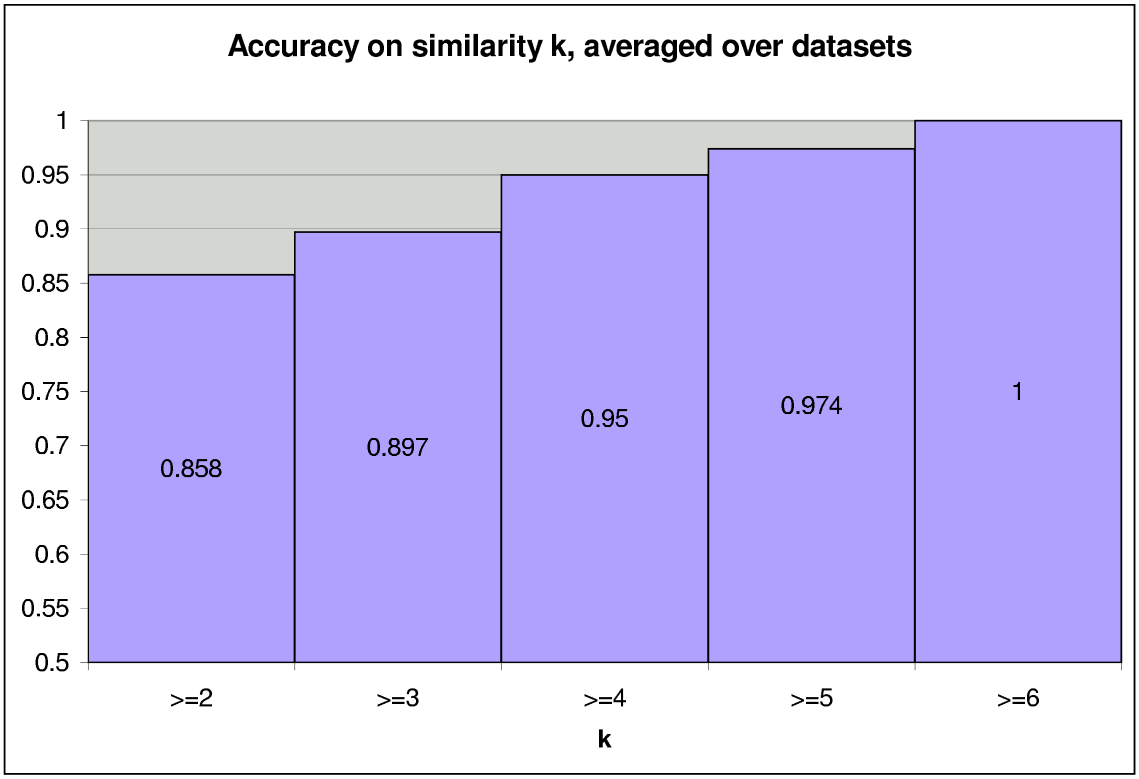

Figure 3: The average, over all nine data-sets of the average, over all seven learning algorithms, of the 10-times 2-fold CV estimates on observations of similarity at least $k = 2, 3, 4, 5$.

6.2 Performance of each learning algorithm

Figure 4 illustrates the accuracies obtained on restricting the domain of prediction to observations of increasing similarity when the LAD learning algorithm is used, and Figure 5 does likewise for the SEE5 algorithm. These accuracies are the average accuracies over all nine data-sets. The results for all the learning algorithms are indicated in the Tables in Section A2 of the appendix (which also indicate the average number of observations having at least a given similarity). Figure 6 shows the average, over all seven algorithms, of the average accuracies over all nine data-sets.

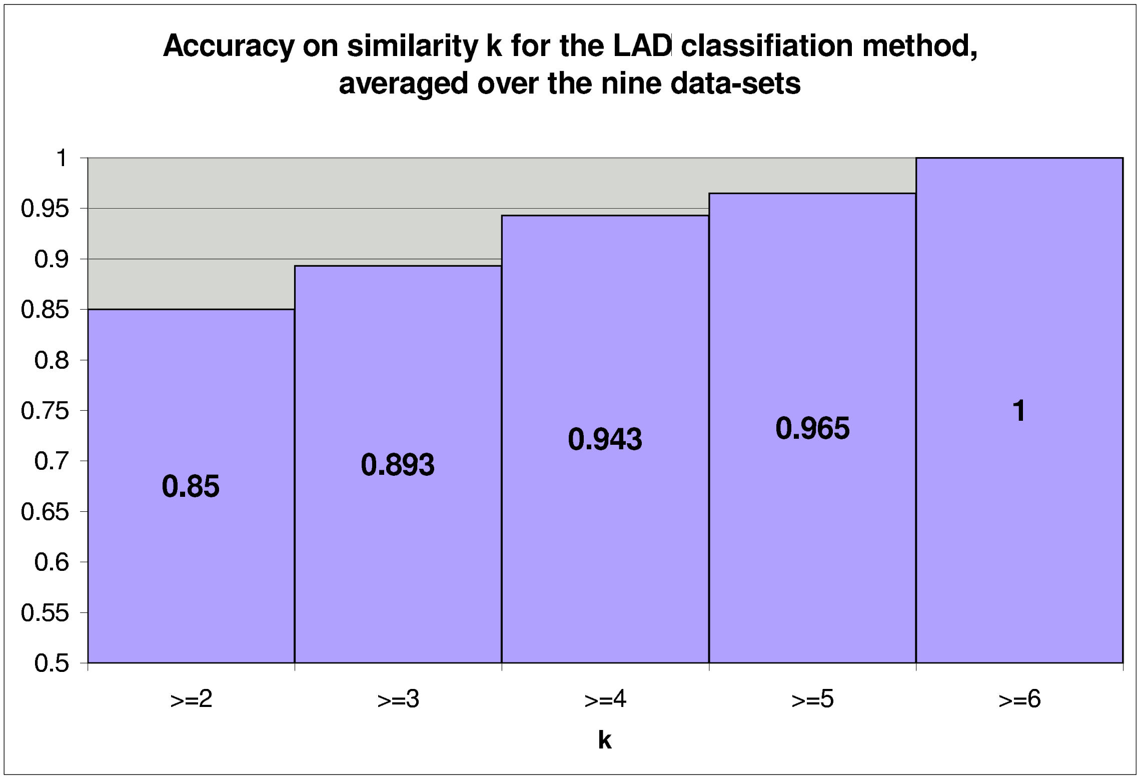

Figure 4: The average, over all nine data sets, of the 10-times 2-fold CV estimates on observations of similarity $k = 2, 3, 4, 5, 6$ using the LAD algorithm

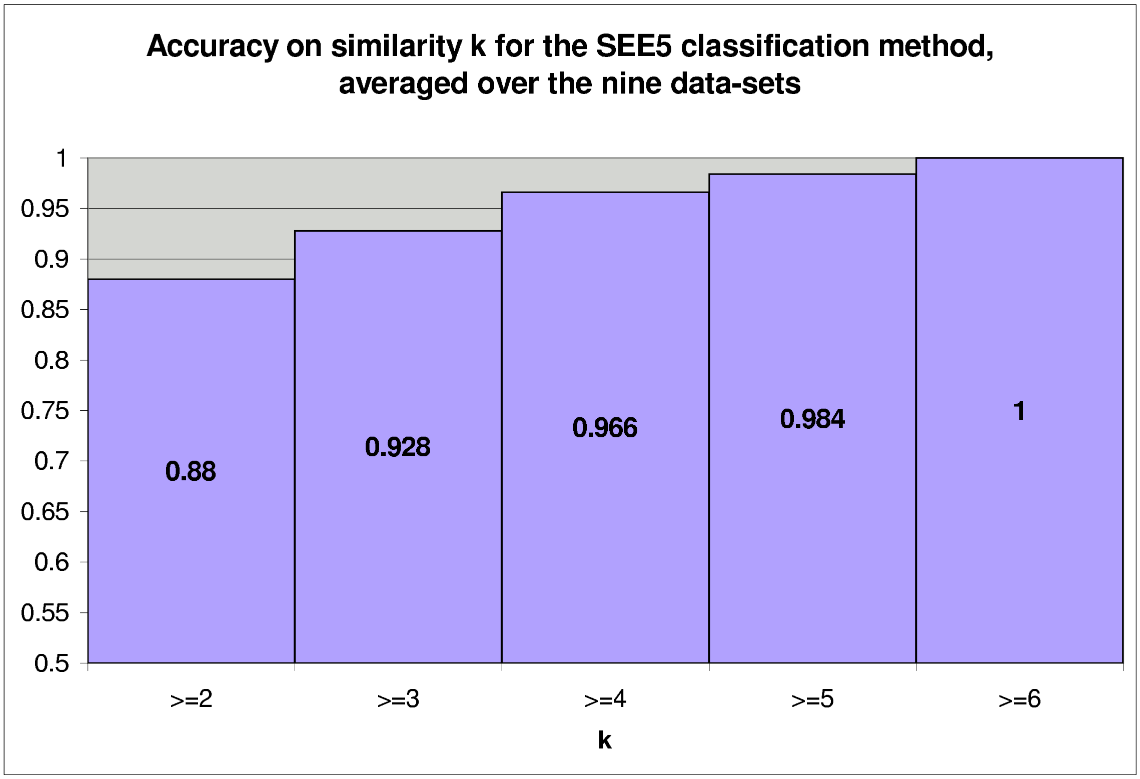

Figure 5: The average, over all nine data sets, of the 10-times 2-fold CV estimates on observations of similarity $k = 2, 3, 4, 5, 6$ using the SEE5 algorithm

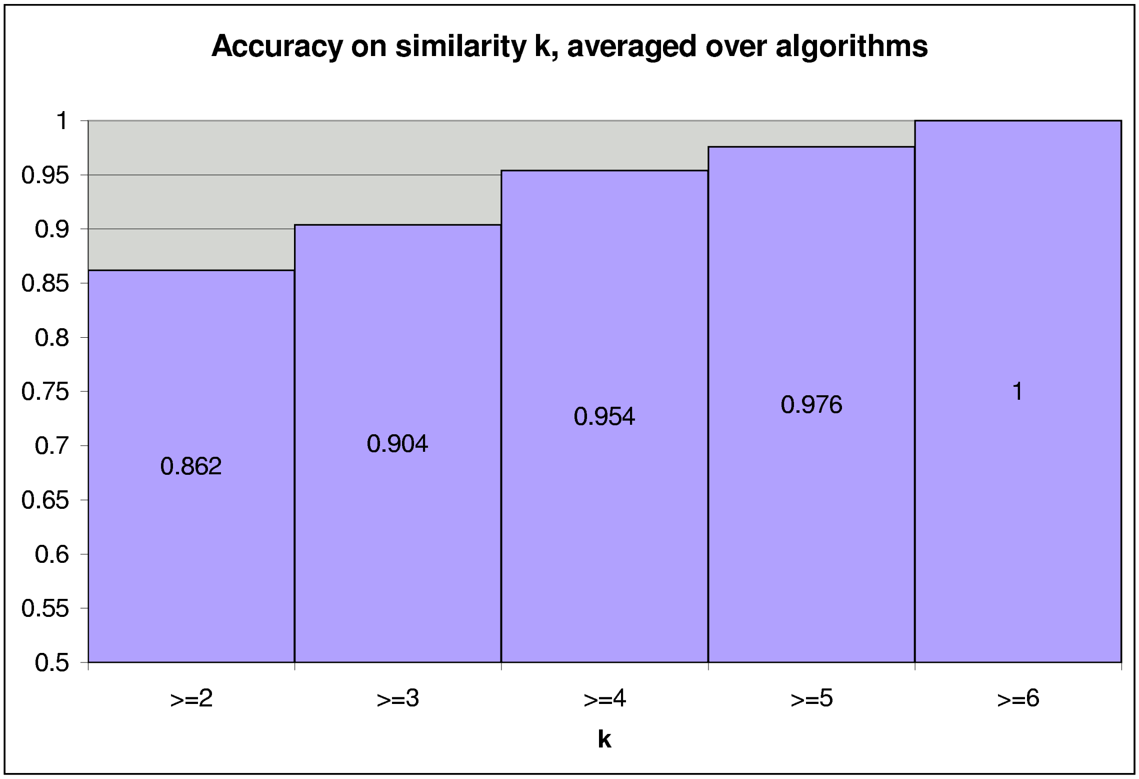

Figure 6: The average, over all seven algorithms, of the average over the nine datasets, of the 10-times 2-fold CV estimates on observations of similarity at least $k = 2, 3, 4, 5$.

