Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms

Remco R. Bouckaert[1,2] and Eibe Frank[2]

[1] Xtal Mountain Information Technology
215 Three Oaks Drive, Dairy Flat, Auckland, New Zealand
rrb@xm.co.nz
[2] Computer Science Department, University of Waikato
Private Bag 3105, Hamilton, New Zealand
{remco,eibe}@cs.waikato.ac.nz

Abstract. Empirical research in learning algorithms for classification tasks generally requires the use of significance tests. The quality of a test is typically judged on Type I error (how often the test indicates a difference when it should not) and Type II error (how often it indicates no difference when it should). In this paper we argue that the replicability of a test is also of importance. We say that a test has low replicability if its outcome strongly depends on the particular random partitioning of the data that is used to perform it. We present empirical measures of replicability and use them to compare the performance of several popular tests in a realistic setting involving standard learning algorithms and benchmark datasets. Based on our results we give recommendations on which test to use.

1 Introduction

Significance tests are often applied to compare performance estimates obtained by resampling methods—such as cross-validation [1]—that randomly partition data. In this paper we consider the problem that a test may be very sensitive to the particular random partitioning used in this process. If this is the case, it is possible that, using the same data, the same learning algorithms A and B, and the same significance test, one researcher finds that method A is preferable, while another finds that there is not enough evidence for this. Lack of replicability can also cause problems when "tuning" an algorithm: a test may judge favorably on the latest modification purely due to its sensitivity to the particular random number seed used to partition the data. In this paper we extend previous work on replicability [2, 3] by studying the replicability of some popular tests in a more realistic setting based on standard benchmark datasets taken from the UCI repository of machine learning problems [4].

The structure of the paper is as follows. In Section 2 we review how significance tests are used for comparing learning algorithms and introduce the notion of replicability. Section 3 discusses some popular tests in detail. Section 4 contains empirical results for these tests and highlights the lack of replicability of some of them. Section 5 summarizes the results and makes some recommendations based on our empirical findings.

2 Evaluating significance tests

We consider a scenario where we have a certain application domain and we are interested in the mean difference in accuracy between two classification algorithms in this domain, given that the two algorithms are trained on a dataset with N instances. We do not know the joint distribution underlying the domain and consequently cannot compute the difference exactly. Hence we need to estimate it, and, to check whether the estimated difference is likely to be a "true" difference, perform a significance test. To this end we also need to estimate the variance of the differences across different training sets.

Obtaining an unbiased estimate of the mean and variance of the difference is easy if there is a sufficient supply of data. In that case we can sample a number of training sets of size N, run the two learning algorithms on each of them, and estimate the difference in accuracy for each pair of classifiers on a large test set. The average of these differences is an estimate of the expected difference in generalization error across all possible training sets of size N, and their variance is an estimate of the variance. Then we can perform a paired *t*-test to check the null hypothesis that the mean difference is zero. The Type I error of a test is the probability that it rejects the null hypothesis incorrectly (i.e. it finds a "significant" difference although there is none). Type II error is the probability that the null hypothesis is not rejected when there actually is a difference. The test's Type I error will be close to the chosen significance level α.

In practice we often only have one dataset of size N and all estimates must be obtained from this one dataset. Different training sets are obtained by subsampling, and the instances not sampled for training are used for testing. For each training set S_i, $1 \leq i \leq k$, we get a matching pair of accuracy estimates and the difference x_i. The mean and variance of the differences x_i is used to estimate the mean and variance of the difference in generalization error across different training sets. Unfortunately this violates the independence assumption necessary for proper significance testing because we re-use the data to obtain the different x_i's. The consequence of this is that the Type I error exceeds the significance level. This is problematic because it is important for the researcher to be able to control the Type I error and know the probability of incorrectly rejecting the null hypothesis. Several heuristic versions of the *t*-test have been developed to alleviate this problem [5, 6].

In this paper we study the *replicability* of significance tests. Consider a test based on the accuracy estimates generated by cross-validation. Before the cross-validation is performed, the data is randomized so that each of the resulting training and test sets exhibits the same distribution. Ideally, we would like the test's outcome to be independent of the particular partitioning resulting from the randomization process because this would make it much easier to replicate experimental results published in the literature. However, in practice there is always a certain sensitivity to the partitioning used. To measure replicability we need to repeat the same test several times on the same data with different random partitionings—in this paper we use ten repetitions—and count how often the outcome is the same. Note that a test will have greater replicability than another test with the same Type I and Type II error if it is more consistent in its outcomes for each individual dataset.