7 Accuracy on Hamming distance hierarchy

The next set of experiments investigates whether the same type of increased accuracy estimates result when the domain of prediction is determined by Hamming distance rather than similarity. We use the same cross-validation partitions as for the previous experiments. The Hamming-distance estimates we use are defined in a a similar way to the CV estimates on observations of similarity at least k. For a range of values of d, we proceed exactly as described in Section 6, but instead of using the accuracies of h_S on $R \cap S_k$ and of h_R on $S \cap R_k$, we instead find the accuracies of h_S on $\{x \in R : d(x, S) \leq d\}$ and of h_R on $\{x \in S : d(x, R) \leq d\}$. We call the resulting version of the 10-times 2-fold CV estimate the *10-times 2-fold CV estimate on observations of Hamming distance at most k*. Again, as for similarity, we report two types of result. First, for each data-set, we report the average, over all seven learning algorithms, of the accuracy estimates. Secondly, we report, for each algorithm, the average of the accuracy estimates over all nine data-sets.

7.1 Performance on each data-set

Figure 7 illustrates the accuracies obtained on restricting the domain of prediction to observations of decreasing Hamming distance for the Cleveland Heart Disease data. Figure 8 does likewise for the German Credit data. These accuracies are the average accuracies over all seven learning algorithms. Results for all the data sets (together with information about the number of observations with a given Hamming distance) can be found in the table in Section A3 of the appendix.

7.2 Performance of each learning algorithm

Figure 9 illustrates the accuracies obtained on restricting the domain of prediction to observations of given Hamming distance when the LAD learning algorithm is used, and Figure 10 does likewise for the SEE5 algorithm. These accuracies are the average accuracies over all nine data-sets. Results for all the algorithms can be found in the table in Section A4 of the appendix.

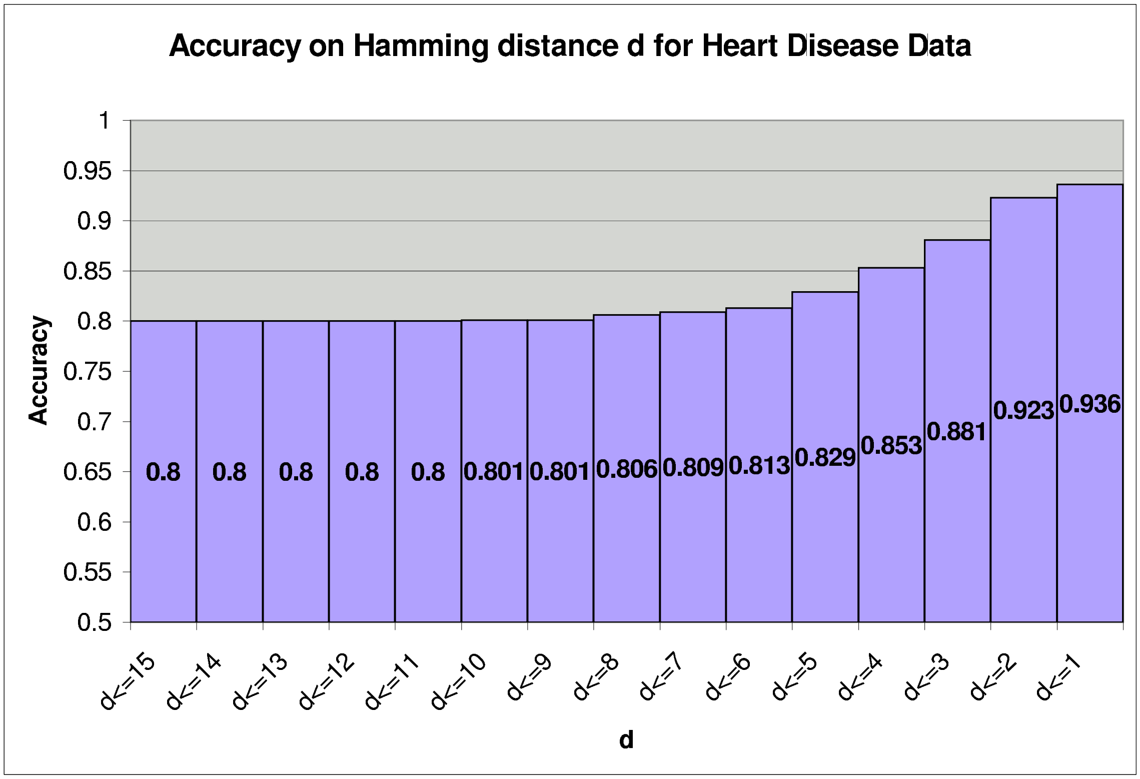

Figure 7: The average, over all seven learning algorithms, of the 10-times 2-fold CV estimates on observations at Hamming distance at most d for the Cleveland Heart Disease data.

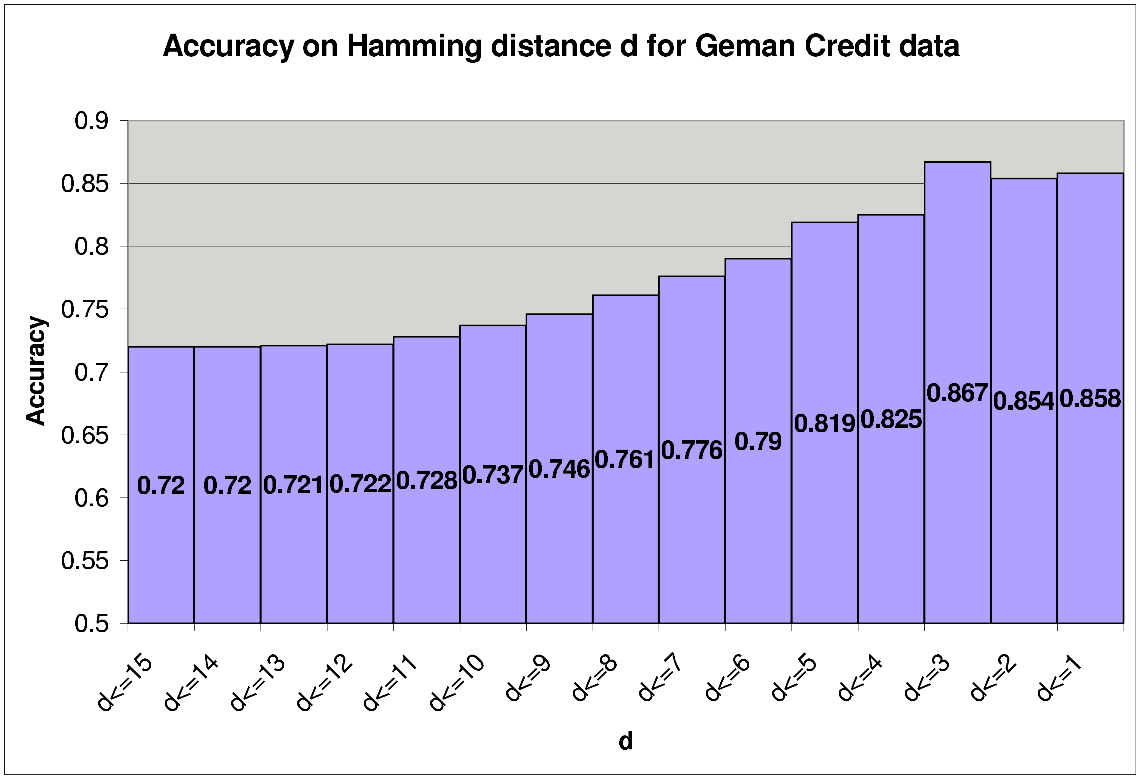

Figure 8: The average, over all seven learning algorithms, of the 10-times 2-fold CV estimates on observations at Hamming distance at most d for the German Credit data.

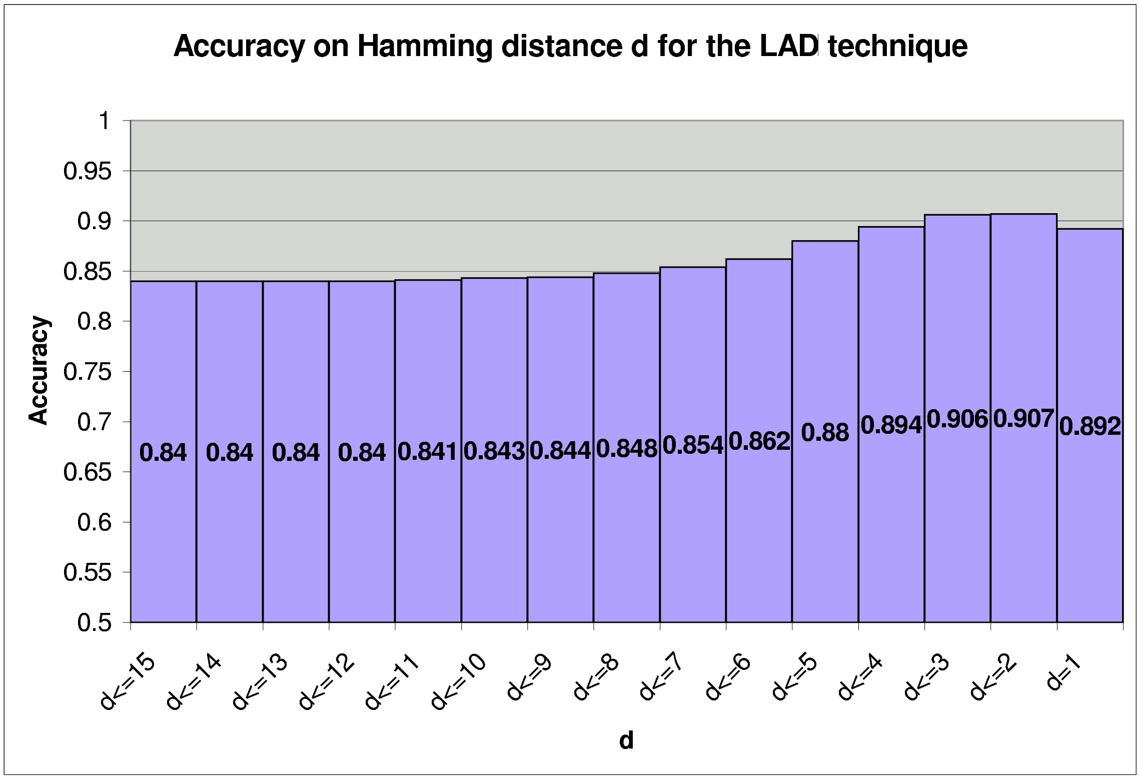

Figure 9: The average, over all nine data-sets, of the 10-times 2-fold CV estimates on observations at Hamming distance at most d for the LAD learning algorithm.

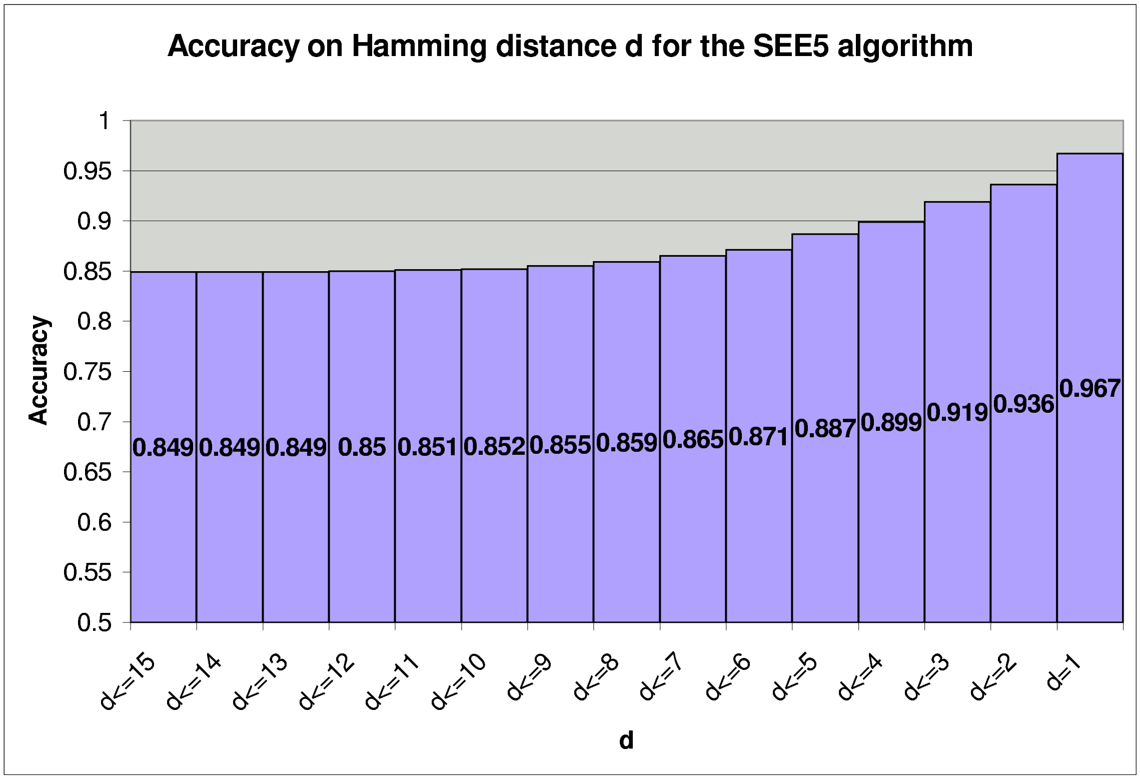

Figure 10: The average, over all nine data-sets, of the 10-times 2-fold CV estimates on observations at Hamming distance at most d for the SEE5 learning algorithm.

8 Comparing error rates on similarity and Hamming hierarchies

The experimental results appear to suggest that higher accuracies are obtained when we restrict prediction to observations of high similarity to those used as input to the learning algorithm, and that we also obtain higher accuracies when we restrict prediction to observations that have small Hamming distance to the set of observations used to produce the classifier. To compare the effects of both type of restriction, the tables in Sections A5.1 of the Appendix show, for each data-set, for each relevant value of k and d, the ratio of the *errors* under each type of restriction, averaged over each learning algorithm. Explicitly, the entry in the row labelled 'HD<= d' and column labelled k is the ratio of the 10-times 2-fold CV error estimates (which are 1 minus the accuracy estimates) on observations of similarity at least k to the 10-times 2-fold CV error estimates on observations of Hamming distance at most d. (We compare error rates rather than accuracy rates because when both accuracy rates are close to 1, as they usually are, the ratio of accuracies will also be very close to 1. For this reason, a comparison of error rates is more revealing.) The cells in these tables that are highlighted in grey are where these ratios are greater than 1 (indicating that the error restricted to the corresponding similarity is greater than that when restricted to the given Hamming distance). The accuracies corresponding to each column and to each row are also indicated. The tables in Section A5.2 indicate the corresponding ratios when the errors are averaged, for each learning algorithm, over all data-sets.

9 Using similarity and Hamming distance together

An observation that has *both* high similarity and low Hamming distance to a given set A is, arguably, strongly 'like' the members of A. We have seen that classification accuracy appears to improve when we, separately, restrict prediction to observations of high similarity to, or small Hamming distance from, those used to produce the classifier. In this section, we report experimental results examining the accuracy when prediction is restricted simultaneously by similarity and Hamming distance. Explicitly, for each d between 1 and 16, and each k between 2 and 6, we proceed exactly as described in Section 6, but using the accuracies of h_S on $\{x \in R :$

$d(x, S) \leq d,\ s(x, S) \geq k\}$ and of h_R on $\{x \in S : d(x, R) \leq d,\ s(x, R) \geq k\}$.

9.1 Performance on each data-set

Figure 11 and Figure 12 illustrate, respectively, the average accuracies on Hamming distance at most d and similarity at least k for the Cleveland Heart Disease and German Credit data-sets.

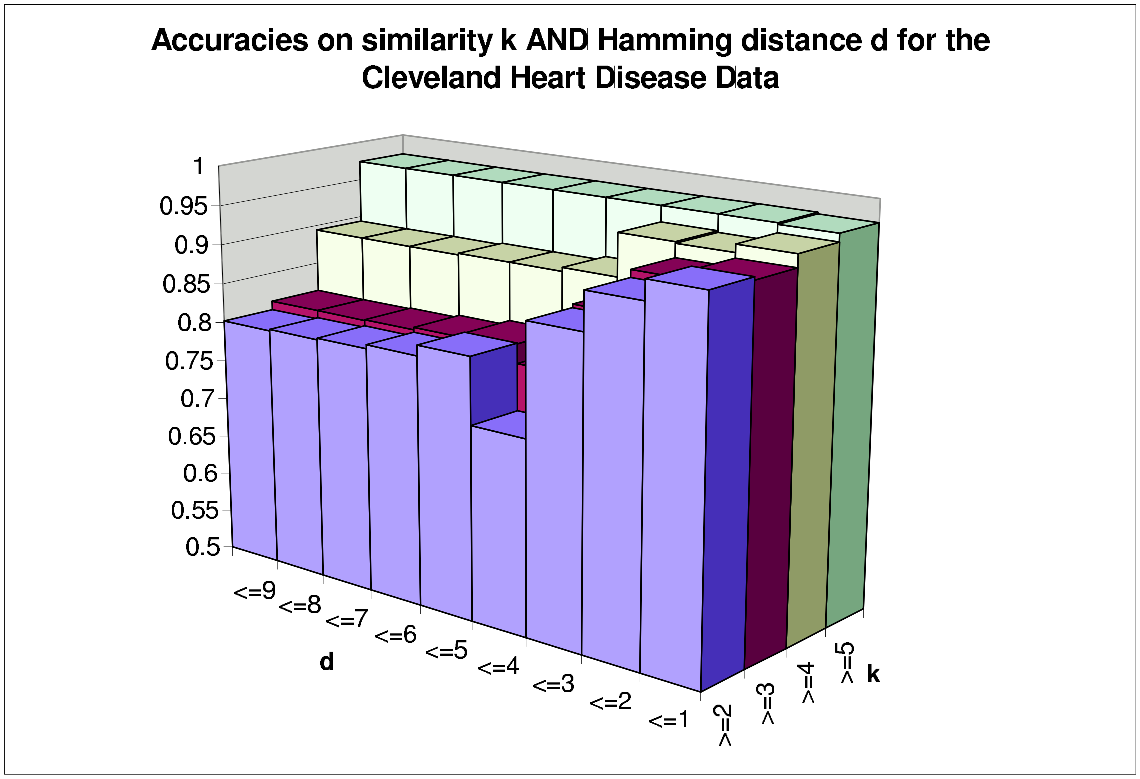

Figure 11: The average, over all seven learning algorithms, of the 10-times 2-fold CV estimates on observations of given similarity and Hamming distance for the Cleveland Heart Disease Data

Figure 13 shows the average, over all nine data sets, of the average, over all seven learning algorithms, of the accuracies on Hamming distance at most d and similarity at least k.

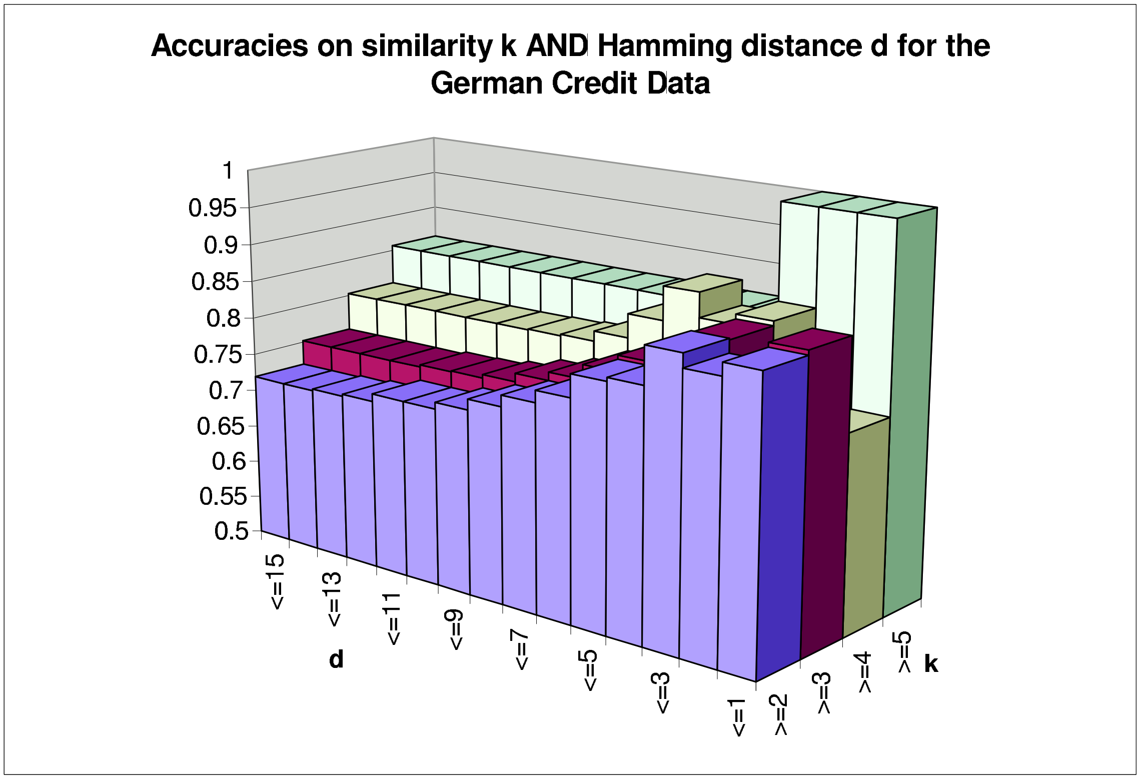

Figure 12: The average, over all seven learning algorithms, of the 10-times 2-fold CV estimates on observations of given similarity and Hamming distance for the German Credit Data

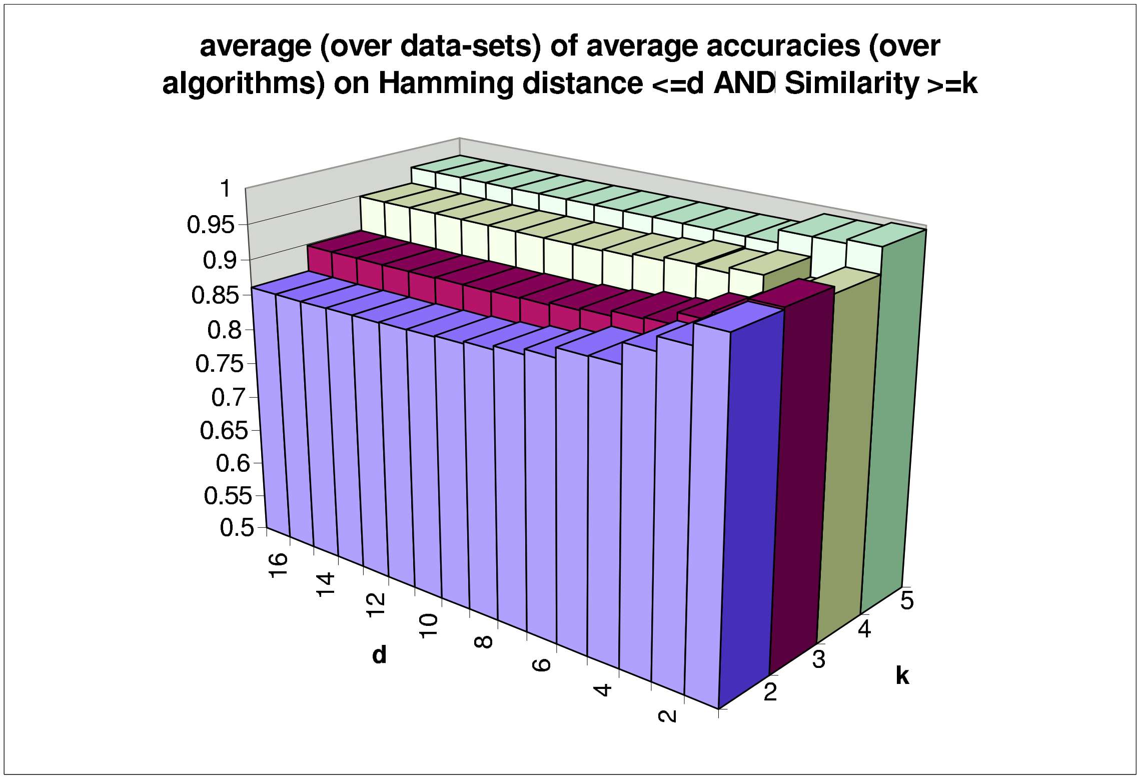

Figure 13: The average, over all nine data sets, of the average, over all seven learning algorithms, of the 10-times 2-fold CV estimates on observations of given similarity and Hamming distance.