We use two measures of replicability. The first measure, which we call *consistency*, is based on the raw counts. If the outcome is the same for every repetition of a test on the same data, we call the test *consistent*, and if there is a difference at most once, we call it *almost consistent*. This procedure is repeated with multiple datasets, and the fraction of outcomes for which a test is consistent or almost consistent is an indication of how replicable the test is. The second measure, which we call *replicability*, is based on the probability that two runs of the test on the same data set will produce the same outcome. This probability is never worse than 0.5. To estimate it we need to consider pairs of randomizations. If we have performed the test based on n different randomizations for a particular dataset then there are $\binom{n}{2}$ such pairs. Assume the tests rejects the null hypothesis for k $(0 \leq k \leq n)$ of the randomizations. Then there are $\binom{k}{2}$ rejecting pairs and $\binom{n-k}{2}$ accepting ones. Based on this the above probability can be estimated as $R(k,n) = (\binom{k}{2} + \binom{n-k}{2})/\binom{n}{2} = \frac{k(k-1)+(n-k)(n-k-1)}{n(n-1)}$. We use this probability to form a measure of replicability across different datasets. Assume there are m datasets and let i_k $(0 \leq k \leq n)$ be the number of datasets for which the test agrees k times (i.e. $\sum_{k=0}^{n} i_k = m$). Then we define replicability as $R = \sum_{k=0}^{n} \frac{i_k}{m} R(k,n)$. The larger the value of this measure, the more likely the test is to produce the same outcome for two different randomizations of a dataset.

3 Significance Tests

In this section we review some tests for comparing learning algorithms. Although testing is essential for empirical research, surprisingly little has been written on this topic.

3.1 The 5x2cv paired t-test

Dietterich [5] evaluates several significance tests by measuring their Type I and Type II error on artificial and real-world data. He finds that the paired t-test applied to random subsampling has an exceedingly large Type I error. In random subsampling a training set is drawn at random without replacement and the remainder of the data is used for testing. This is repeated a given number of times. In contrast to cross-validation, random subsampling does not ensure that the test sets do not overlap. Ten-fold cross-validation can be viewed as a special case of random subsampling repeated ten times, where 90% of the data is used for training, and it is guaranteed that the ten test sets do not overlap. The paired t-test based on ten-fold cross-validation fares better in the experiments in [5] but also exhibits an inflated Type I error. On one of the real-world datasets its Type I error is approximately twice the significance level.

As an alternative [5] proposes a heuristic test based on five runs of two-fold cross-validation, called “5x2cv paired t-test”. In an r-times k-fold cross-validations there are r, $r > 1$, runs and k, $k > 1$, folds. For each run j, $1 \leq j \leq r$, the data is randomly permutated and split into k subsets of equal size.[1] We call these i, $1 \leq i \leq k$, subsets the k folds of run j. We consider two learning schemes A and B and measure

[1] Of course, in some cases it may not be possible to split the data into subsets that have exactly the same size.

their respective accuracies a_{ij} and b_{ij} for fold i and run j. To obtain a_{ij} and b_{ij} the corresponding learning scheme is trained on all the data excluding that in fold i of run j and tested on the remainder. Note that exactly the same pair of training and test sets is used to obtain both a_{ij} and b_{ij}. That means a paired significance test is appropriate and we can consider the individual differences in accuracy $x_{ij} = a_{ij} - b_{ij}$ as the input for the test.

Let $x_{.j}$ denote the mean difference for a single run of 2-fold cross-validation, $x_{.j} = (x_{1j} + x_{2j})/2$. The variance is $\hat{\sigma}_j^2 = (x_{1j} - x_{.j})^2 + (x_{2j} - x_{.j})^2$. The 5x2cv paired t-test uses the following test statistic:

$$t = \frac{x_{11}}{\sqrt{\frac{1}{5}\sum_{j=1}^{5}\hat{\sigma}_j^2}}$$

This statistic is plugged into the Student-*t* distribution with five degrees of freedom. Note that the numerator only uses the term x_{11} and not the other differences x_{ij}. Consequently the outcome of the test is strongly dependent on the particular partitioning of the data used when the test is performed. Therefore it can be expected that the replicability of this test is not high. Our empirical evaluation demonstrates that this is indeed the case.

The empirical results in [5] show that the 5×2cv paired t-test has a Type I error at or below the significance level. However, they also show that it has a much higher Type II error than the standard t-test applied to ten-fold cross-validation. Consequently the former test is recommended in [5] when a low Type I error is essential, and the latter test otherwise.

The other two tests evaluated in [5] are McNemar's test and the test for the difference of two proportions. Both of these tests are based on a single train/test split and consequently cannot take variance due to the choice of training and test set into account. Of these two tests, McNemar's test performs better overall: it has an acceptable Type I error and the Type II error is only slightly lower than that of the 5×2cv paired t-test. However, because these two tests are inferior to the 5×2cv test, we will not consider them in our experiments.