Further data can be found in the Tables in Section A6.1 of the Appendix, where numbers of observations of at most a given Hamming distance and at least a given similarity are also indicated.

9.2 Performance of each learning algorithm

Figure 14 and Figure 15 illustrate, respectively, the average accuracies, over all data-sets, on Hamming distance at most d and similarity at least k when using the LAD and SEE5 classification techniques.

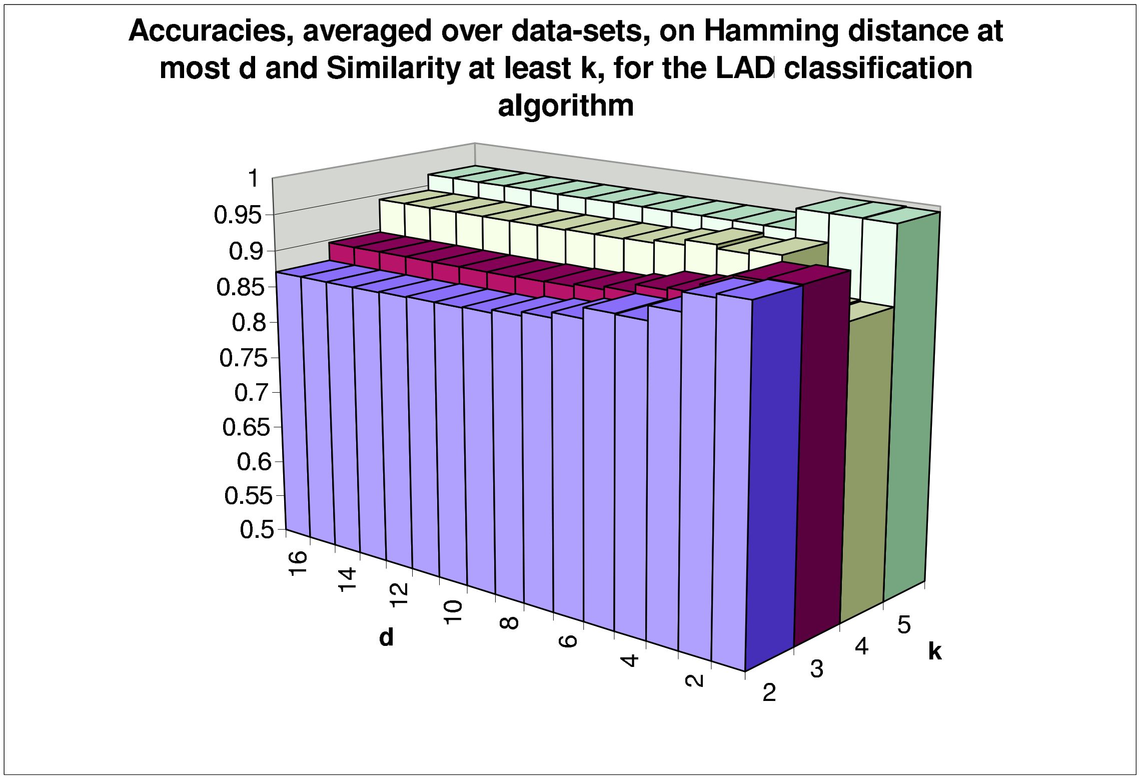

Figure 14: The average, over all nine data-sets, of the 10-times 2-fold CV estimates on observations of given similarity and Hamming distance when using the LAD classification technique.

Figure 16 shows the average, over all seven learning algorithms, of the average, over

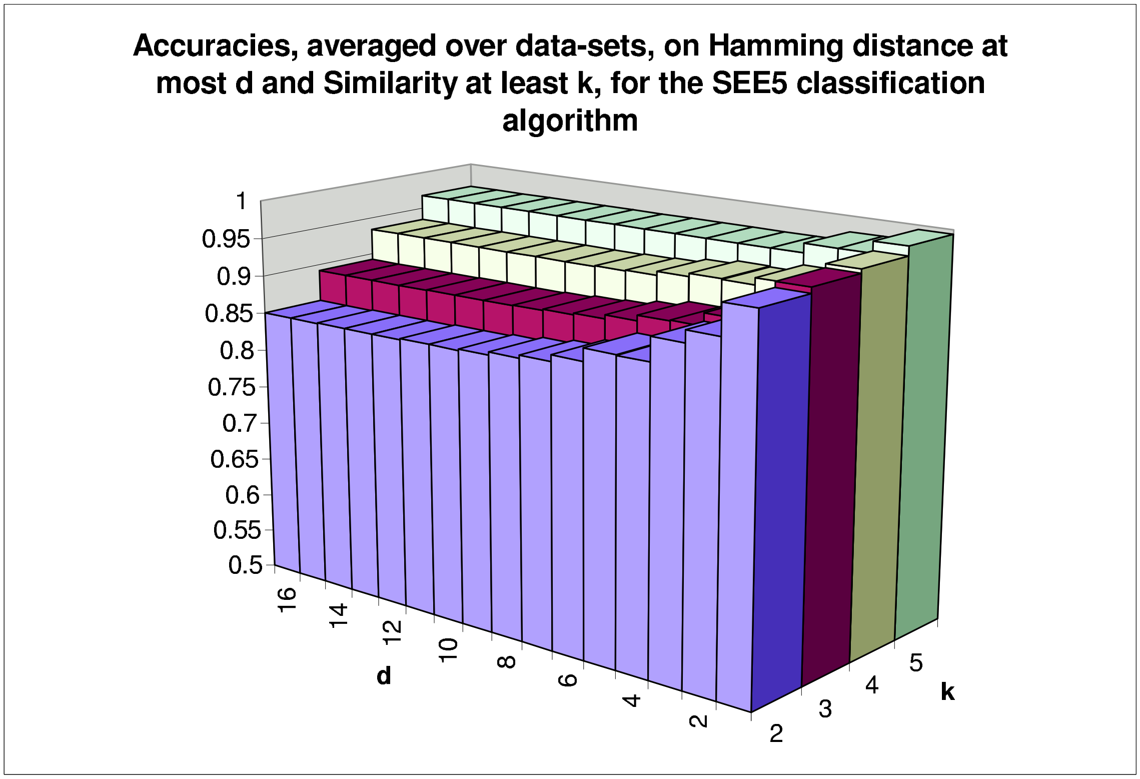

Figure 15: The average, over all nine data-sets, of the 10-times 2-fold CV estimates on observations of given similarity and Hamming distance when using the SEE5 classification technique.

all nine data-sets, of the accuracies on Hamming distance at most d and similarity at least k.

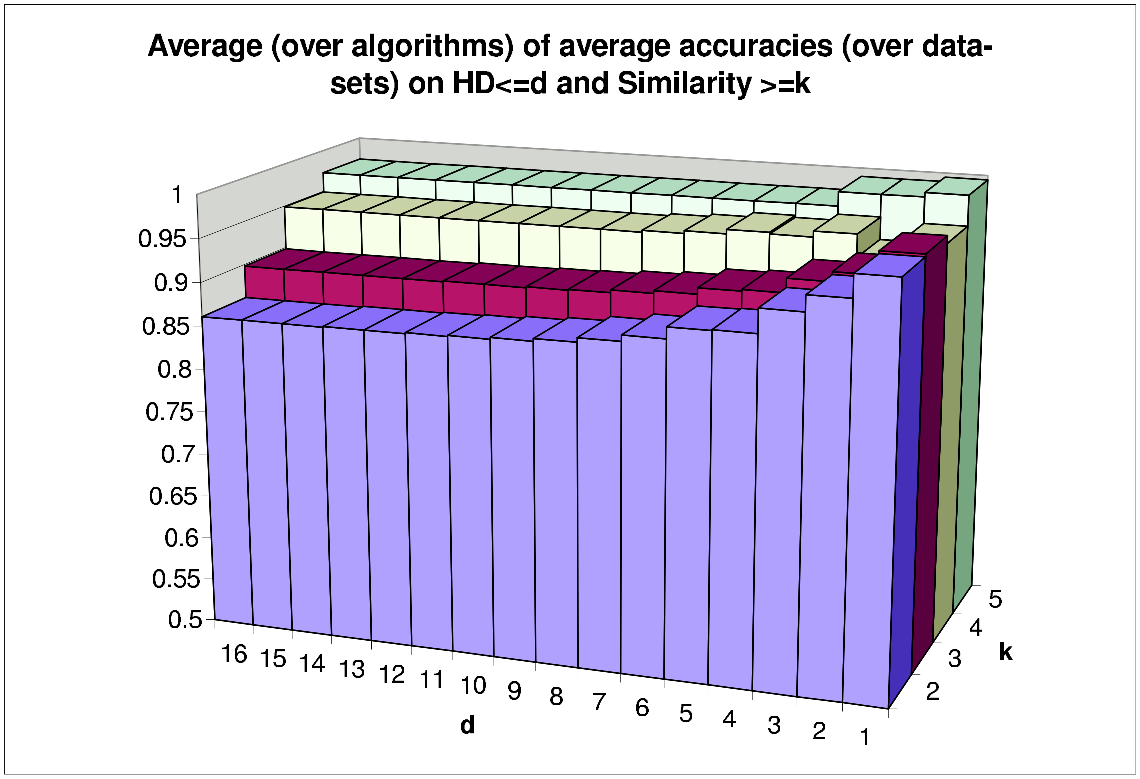

Figure 16: The average, over all seven learning algorithms, of the average, over all nine data-sets, of the 10-times 2-fold CV estimates on observations of given similarity and Hamming distance.

Further data can be found in the Tables in Section A6.2 of the Appendix.

10 Conclusions

The experimental results here indicate that there is some advantage in using 'similarity' and Hamming distance, separately and in combination, to restrict the observations on which one is willing to offer a confident prediction. As noted, there are provably cases in which this is not so, but the principle does appear generally to be borne out by the data-sets and algorithms used here.

11 Acknowledgements

Martin Anthony's work is supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. Thanks to Iain Morrow for advice on graphs. Peter Hammer's work is partially supported by NSF Grant NSF-IIS-0312953 and NIH Grants NIH-002748-001 and NIH-HL-072771-01.

References

[1] D. Aha and D. Kibler. Instance-based learning algorithms, *Machine Learning*, vol.6, 1991: 37-66.

[2] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, Cambridge, UK, 1999.

[3] Martin Anthony and Norman Biggs. *Computational Learning Theory: An Introduction*. Cambridge University Press, Cambridge, UK, 1992.

[4] Martin Anthony and Peter L. Hammer. *A Boolean measure of similarity.* RUTCOR Research Report RRR-27-2004, RUTCOR, Rutgers Center for Operations Research, Rutgers University, New Jersey, August 2004. In press, *Discrete Applied Mathematics.*

[5] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler and Manfred Warmuth, Learnability and the Vapnik-Chervonenkis dimension, *Journal of the ACM*, 36(4), 1989: 929–965.

[6] Endre Boros, Peter L. Hammer, Toshihide Ibaraki and Alexander Kogan. Logical analysis of numerical data. *Mathematical Programming* 79, 1997: 163–190.

[7] http://rutcor.rutgers.edu/ salexe/LAD_kit/SETUP-LAD-DS-SE20.zip

[8] Endre Boros, Peter L. Hammer, Toshihide Ibaraki, Alexander Kogan, Eddy Mayoraz and Ilya Muchnik. *An Implementation of Logical Analysis of Data.* RUTCOR Research Report RRR-22-96, RUTCOR, Rutgers Center for Operations Research, Rutgers University, New Jersey, 1996.

[9] Endre Boros, Peter L. Hammer, Toshihide Ibaraki, Alexander Kogan, Eddy Mayoraz and Ilya Muchnik. An Implementation of LOgical Analysis of Data. *IEEE Trans. on Knowledge and Data Engineering* 12-1 (2000): 292–306

[10] Michael J. Kearns and Umesh Vazirani, *Introduction to Computational Learning Theory*, MIT Press, Cambridge, MA, 1995.

[11] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya and K.R.K. Murthy. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation*, 13(3), 2001: 637-649.

[12] Peter L. Hammer. *Partially Defined Boolean Functions and Cause-Effect Relationships.* Presented at the International Conference on Multi-Attribute Decision Making Via OR-Based Expert Systems, University of Passau, Passau, Germanym April 1986.

[13] Yves Crama, Peter L. Hammer and Toshihide Ibaraki. Cause-effect relationships and partially-defined Boolean functions. *Annals of Operations Research* 16, 1988: 299–326.

[14] N.Landwehr, M.Hall and E. Frank. Logistic Model Trees. ECML 2003.

[15] Pierre Lemaire, *Ladoscope - a set of tools for the logical analysis of data.* `http://rutcor.rutgers.edu/ lemaire/LAD/`

[16] John Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization, in *Advances in Kernel Methods - Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smola, eds., MIT Press, 1998.

[17] Ross Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[18] University of California at Irvince Machine Learning Repository. `http://www.ics.uci.edu/ mlearn/MLRepository.html`

[19] Leslie G. Valiant, A theory of the learnable. *Communications of the ACM*, 27 (11), 1984: 1134–1142.

[20] C. Saunders, A. Gammerman and V. Vovk. Transduction with confidence and credibility. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1999: 722–726.

[21] Vladimir N. Vapnik: *Statistical Learning Theory*, Wiley, 1998.

[22] Bean Veal. *Properties of a Binary Similarity Measure*, CDAM Research Report LSE-CDAM-2005-06, Centre for Discrete and Applicable Mathematics, London School of Economics.

[23] V. Vovk. On-line confidence machines are well-calibrated. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*, 2002, Los Alamitos, CA, IEEE Computer Society: 187–196.

[24] V. Vovk. Asymptotic optimality of Transductive Confidence Machine. In *Proceedings of the 13th International Conference on Algorithmic Learning Theory* (ed by N Cesa-Bianchi, M Numao and R Reischuk), Lecture Notes in Artificial Intelligence, vol 2533, 2002: 336–350.

[25] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000.

APPENDIX

A1. Average cross-validation accuracy on observations of at least a given similarity, for each of the data-sets, averaged over all learning algorithms.

For values of k between 2 and 6, and for each of the nine data-sets the following tables show the following: (1) in the small boxes, the average (over the 10 cross-validations) of the numbers of non-training observations having at least a given similarity to the training set, (2) the average, over all seven learning algorithms, of the 10-times 2-fold cross-validation estimate on observations of similarity at least k, and (3) (labeled 'ALL') the average accuracy, over all seven learning algorithms, on all non-training observations.

CLEVELAND HEART DISEASE DATASET

k	6	5	4	3	2
# of observations with similarity at least k, and accuracy on these	1 1	3 0.974	14 0.893	55 0.814	115 0.802
ALL	148 0.801				

DIABETES DATASET

k	6	5	4	3	2
# of observations with similarity at least k, and accuracy on these	1 1	1 1	2 0.987	170 0.790	190 0.746
ALL	196 0.746				

GERMAN CREDIT DATASET (nominal)

k	6	5	4	3	2
# of observations with similarity at least k, and accuracy on these		1 0.857	41 0.803	193 0.751	493 0.720
ALL	499 0.712				

HEPATITIS DATASET

k	**6**	**5**	**4**	**3**	**2**
# of observations with similarity at least k, and accuracy on these	0	0	21 0.998	6 0.960	40 0.845
ALL	56 0.811				

IONOSPHERE DATASET

k	**6**	**5**	**4**	**3**	**2**
# of observations with similarity at least k, and accuracy on these	0	1 1	7 0.990	61 0.948	170 0.855
ALL	171 0.855				

MUSHROOM DATASET

k	**6**	**5**	**4**	**3**	**2**
# of observations with similarity at least k, and accuracy on these		1693 0.999	1896 0.998	1989 0.997	2068 0.993
ALL	2117 0.981				

TIC-TAC-TOE DATASET

k	**6**	**5**	**4**	**3**	**2**
# of observations with similarity at least k, and accuracy on these	0	7 0.969	287 0.899	473 0.900	479 0.900
ALL	479 0.900				

VOTING DATASET

k	6	5	4	3	2
# of observations with similarity at least k, and accuracy on these	1 1	2 1	13 0.996	47 0.957	76 0.935
ALL	80 0.928				

WISCONSIN BREAST CANCER DATASET

k	6	5	4	3	2
# of observations with similarity at least k, and accuracy on these	3 1	9 0.992	36 0.983	121 0.951	185 0.927
ALL	193 0.926				

AVERAGE OF AVERAGE PREDICTIONS

k	6	5	4	3	2
Accuracy on observations of similarity at least k	1	0.974	0.950	0.897	0.858
ALL	0.851				

A2. Average cross-validation accuracy on observations of at least a given similarity, for each of the learning algorithms, averaged over all data-sets.

For values of k between 2 and 6, and for each of the seven learning algorithms, the following tables show the following: (1) the average, over all nine data-sets, of the 10-times 2-fold cross-validation estimate on observations of similarity at least k, and (2) (labeled ‘ALL’) the average accuracy, over all nine data-sets, on all non-training observations.

LAD

k	6	5	4	3	2
Accuracy on observations of similarity at least k	1	0.965	0.943	0.893	0.850
ALL	0.842				

SEE5

k	6	5	4	3	2
Accuracy on observations of similarity at least k	1	0.984	0.966	0.928	0.880
ALL	0.871				

SMO

k	6	5	4	3	2
Accuracy on observations of similarity at least k	1	0.972	0.967	0.913	0.874
ALL	0.868				

SIMPLELOGISTIC

k	6	5	4	3	2
Accuracy on observations of similarity at least k	1	0.972	0.960	0.913	0.874
ALL	0.869				

MULTILAYERPERCETRON

k	6	5	4	3	2
Accuracy on observations of similarity at least k	1	0.991	0.958	0.903	0.864
ALL	0.857				

IB3

k	6	5	4	3	2
Accuracy on observations of similarity at least k	1	0.979	0.942	0.881	0.842
ALL	0.840				

J48

k	6	5	4	3	2
Accuracy on observations of similarity at least k	1	0.967	0.941	0.893	0.851
ALL	0.843				

AVERAGE OF AVERAGE PREDICTIONS

k	6	5	4	3	2
Accuracy on observations of similarity at least k	1	0.976	0.954	0.904	0.862
ALL	0.856				

A3. Average cross-validation accuracy on observations of at most a given Hamming distance, for each of the data-sets, averaged over all learning algorithms.

For values of d between 1 and 16, and for each of the nine data-sets the following table shows the following: (1) in the small boxes, the average (over the 10 cross-validations) of the numbers of non-training observations having at most Hamming distance d to the training set, and (2) the average, over all seven learning algorithms, of the 10-times 2-fold cross-validation estimate on observations of Hamming distance at most d.

	hea	**pid**	**GermanCredit**	**hepatitis**	**ionosphere**	**mushroom**	**tic-tac-toe**	**vot**	**bcw**	**AVERAGE**
HD=1	10 0.936	3 0.980	2 0.858	1 0.996	15 0.949	1581 0.993	0	36 0.964	49 0.984	**0.958**
HD<=2	25 0.923	13 0.842	5 0.854	6 0.996	43 0.965	1980 0.995	325 0.917	64 0.938	91 0.986	**0.935**
HD<=3	45 0.881	38 0.833	15 0.867	16 0.951	62 0.949	2023 0.994	394 0.891	76 0.929	120 0.978	**0.919**
HD<=4	72 0.853	75 0.832	34 0.825	29 0.886	79 0.934	2048 0.992	469 0.901	80 0.929	144 0.972	**0.903**
HD<=5	98 0.829	113 0.797	84 0.819	41 0.855	98 0.925	2065 0.990	479 0.900	80 0.929	159 0.964	**0.890**
HD<=6	120 0.813	151 0.769	146 0.790	50 0.823	115 0.901	2078 0.988	479 0.900	80 0.929	172 0.951	**0.874**
HD<=7	131 0.809	178 0.761	226 0.776	54 0.818	128 0.886	2090 0.987	479 0.900	80 0.929	181 0.943	**0.868**
HD<=8	138 0.806	190 0.752	311 0.761	55 0.816	141 0.873	2098 0.985	479 0.900	80 0.929	186 0.934	**0.862**
HD<=9	143 0.801	195 0.749	388 0.746	56 0.815	150 0.865	2107 0.983	479 0.900	80 0.929	190 0.929	**0.857**
HD<=10	145 0.801	196 0.748	436 0.737	56 0.815	158 0.861	2110 0.982	479 0.900	80 0.929	191 0.927	**0.856**
HD<=11	146 0.800	196 0.747	473 0.728	56 0.815	163 0.857	2112 0.981	479 0.900	80 0.929	192 0.927	**0.854**
HD<=12	147 0.800	196 0.747	489 0.722	56 0.815	166 0.855	2114 0.981	479 0.900	80 0.929	192 0.926	**0.853**
HD<=13	147 0.800	196 0.747	495 0.721	56 0.815	169 0.854	2116 0.980	479 0.900	80 0.929	193 0.926	**0.852**
HD<=14	147 0.800	196 0.747	498 0.720	56 0.815	170 0.853	2117 0.980	479 0.900	80 0.929	193 0.926	**0.852**
HD<=15	148 0.800	196 0.747	498 0.720	56 0.815	170 0.853	2117 0.979	479 0.900	80 0.929	193 0.926	**0.852**
HD<=16	148 0.800	196 0.747	499 0.720	56 0.815	170 0.853	2117 0.979	479 0.900	80 0.929	193 0.926	**0.852**

A4. Average cross-validation accuracy on observations of at most a given Hamming distance, for each of the learning algorithms, averaged over all the data-sets.

For values of d between 1 and 16, and for each of the learning algorithms, the following table shows the average, over all nine data-sets of the 10-times 2-fold cross-validation estimate on observations of Hamming distance at most d.

AVERAGE PREDICTION FOR ALL METHODS

	LAD	SEE5	SMO	SimpleLogistic	MultilayerPerceptron	IB3	J48	AVERAGE
HD=1	0.892	0.967	0.967	0.973	0.967	0.968	0.969	**0.958**
HD<=2	0.907	0.936	0.952	0.951	0.937	0.930	0.932	**0.935**
HD<=3	0.906	0.919	0.933	0.934	0.927	0.899	0.918	**0.919**
HD<=4	0.894	0.899	0.917	0.913	0.907	0.889	0.900	**0.903**
HD<=5	0.880	0.887	0.903	0.903	0.893	0.875	0.886	**0.890**
HD<=6	0.862	0.871	0.888	0.887	0.878	0.863	0.868	**0.874**
HD<=7	0.854	0.865	0.884	0.883	0.873	0.856	0.860	**0.868**
HD<=8	0.848	0.859	0.878	0.878	0.867	0.850	0.853	**0.862**
HD<=9	0.844	0.855	0.874	0.873	0.862	0.846	0.848	**0.857**
HD<=10	0.843	0.852	0.872	0.872	0.860	0.843	0.846	**0.855**
HD<=11	0.841	0.851	0.870	0.870	0.858	0.841	0.845	**0.854**
HD<=12	0.840	0.850	0.869	0.869	0.857	0.840	0.844	**0.853**
HD<=13	0.840	0.849	0.869	0.869	0.856	0.840	0.843	**0.852**
HD<=14	0.840	0.849	0.869	0.869	0.856	0.840	0.843	**0.852**
HD<=15	0.840	0.849	0.869	0.869	0.856	0.840	0.843	**0.852**
HD<=16	0.840	0.849	0.869	0.869	0.856	0.840	0.843	**0.852**

A5.1 Comparing Hamming distance and similarity approaches: for each data set

The following tables show the ratios of the average cross-validation error estimates on similarity at least k to the average cross-validation error estimates on observations of Hamming distance at most d for each of the data-sets, averaged over all the learning algorithms.