3.2 Tests based on random subsampling

As mentioned above, Dietterich [5] found that the standard *t*-test has a high Type I error when used in conjunction with random subsampling. Nadeau and Bengio [6] observe that this is due to an underestimation of the variance because the samples are not independent (i.e. the different training and test sets overlap). Consequently they propose to correct the variance estimate by taking this dependency into account.

Let a_j and b_j be the accuracy of algorithms A and B respectively, measured on run j ($1 \leq j \leq n$). Assume that in each run n_1 instances are used for training, and the remaining n_2 instances for testing. Let x_j be the difference $x_j = a_j - b_j$, and $\hat{\mu}$ and $\hat{\sigma}^2$ the estimates of the mean and variance of the n differences. The statistic of the "corrected resampled *t*-test" is:

$$t = \frac{\frac{1}{n}\sum_{j=1}^{n} x_j}{\sqrt{(\frac{1}{n} + \frac{n_2}{n_1})\hat{\sigma}^2}}$$

This statistic is used in conjunction with the Student-*t* distribution and $n-1$ degrees of freedom. The only difference to the standard *t*-test is that the factor $\frac{1}{n}$ in the denominator has been replaced by the factor $\frac{1}{n}+\frac{n_2}{n_1}$. Nadeau and Bengio [6] suggest that "... normal usage would call for n_1 to be 5 or 10 times larger than n_2, ...".

Empirical results show that this test dramatically improves on the standard resampled *t*-test: the Type I error is close to the significance level, and, unlike McNemar's test and the 5×2cv test, it does not suffer from high Type II error [6].

3.3 Tests based on repeated k-fold cross validation

Here we consider tests based on r-times k-fold cross-validation where r and k can have any value. As in Section 3.1, we observe differences $x_{ij} = a_{ij} - b_{ij}$ for fold i and run j. One could simply use $m = \frac{1}{k.r}\sum_{i=1}^{k}\sum_{j=1}^{r} x_{ij}$ as an estimate for the mean and $\hat{\sigma}^2 = \frac{1}{k.r-1}\sum_{i=1}^{k}\sum_{j=1}^{r}(x_{ij}-m)^2$ as an estimate for the variance. Then, assuming the various values of x_{ij} are independent, the test statistic $t = m/\sqrt{(1/k.r)\hat{\sigma}^2}$ is distributed according to a *t*-distribution with $df = k.r-1$ degrees of freedom. Unfortunately, the independence assumption is highly flawed, and tests based on this assumption show very high Type I error, similar to plain subsampling.

However, the same variance correction as in the previous subsection can be performed here because cross-validation is a special case of random subsampling where we ensure that the test sets in one run do not overlap. (Of course, test sets from different runs will overlap.) This results in the following statistic:

$$t = \frac{\frac{1}{k.r}\sum_{i=1}^{k}\sum_{j=1}^{r} x_{ij}}{\sqrt{(\frac{1}{k.r}+\frac{n_2}{n_1})\hat{\sigma}^2}}$$

where n_1 is the number of instances used for training, and n_2 the number of instances used for testing. We call this test the "corrected repeated k-fold cv test".

4 Empirical evaluation

To evaluate how replicability affects the various tests, we performed experiments on a selection of datasets from the UCI repository [4]. We used naive Bayes, C4.5 [7], and the nearest neighbor classifier, with default settings as implemented in Weka[2] version 3.3 [1]. For tests that involve multiple folds, the folds were chosen using stratification, which ensures that the class distribution in the whole dataset is reflected in each of the folds. Each of the tests was run ten times for each pair of learning schemes and a 5% significance level was used in all tests unless stated otherwise.

4.1 Results for the 5x2cv paired t-test

Table 1 shows the datasets and their properties, and the results for the 5x2 cross validation test. The three right-most columns show the number of times the test does not reject

[2] Weka is freely available with source from http://www.cs.waikato.ac.nz/ml.

dataset	#inst.	#atts.	#cl.	NB vs C4.5	NB vs NN	C4.5 vs NN
anneal	898	38	5	4	4	10
arrhythmia	452	280	13	9	9	2
audiology	226	69	24	5	10	8
autos	205	25	6	10	7	10
balance-scale	625	4	3	1	4	7
breast-cancer	286	9	2	10	9	8
credit-rating	690	16	2	6	8	10
ecoli	336	8	8	7	10	10
German credit	1000	20	2	9	6	10
glass	214	9	6	6	6	9
heart-statlog	270	13	2	4	5	9
hepatitis	155	19	2	9	10	10
horse-colic	368	22	2	8	10	7
Hungarian heart disease	294	13	2	10	10	10
ionosphere	351	34	2	10	10	8
iris	150	4	3	10	10	10
labor	57	16	2	8	10	10
lymphography	148	18	4	9	10	10
pima-diabetes	768	8	2	10	6	7
primary-tumor	339	17	21	7	3	10
sonar	208	60	2	10	9	6
soybean	683	35	19	8	8	9
vehicle	846	18	4	0	0	9
vote	435	16	2	4	9	7
vowel	990	13	11	4	0	0
Wisconsin breast cancer	699	9	2	8	9	10
zoo	101	16	7	10	10	8
Consistent:				9	12	13
Almost consistent:				14	17	17
Replicability (R):				0.737	0.783	0.816