CLEVELAND HEART DISEASE DATASET

k:		6	5	4	3	2
		1	0.974	0.893	0.814	0.802
HD=1	0.936	0	0.406	1.672	2.906	3.094
HD<=2	0.923	0	0.338	1.390	2.416	2.571
HD<=3	0.881	0	0.218	0.899	1.563	1.664
HD<=4	0.853	0	0.177	0.728	1.265	1.347
HD<=5	0.829	0	0.152	0.626	1.088	1.158
HD<=6	0.813	0	0.139	0.572	0.995	1.059
HD<=7	0.809	0	0.136	0.560	0.974	1.037
HD<=8	0.806	0	0.134	0.552	0.959	1.021
HD<=9	0.801	0	0.131	0.538	0.935	0.995
HD<=10	0.801	0	0.131	0.538	0.935	0.995
HD<=11	0.800	0	0.130	0.535	0.930	0.990
HD<=12	0.800	0	0.130	0.535	0.930	0.990
HD<=13	0.800	0	0.130	0.535	0.930	0.990
HD<=14	0.800	0	0.130	0.535	0.930	0.990
HD<=15	0.800	0	0.130	0.535	0.930	0.990

DIABETES DATASET

k:		6	5	4	3	2
		1	1	0.987	0.790	0.746
HD=1	0.980	0	0	0.650	10.500	12.700
HD<=2	0.842	0	0	0.082	1.329	1.608
HD<=3	0.833	0	0	0.078	1.257	1.521
HD<=4	0.832	0	0	0.077	1.250	1.512
HD<=5	0.797	0	0	0.064	1.034	1.251
HD<=6	0.769	0	0	0.056	0.909	1.100
HD<=7	0.761	0	0	0.054	0.879	1.063
HD<=8	0.752	0	0	0.052	0.847	1.024
HD<=9	0.749	0	0	0.052	0.837	1.012
HD<=10	0.748	0	0	0.052	0.833	1.008
HD<=11	0.747	0	0	0.051	0.830	1.004
HD<=12	0.747	0	0	0.051	0.830	1.004

GERMAN CREDIT DATASET (nominal)

k:		5	4	3	2
		0.857	0.803	0.751	0.720
HD=1	0.858	1.007	1.387	1.754	1.972
HD<=2	0.854	0.979	1.349	1.705	1.918
HD<=3	0.867	1.075	1.481	1.872	2.105
HD<=4	0.825	0.817	1.126	1.423	1.600
HD<=5	0.819	0.790	1.088	1.376	1.547
HD<=6	0.790	0.681	0.938	1.186	1.333
HD<=7	0.776	0.638	0.879	1.112	1.250
HD<=8	0.761	0.598	0.824	1.042	1.172
HD<=9	0.746	0.563	0.776	0.980	1.102
HD<=10	0.737	0.544	0.749	0.947	1.065
HD<=11	0.728	0.526	0.724	0.915	1.029
HD<=12	0.722	0.514	0.709	0.896	1.007
HD<=13	0.721	0.513	0.706	0.892	1.004
HD<=14	0.720	0.511	0.704	0.889	1.000
HD<=15	0.720	0.511	0.704	0.889	1.000

HD<=16	**0.720**	0.511	0.704	0.889	1.000

IONOSPHERE DATASET

k:		**5**	**4**	**3**	**2**
		1	**0.990**	**0.948**	**0.855**
HD=1	**0.949**	0	0.196	1.020	2.843
HD<=2	**0.965**	0	0.286	1.486	4.143
HD<=3	**0.949**	0	0.196	1.020	2.843
HD<=4	**0.934**	0	0.152	0.788	2.197
HD<=5	**0.925**	0	0.133	0.693	1.933
HD<=6	**0.901**	0	0.101	0.525	1.465
HD<=7	**0.886**	0	0.088	0.456	1.272
HD<=8	**0.873**	0	0.079	0.409	1.142
HD<=9	**0.865**	0	0.074	0.385	1.074
HD<=10	**0.861**	0	0.072	0.374	1.043
HD<=11	**0.857**	0	0.070	0.364	1.014
HD<=12	**0.855**	0	0.069	0.359	1.000
HD<=13	**0.854**	0	0.068	0.356	0.993
HD<=14	**0.853**	0	0.068	0.354	0.986
HD<=15	**0.853**	0	0.068	0.354	0.986
HD<=16	**0.853**	0	0.068	0.354	0.986

MUSHROOM DATASET

k:		5	4	3	2
		0.999	0.998	0.997	0.993
HD=1	0.993	0.143	0.286	0.430	1.002
HD<=2	0.995	0.188	0.377	0.565	1.319
HD<=3	0.994	0.162	0.323	0.485	1.132
HD<=4	0.992	0.128	0.255	0.383	0.893
HD<=5	0.990	0.102	0.204	0.305	0.713
HD<=6	0.988	0.086	0.172	0.259	0.604
HD<=7	0.987	0.075	0.151	0.226	0.527
HD<=8	0.985	0.066	0.132	0.198	0.463
HD<=9	0.983	0.058	0.116	0.174	0.406
HD<=10	0.982	0.055	0.111	0.166	0.387
HD<=11	0.981	0.053	0.106	0.160	0.373
HD<=12	0.981	0.052	0.104	0.155	0.363
HD<=13	0.980	0.050	0.100	0.150	0.350
HD<=14	0.980	0.049	0.098	0.147	0.343
HD<=15	0.979	0.048	0.097	0.145	0.339
HD<=16	0.979	0.048	0.097	0.145	0.338

TIC-TAC-TOE DATASET

k:		5	4	3	2
		0.969	0.899	0.900	0.900
HD<=2	0.917	0.373	1.217	1.205	1.205
HD<=3	0.891	0.284	0.927	0.917	0.917
HD<=4	0.901	0.313	1.020	1.010	1.010
HD<=5	0.900	0.310	1.010	1.000	1.000

VOTING DATASET

k:		6	5	4	3	2
		1	1	0.996	0.957	0.935
HD=1	0.964	0	0	0.111	1.194	1.806
HD<=2	0.938	0	0	0.065	0.694	1.048
HD<=3	0.929	0	0	0.056	0.606	0.915
HD<=4	0.929	0	0	0.056	0.606	0.915
HD<=5	0.929	0	0	0.056	0.606	0.915

WISCONSIN BREAST CANCER DATASET

k:		6	5	4	3	2
		1	0.992	0.983	0.951	0.927
HD=1	0.984	0	0.500	1.063	3.063	4.562
HD<=2	0.986	0	0.571	1.214	3.500	5.214
HD<=3	0.978	0	0.364	0.773	2.227	3.318
HD<=4	0.972	0	0.286	0.607	1.750	2.607
HD<=5	0.964	0	0.222	0.472	1.361	2.028
HD<=6	0.951	0	0.163	0.347	1.000	1.490
HD<=7	0.943	0	0.140	0.298	0.860	1.281
HD<=8	0.934	0	0.121	0.258	0.742	1.106
HD<=9	0.929	0	0.113	0.239	0.690	1.028
HD<=10	0.927	0	0.110	0.233	0.671	1.000
HD<=11	0.927	0	0.110	0.233	0.671	1.000
HD<=12	0.926	0	0.108	0.230	0.662	0.986
HD<=13	0.926	0	0.108	0.230	0.662	0.986

The following table shows the ratios for the averages, over the data-sets, of the two error estimates.

AVERAGE ERROR RATIO

k:		6	5	4	3	2
		1	0.974	0.950	0.897	0.858
HD=1	0.958	0	0.619	1.190	2.452	3.381
HD<=2	0.935	0	0.400	0.769	1.585	2.185
HD<=3	0.919	0	0.321	0.617	1.272	1.753
HD<=4	0.903	0	0.268	0.515	1.062	1.464
HD<=5	0.890	0	0.236	0.455	0.936	1.291
HD<=6	0.874	0	0.206	0.397	0.817	1.127
HD<=7	0.868	0	0.197	0.379	0.780	1.076
HD<=8	0.862	0	0.188	0.362	0.746	1.029
HD<=9	0.857	0	0.182	0.350	0.720	0.993
HD<=10	0.856	0	0.181	0.347	0.715	0.986
HD<=11	0.854	0	0.178	0.342	0.705	0.973
HD<=12	0.853	0	0.177	0.340	0.701	0.966
HD<=13	0.852	0	0.176	0.339	0.698	0.962
HD<=14	0.852	0	0.176	0.339	0.698	0.962
HD<=15	0.852	0	0.176	0.339	0.698	0.962
HD<=16	0.852	0	0.176	0.339	0.698	0.962

A5.2 Comparing Hamming distance and similarity approaches: for each learning algorithm

The following tables show the ratios of the average cross-validation error estimates on similarity at least k to the average cross-validation error estimates on observations of Hamming distance at most d for each learning algorithm, averaged over all the data-sets.

LAD

k:		6	5	4	3	2
		1	0.965	0.943	0.893	0.850
HD=1	0.892	0	0.324	0.528	0.991	1.389
HD<=2	0.907	0	0.376	0.613	1.151	1.613
HD<=3	0.906	0	0.372	0.606	1.138	1.596
HD<=4	0.894	0	0.330	0.538	1.009	1.415
HD<=5	0.880	0	0.292	0.475	0.892	1.250
HD<=6	0.862	0	0.254	0.413	0.775	1.087
HD<=7	0.854	0	0.240	0.390	0.733	1.027
HD<=8	0.848	0	0.230	0.375	0.704	0.987
HD<=9	0.844	0	0.224	0.365	0.686	0.962
HD<=10	0.843	0	0.223	0.363	0.682	0.955
HD<=11	0.841	0	0.220	0.358	0.673	0.943
HD<=12	0.840	0	0.219	0.356	0.669	0.938
HD<=13	0.840	0	0.219	0.356	0.669	0.938
HD<=14	0.840	0	0.219	0.356	0.669	0.938
HD<=15	0.840	0	0.219	0.356	0.669	0.938
HD<=16	0.840	0	0.219	0.356	0.669	0.938

SEE5

k:		6	5	4	3	2
		1	0.984	0.966	0.928	0.880
HD=1	0.967	0	0.485	1.030	2.182	3.636
HD<=2	0.936	0	0.250	0.531	1.125	1.875
HD<=3	0.919	0	0.198	0.420	0.889	1.481
HD<=4	0.899	0	0.158	0.337	0.713	1.188
HD<=5	0.887	0	0.142	0.301	0.637	1.062
HD<=6	0.871	0	0.124	0.264	0.558	0.930
HD<=7	0.865	0	0.119	0.252	0.533	0.889
HD<=8	0.859	0	0.113	0.241	0.511	0.851
HD<=9	0.855	0	0.110	0.234	0.497	0.828
HD<=10	0.852	0	0.108	0.230	0.486	0.811
HD<=11	0.851	0	0.107	0.228	0.483	0.805
HD<=12	0.850	0	0.107	0.227	0.480	0.800
HD<=13	0.849	0	0.106	0.225	0.477	0.795
HD<=14	0.849	0	0.106	0.225	0.477	0.795
HD<=15	0.849	0	0.106	0.225	0.477	0.795
HD<=16	0.849	0	0.106	0.225	0.477	0.795

SMO

k:		6	5	4	3	2
		1	0.972	0.967	0.913	0.874
HD=1	0.967	0	0.848	1.000	2.636	3.818
HD<=2	0.952	0	0.583	0.688	1.813	2.625
HD<=3	0.933	0	0.418	0.493	1.299	1.881
HD<=4	0.917	0	0.337	0.398	1.048	1.518
HD<=5	0.903	0	0.289	0.340	0.897	1.299
HD<=6	0.888	0	0.250	0.295	0.777	1.125
HD<=7	0.884	0	0.241	0.284	0.750	1.086
HD<=8	0.878	0	0.230	0.270	0.713	1.033
HD<=9	0.874	0	0.222	0.262	0.690	1.000
HD<=10	0.872	0	0.219	0.258	0.680	0.984
HD<=11	0.870	0	0.215	0.254	0.669	0.969
HD<=12	0.869	0	0.214	0.252	0.664	0.962
HD<=13	0.869	0	0.214	0.252	0.664	0.962
HD<=14	0.869	0	0.214	0.252	0.664	0.962
HD<=15	0.869	0	0.214	0.252	0.664	0.962
HD<=16	0.869	0	0.214	0.252	0.664	0.962

SIMPLELOGISTIC

k:		6	5	4	3	2
		1	0.972	0.960	0.913	0.874
HD=1	0.973	0	1.037	1.481	3.222	4.667
HD<=2	0.951	0	0.571	0.816	1.776	2.571
HD<=3	0.934	0	0.424	0.606	1.318	1.909
HD<=4	0.913	0	0.322	0.460	1.000	1.448
HD<=5	0.903	0	0.289	0.412	0.897	1.299
HD<=6	0.887	0	0.248	0.354	0.770	1.115
HD<=7	0.883	0	0.239	0.342	0.744	1.077
HD<=8	0.878	0	0.230	0.328	0.713	1.033
HD<=9	0.873	0	0.220	0.315	0.685	0.992
HD<=10	0.872	0	0.219	0.313	0.680	0.984
HD<=11	0.870	0	0.215	0.308	0.669	0.969
HD<=12	0.869	0	0.214	0.305	0.664	0.962
HD<=13	0.869	0	0.214	0.305	0.664	0.962
HD<=14	0.869	0	0.214	0.305	0.664	0.962
HD<=15	0.869	0	0.214	0.305	0.664	0.962
HD<=16	0.869	0	0.214	0.305	0.664	0.962

MULTILAYERPERCEPTRON

k:		6	5	4	3	2
		1	0.991	0.958	0.903	0.864
HD=1	0.967	0	0.273	1.273	2.939	4.121
HD<=2	0.937	0	0.143	0.667	1.540	2.159
HD<=3	0.927	0	0.123	0.575	1.329	1.863
HD<=4	0.907	0	0.097	0.452	1.043	1.462
HD<=5	0.893	0	0.084	0.393	0.907	1.271
HD<=6	0.878	0	0.074	0.344	0.795	1.115
HD<=7	0.873	0	0.071	0.331	0.764	1.071
HD<=8	0.867	0	0.068	0.316	0.729	1.023
HD<=9	0.862	0	0.065	0.304	0.703	0.986
HD<=10	0.860	0	0.064	0.300	0.693	0.971
HD<=11	0.858	0	0.063	0.296	0.683	0.958
HD<=12	0.857	0	0.063	0.294	0.678	0.951
HD<=13	0.856	0	0.063	0.292	0.674	0.944
HD<=14	0.856	0	0.063	0.292	0.674	0.944
HD<=15	0.856	0	0.063	0.292	0.674	0.944
HD<=16	0.856	0	0.063	0.292	0.674	0.944

IB3

k:		6	5	4	3	2
		1	0.979	0.942	0.881	0.842
HD=1	0.968	0	0.656	1.813	3.719	4.938
HD<=2	0.930	0	0.300	0.829	1.700	2.257
HD<=3	0.899	0	0.208	0.574	1.178	1.564
HD<=4	0.889	0	0.189	0.523	1.072	1.423
HD<=5	0.875	0	0.168	0.464	0.952	1.264
HD<=6	0.863	0	0.153	0.423	0.869	1.153
HD<=7	0.856	0	0.146	0.403	0.826	1.097
HD<=8	0.850	0	0.140	0.387	0.793	1.053
HD<=9	0.846	0	0.136	0.377	0.773	1.026
HD<=10	0.843	0	0.134	0.369	0.758	1.006
HD<=11	0.841	0	0.132	0.365	0.748	0.994
HD<=12	0.840	0	0.131	0.363	0.744	0.988
HD<=13	0.840	0	0.131	0.363	0.744	0.988
HD<=14	0.840	0	0.131	0.363	0.744	0.988
HD<=15	0.840	0	0.131	0.363	0.744	0.988
HD<=16	0.840	0	0.131	0.363	0.744	0.988

J48

k:		6	5	4	3	2
		1	0.967	0.941	0.893	0.851
HD=1	0.969	0	1.065	1.903	3.452	4.806
HD<=2	0.932	0	0.485	0.868	1.574	2.191
HD<=3	0.918	0	0.402	0.720	1.305	1.817
HD<=4	0.900	0	0.330	0.590	1.070	1.490
HD<=5	0.886	0	0.289	0.518	0.939	1.307
HD<=6	0.868	0	0.250	0.447	0.811	1.129
HD<=7	0.860	0	0.236	0.421	0.764	1.064
HD<=8	0.853	0	0.224	0.401	0.728	1.014
HD<=9	0.848	0	0.217	0.388	0.704	0.980
HD<=10	0.846	0	0.214	0.383	0.695	0.968
HD<=11	0.845	0	0.213	0.381	0.690	0.961
HD<=12	0.844	0	0.212	0.378	0.686	0.955
HD<=13	0.843	0	0.210	0.376	0.682	0.949
HD<=14	0.843	0	0.210	0.376	0.682	0.949
HD<=15	0.843	0	0.210	0.376	0.682	0.949
HD<=16	0.843	0	0.210	0.376	0.682	0.949

The following table shows the ratios for the averages, over the learning algorithms, of the two error estimates.

AVERAGE ERROR RATIO

k:		6	5	4	3	2
		1	0.976	0.954	0.904	0.862
HD=1	**0.958**	0	0.571	1.095	2.286	3.286
HD<=2	**0.935**	0	0.369	0.708	1.477	2.123
HD<=3	**0.919**	0	0.296	0.568	1.185	1.704
HD<=4	**0.903**	0	0.247	0.474	0.990	1.423
HD<=5	**0.890**	0	0.218	0.418	0.873	1.255
HD<=6	**0.874**	0	0.190	0.365	0.762	1.095
HD<=7	**0.868**	0	0.182	0.348	0.727	1.045
HD<=8	**0.862**	0	0.174	0.333	0.696	1.000
HD<=9	**0.857**	0	0.168	0.322	0.671	0.965
HD<=10	**0.855**	0	0.166	0.317	0.662	0.952
HD<=11	**0.854**	0	0.164	0.315	0.658	0.945
HD<=12	**0.853**	0	0.163	0.313	0.653	0.939
HD<=13	**0.852**	0	0.162	0.311	0.649	0.932
HD<=14	**0.852**	0	0.162	0.311	0.649	0.932
HD<=15	**0.852**	0	0.162	0.311	0.649	0.932
HD<=16	**0.852**	0	0.162	0.311	0.649	0.932

A6.1 Combining the Hamming distance and similarity approaches: for each data-set

The following tables show, for each data-set, the average, over all seven learning algorithms, of the 10-times 2-fold cross validation error estimates on observations of similarity at least k and of at most a given Hamming distance from the training set. The numbers in small boxes are the average numbers of observations of at least the given similarity and at most the given Hamming distance.

CLEVELAND HEART DISEASE

	Error rates				
k:	**2**	**3**	**4**	**5**	**6**
HD=1	9 0.050	7 0.054	5 0.038	1 0.028	1 0
HD<=2	23 0.074	18 0.065	9 0.048	2 0.026	1 0
HD<=3	42 0.121	29 0.117	11 0.046	3 0.026	1 0
HD<=4	65 0.259	42 0.201	13 0.099	3 0.026	1 0
HD<=5	86 0.176	50 0.177	14 0.104	3 0.026	1 0
HD<=6	102 0.189	54 0.183	14 0.104	3 0.026	1 0
HD<=7	109 0.194	54 0.186	14 0.106	3 0.026	1 0
HD<=8	112 0.196	55 0.187	14 0.106	3 0.026	1 0
HD<=9	114 0.198	55 0.187	14 0.106	3 0.026	1 0
HD<=10	114 0.198	55 0.187	14 0.106	3 0.026	1 0
HD<=11	114 0.198	55 0.187	14 0.106	3 0.026	1 0
HD<=12	115 0.198	55 0.187	14 0.106	3 0.026	1 0
HD<=13	115 0.198	55 0.187	14 0.106	3 0.026	1 0
HD<=14	115 0.198	55 0.187	14 0.106	3 0.026	1 0
HD<=15	115 0.198	55 0.187	14 0.106	3 0.026	1 0

DIABETES

Error rates

k:	**2**	**3**	**4**	**5**	**6**
HD=1	$\boxed{2}$ 0	$\boxed{2}$ 0	$\boxed{1}$ 0	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=2	$\boxed{13}$ 0.133	$\boxed{9}$ 0.132	$\boxed{1}$ 0.036	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=3	$\boxed{37}$ 0.171	$\boxed{23}$ 0.149	$\boxed{1}$ 0.020	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=4	$\boxed{73}$ 0.169	$\boxed{40}$ 0.158	$\boxed{2}$ 0.019	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=5	$\boxed{111}$ 0.204	$\boxed{54}$ 0.194	$\boxed{2}$ 0.025	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=6	$\boxed{147}$ 0.234	$\boxed{66}$ 0.208	$\boxed{2}$ 0.022	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=7	$\boxed{173}$ 0.240	$\boxed{72}$ 0.209	$\boxed{2}$ 0.022	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=8	$\boxed{184}$ 0.249	$\boxed{74}$ 0.211	$\boxed{2}$ 0.022	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=9	$\boxed{188}$ 0.251	$\boxed{74}$ 0.211	$\boxed{2}$ 0.022	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=10	$\boxed{189}$ 0.254	$\boxed{74}$ 0.211	$\boxed{2}$ 0.022	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=11	$\boxed{189}$ 0.254	$\boxed{74}$ 0.211	$\boxed{2}$ 0.022	$\boxed{1}$ 0	$\boxed{1}$ 0
HD<=12	$\boxed{190}$ 0.254	$\boxed{74}$ 0.211	$\boxed{2}$ 0.022	$\boxed{1}$ 0	$\boxed{1}$ 0

GERMAN CREDIT

Error rates

k:	**2**	**3**	**4**	**5**
HD=1	2 0.131	2 0.125	1 0.243	
HD<=2	5 0.147	3 0.160	1 0.353	
HD<=3	15 0.129	13 0.131	4 0.125	
HD<=4	34 0.177	29 0.178	8 0.138	1 0.143
HD<=5	84 0.182	69 0.177	17 0.106	1 0.143
HD<=6	146 0.212	115 0.198	24 0.150	1 0.143
HD<=7	225 0.228	170 0.217	32 0.181	1 0.143
HD<=8	310 0.243	222 0.228	38 0.194	1 0.143
HD<=9	386 0.257	263 0.239	40 0.196	1 0.143
HD<=10	433 0.266	283 0.244	41 0.197	1 0.143
HD<=11	468 0.266	292 0.245	41 0.197	1 0.143
HD<=12	483 0.276	294 0.248	41 0.197	1 0.143
HD<=13	490 0.279	295 0.248	41 0.197	1 0.143
HD<=14	492 0.281	295 0.248	41 0.197	1 0.143
HD<=15	492 0.281	295 0.248	41 0.197	1 0.143
HD<=16	493 0.281	295 0.248	41 0.197	1 0.143

HEPATITIS

Error rates

k:	2	3	4
HD=1	1 0	1 0	1 0
HD<=2	6 0.001	3 0	2 0
HD<=3	15 0.037	5 0.014	2 0
HD<=4	24 0.083	5 0.020	2 0
HD<=5	33 0.108	5 0.020	2 0
HD<=6	37 0.122	6 0.020	2 0
HD<=7	38 0.129	6 0.020	2 0
HD<=8	40 0.125	6 0.020	2 0
HD<=9	40 0.125	6 0.020	2 0

IONOSPHERE

Error rates

k:	2	3	4	5
HD=1	15 0.038	13 0.032	4 0.018	1 0
HD<=2	43 0.031	31 0.022	6 0.014	1 0
HD<=3	61 0.047	42 0.030	7 0.010	1 0
HD<=4	79 0.063	49 0.033	7 0.010	1 0
HD<=5	98 0.073	55 0.034	7 0.010	1 0
HD<=6	115 0.097	58 0.044	7 0.010	1 0
HD<=7	128 0.113	58 0.045	7 0.010	1 0
HD<=8	141 0.125	60 0.049	7 0.010	1 0
HD<=9	150 0.134	60 0.052	7 0.010	1 0
HD<=10	158 0.138	61 0.053	7 0.010	1 0
HD<=11	163 0.142	61 0.053	7 0.010	1 0
HD<=12	166 0.144	61 0.053	7 0.010	1 0
HD<=13	169 0.145	61 0.053	7 0.010	1 0
HD<=14	170 0.145	61 0.053	7 0.010	1 0
HD<=15	170 0.146	61 0.053	7 0.010	1 0
HD<=16	170 0.146	61 0.053	7 0.010	1 0

MUSHROOM

Error rates

k:	2	3	4	5
HD=1	1581 0.002	1573 0.002	1536 0.001	1418 0
HD<=2	1979 0.004	1957 0.002	1884 0.002	1689 0.001
HD<=3	2022 0.005	1983 0.003	1896 0.002	1693 0.001
HD<=4	2045 0.006	1989 0.003	1896 0.002	1693 0.001
HD<=5	2057 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=6	2064 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=7	2067 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=8	2068 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=9	2068 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=10	2068 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=11	2068 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=12	2068 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=13	2068 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=14	2068 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=15	2068 0.007	1989 0.003	1896 0.002	1693 0.001
HD<=16	2068 0.007	1989 0.003	1896 0.002	1693 0.001

TIC-TAC-TOE

Error rates

AVERAGE

k:	**2**	**3**	**4**	**5**
HD=1				
HD<=2	325 0.083	325 0.083	242 0.084	7 0.036
HD<=3	394 0.109	394 0.109	276 0.102	7 0.036
HD<=4	469 0.099	464 0.100	287 0.101	7 0.036
HD<=5	479 0.100	473 0.101	287 0.101	7 0.036

VOTING

Error rates

AVERAGE

k:	**2**	**3**	**4**	**5**	**6**
HD=1	35 0.032	29 0.021	12 0.004	2 0	1 0
HD<=2	61 0.057	43 0.037	13 0.004	2 0	1 0
HD<=3	73 0.064	46 0.044	13 0.004	2 0	1 0
HD<=4	76 0.065	47 0.044	13 0.004	2 0	1 0
HD<=5	76 0.065	47 0.044	13 0.004	2 0	1 0

WISCONSIN BREAST CANCER

Error rates

AVERAGE

k:	**2**	**3**	**4**	**5**	**6**
HD=1	48 0.016	43 0.015	22 0.014	7 0.001	2 0
HD<=2	91 0.014	76 0.015	32 0.016	8 0.008	3 0
HD<=3	119 0.022	93 0.017	35 0.015	9 0.008	3 0
HD<=4	141 0.028	107 0.021	36 0.015	9 0.008	3 0
HD<=5	156 0.036	114 0.029	36 0.016	9 0.008	3 0
HD<=6	168 0.049	117 0.037	36 0.016	9 0.008	3 0
HD<=7	175 0.057	119 0.041	36 0.016	9 0.008	3 0
HD<=8	179 0.065	120 0.046	36 0.016	9 0.008	3 0
HD<=9	183 0.070	121 0.049	36 0.016	9 0.008	3 0
HD<=10	184 0.071	121 0.049	36 0.016	9 0.008	3 0
HD<=11	184 0.072	121 0.049	36 0.016	9 0.008	3 0
HD<=12	185 0.073	121 0.049	36 0.016	9 0.008	3 0
HD<=13	185 0.073	121 0.049	36 0.016	9 0.008	3 0

The following table shows the ratios for the averages, over the datasets, of the error estimates on observations of similarity at least k and of at most a given Hamming distance from the training set.

AVERAGE OF AVERAGE ERROR RATE

k:	**2**	**3**	**4**	**5**	**6**
HD=1	0.034	0.031	0.040	0.005	0
HD<=2	0.060	0.057	0.062	0.010	0
HD<=3	0.078	0.068	0.036	0.010	0
HD<=4	0.105	0.084	0.043	0.027	0
HD<=5	0.106	0.086	0.041	0.027	0
HD<=6	0.119	0.093	0.046	0.027	0
HD<=7	0.126	0.096	0.049	0.027	0
HD<=8	0.131	0.099	0.051	0.027	0
HD<=9	0.134	0.100	0.051	0.027	0
HD<=10	0.136	0.101	0.051	0.027	0
HD<=11	0.137	0.101	0.051	0.027	0
HD<=12	0.138	0.102	0.051	0.027	0
HD<=13	0.138	0.102	0.051	0.027	0
HD<=14	0.139	0.102	0.051	0.027	0
HD<=15	0.139	0.102	0.051	0.027	0
HD<=16	0.139	0.102	0.051	0.027	

A6.2 Combining the Hamming distance and similarity approaches: for each learning algorithm

The following tables show, for each learning algorithm, the average, over all nine data-sets, of the 10-times 2-fold cross validation error estimates on observations of similarity at least k and of at most a given Hamming distance from the training set.

LAD

k:	2	3	4	5	6
HD=1	0.050	0.049	0.112	0.007	0
HD<=2	0.055	0.052	0.110	0.006	0
HD<=3	0.079	0.070	0.044	0.006	0
HD<=4	0.097	0.082	0.049	0.034	0
HD<=5	0.096	0.085	0.044	0.034	0
HD<=6	0.110	0.095	0.051	0.034	0
HD<=7	0.117	0.099	0.055	0.034	0
HD<=8	0.122	0.102	0.056	0.034	0
HD<=9	0.126	0.103	0.056	0.034	0
HD<=10	0.127	0.104	0.057	0.034	0
HD<=11	0.127	0.104	0.057	0.034	0
HD<=12	0.128	0.104	0.057	0.034	0
HD<=13	0.129	0.104	0.057	0.034	0
HD<=14	0.129	0.104	0.057	0.034	0
HD<=15	0.129	0.104	0.057	0.034	0
HD<=16	0.129	0.104	0.057	0.034	

SEE5

k:	2	3	4	5	6
HD=1	0.029	0.022	0.017	0.005	0
HD<=2	0.068	0.062	0.064	0.020	0
HD<=3	0.083	0.073	0.047	0.020	0
HD<=4	0.113	0.089	0.056	0.032	0
HD<=5	0.112	0.090	0.055	0.032	0
HD<=6	0.128	0.097	0.059	0.032	0
HD<=7	0.136	0.102	0.064	0.032	0
HD<=8	0.140	0.104	0.065	0.032	0
HD<=9	0.144	0.106	0.066	0.032	0
HD<=10	0.146	0.107	0.066	0.032	0
HD<=11	0.146	0.107	0.066	0.032	0
HD<=12	0.148	0.107	0.066	0.032	0
HD<=13	0.148	0.108	0.066	0.032	0
HD<=14	0.148	0.108	0.066	0.032	0
HD<=15	0.149	0.108	0.066	0.032	0
HD<=16	0.149	0.108	0.066	0.032	

SMO

k:	2	3	4	5	6
HD=1	0.033	0.037	0.051	0.005	0
HD<=2	0.049	0.047	0.048	0.005	0
HD<=3	0.065	0.054	0.021	0.005	0
HD<=4	0.094	0.071	0.027	0.028	0
HD<=5	0.096	0.075	0.025	0.028	0
HD<=6	0.109	0.080	0.028	0.028	0
HD<=7	0.114	0.082	0.031	0.028	0
HD<=8	0.119	0.084	0.033	0.028	0
HD<=9	0.122	0.086	0.033	0.028	0
HD<=10	0.124	0.087	0.033	0.028	0
HD<=11	0.124	0.087	0.033	0.028	0
HD<=12	0.125	0.087	0.033	0.028	0
HD<=13	0.126	0.087	0.033	0.028	0
HD<=14	0.126	0.087	0.033	0.028	0
HD<=15	0.126	0.087	0.033	0.028	0
HD<=16	0.126	0.087	0.033	0.028	

SimpleLogistic

k:	2	3	4	5	6
HD=1	0.027	0.030	0.039	0.005	0
HD<=2	0.049	0.047	0.048	0.006	0
HD<=3	0.067	0.057	0.026	0.006	0
HD<=4	0.095	0.074	0.036	0.029	0
HD<=5	0.095	0.077	0.032	0.029	0
HD<=6	0.109	0.083	0.036	0.029	0
HD<=7	0.114	0.084	0.039	0.029	0
HD<=8	0.118	0.086	0.040	0.029	0
HD<=9	0.121	0.088	0.040	0.029	0
HD<=10	0.123	0.088	0.040	0.029	0
HD<=11	0.124	0.089	0.040	0.029	0
HD<=12	0.125	0.089	0.040	0.029	0
HD<=13	0.125	0.089	0.040	0.029	0
HD<=14	0.125	0.089	0.040	0.029	0
HD<=15	0.126	0.089	0.040	0.029	0
HD<=16	0.126	0.089	0.040	0.029	

MultilayerPerceptron

k:	2	3	4	5	6
HD=1	0.033	0.033	0.036	0.005	0
HD<=2	0.061	0.061	0.052	0.006	0
HD<=3	0.073	0.064	0.027	0.006	0
HD<=4	0.104	0.081	0.033	0.010	0
HD<=5	0.104	0.083	0.031	0.010	0
HD<=6	0.117	0.089	0.036	0.010	0
HD<=7	0.123	0.092	0.039	0.010	0
HD<=8	0.127	0.094	0.040	0.010	0
HD<=9	0.130	0.096	0.041	0.010	0
HD<=10	0.132	0.096	0.041	0.010	0
HD<=11	0.133	0.097	0.041	0.010	0
HD<=12	0.135	0.097	0.041	0.010	0
HD<=13	0.135	0.097	0.041	0.010	0
HD<=14	0.135	0.097	0.041	0.010	0
HD<=15	0.136	0.097	0.041	0.010	0
HD<=16	0.136	0.097	0.041	0.010	

IB3

k:	**2**	**3**	**4**	**5**	**6**
HD=1	0.032	0.026	0.012	0.005	0
HD<=2	0.070	0.065	0.043	0.007	0
HD<=3	0.098	0.085	0.043	0.007	0
HD<=4	0.123	0.101	0.050	0.021	0
HD<=5	0.125	0.105	0.048	0.021	0
HD<=6	0.136	0.111	0.053	0.021	0
HD<=7	0.143	0.114	0.056	0.021	0
HD<=8	0.148	0.117	0.057	0.021	0
HD<=9	0.152	0.118	0.058	0.021	0
HD<=10	0.154	0.119	0.058	0.021	0
HD<=11	0.155	0.119	0.058	0.021	0
HD<=12	0.157	0.120	0.058	0.021	0
HD<=13	0.157	0.120	0.058	0.021	0
HD<=14	0.158	0.120	0.058	0.021	0
HD<=15	0.158	0.120	0.058	0.021	0
HD<=16	0.158	0.120	0.058	0.021	

J48

k:	**2**	**3**	**4**	**5**	**6**
HD=1	0.032	0.021	0.011	0.005	0
HD<=2	0.071	0.068	0.068	0.021	0
HD<=3	0.083	0.073	0.045	0.021	0
HD<=4	0.112	0.089	0.053	0.033	0
HD<=5	0.111	0.090	0.052	0.033	0
HD<=6	0.127	0.096	0.057	0.033	0
HD<=7	0.135	0.100	0.061	0.033	0
HD<=8	0.140	0.103	0.063	0.033	0
HD<=9	0.144	0.105	0.063	0.033	0
HD<=10	0.146	0.106	0.063	0.033	0
HD<=11	0.146	0.106	0.063	0.033	0
HD<=12	0.147	0.107	0.063	0.033	0
HD<=13	0.148	0.107	0.063	0.033	0
HD<=14	0.148	0.107	0.063	0.033	0
HD<=15	0.148	0.107	0.063	0.033	0
HD<=16	0.148	0.107	0.063	0.033	

The following table shows the ratios for the averages, over the learning algorithms, of the error estimates on observations of similarity at least k and of at most a given Hamming distance from the training set.

AVERAGE OF AVERAGE ERROR RATE

k:	2	3	4	5	6
HD=1	0.034	0.031	0.040	0.005	0
HD<=2	0.060	0.057	0.062	0.010	0
HD<=3	0.078	0.068	0.036	0.010	0
HD<=4	0.105	0.084	0.043	0.027	0
HD<=5	0.106	0.086	0.041	0.027	0
HD<=6	0.119	0.093	0.046	0.027	0
HD<=7	0.126	0.096	0.049	0.027	0
HD<=8	0.131	0.099	0.051	0.027	0
HD<=9	0.134	0.100	0.051	0.027	0
HD<=10	0.136	0.101	0.051	0.027	0
HD<=11	0.137	0.101	0.051	0.027	0
HD<=12	0.138	0.102	0.051	0.027	0
HD<=13	0.138	0.102	0.051	0.027	0
HD<=14	0.139	0.102	0.051	0.027	0
HD<=15	0.139	0.102	0.051	0.027	0
HD<=16	0.139	0.102	0.051	0.027	