Table 1. The number of cases (#inst.), attributes (#atts.), and classes (#cl.) for each dataset; and the number of draws for each pair of classifiers based on the 5x2 cross validation test (NB = naive Bayes, NN = nearest neighbor).

the null hypothesis, i.e, the number of times the 5x2 cross validation test indicates that there is no difference between the corresponding pair of classifiers. For example, for the anneal dataset, the test indicates no difference between naive Bayes and C4.5 four times, so six times it does indicate a difference. Note that the same dataset, the same algorithm, the same settings, and the same significance test were used in each of the ten experiments. The only difference was in the way the dataset was split into the 2 folds in each of the 5 runs. Clearly, the test is very sensitive to the particular partitioning of the anneal data.

Looking at the column for naive Bayes vs. C4.5, this test could be used to justify the claim that the two perform the same for all datasets except the vehicle dataset just by choosing appropriate random number seeds. However, it could just as well be used to support the claim that the two algorithms perform differently in 19 out of 27 cases.

For some rows, the test consistently indicates no difference between any two of the three schemes, in particular for the iris and Hungarian heart disease datasets. However, most rows contain at least one cell where the outcomes of the test are not consistent.

The row labeled "consistent" at the bottom of the table lists the number of datasets for which all outcomes of the test are the same. These are calculated as the number of 0's and 10's in the column. For any of the compared schemes, less than 50% of the results turn out to be consistent.

Note that, it is possible that, when comparing algorithms A and B, sometimes A is preferred and sometimes B if the null hypothesis is rejected. However, closer inspection of the data reveals that this only happens when the null hypothesis is accepted most of the time, except for 2 or 3 runs. Consequently these cases do not contribute to the value of the consistency measure.

If we could accept that one outcome of the ten runs does not agree with the rest, we get the number labeled "almost consistent" in Table 1 (i.e. the number of 0's, 1's, 9's and 10's in a column). The 5x2 cross validation test is almost consistent in fewer than 66% of the cases, which is still a very low rate.

The last row shows the value of the replicability measure R for the three pairs of learning schemes considered. These results reflect the same behaviour as the consistency measures. The replicability values are pretty low considering that R cannot be smaller than 0.5.

4.2 Results for the corrected resampled *t*-test

In the resampling experiments, the data was randomized, 90% of it used for training, and the remaining 10% used to measure accuracy. This was repeated with a different random number seed for each run. Table 2 shows the results for the corrected resampled *t*-test. The number of runs used in resampling was varied from 10 to 100 to see the effect on the replicability.

The replicability increases with the number of runs almost everywhere. The only exception is in the last row, where the "almost consistent" value decreases by one when increasing the runs from 10 to 20. This can be explained by random fluctuations due to the random partitioning of the datasets. Overall, the replicability becomes reasonably acceptable when the number of runs is 100. In this case 80% of the results are "almost consistent", and the value of the replicability measure R is approximately 0.9 or above.

4.3 Results for tests based on (repeated) cross validation

For the standard *t*-test based on a single run of 10-fold cross validation we observed consistent results for 15, 16, and 14 datasets, comparing NB with C4.5, NB with NN, and C4.5 with NN respectively. Contrasting this with corrected resampling with 10 runs, which takes the same computational effort, we see that 10-fold cross validation

	#Runs			
	10	20	50	100
NB vs C4.5				
consistent	15	14	19	21
almost consistent	16	19	22	23
replicability (R)	0.801	0.843	0.892	0.922
NB vs NN				
consistent	12	15	18	20
almost consistent	20	21	22	23
replicability (R)	0.835	0.865	0.882	0.899
C4.5 vs NN				
consistent	12	14	18	23
almost consistent	18	17	22	24
replicability (R)	0.819	0.825	0.878	0.935

Table 2. Replicability for corrected resampled *t*-test.

is at least as consistent. However, it is substantially less consistent than (corrected) resampling at 100 runs. Note also that this test has an inflated Type I error [5].

Performing the same experiment in conjunction with the standard *t*-test based on the 100 differences obtained by 10-times 10-fold cross validation, produced consistent results for 25, 24, and 18 datasets, based on NB with C4.5, NB with NN, and C4.5 with NN respectively. This looks impressive compared to any of the tests we have evaluated so far. However, the Type I error of this test is very high (because of the overlapping training and test sets) and therefore it should not be used in practice.

To reduce Type I error it is necessary to correct the variance. Table 3 shows the same results for the corrected paired *t*-test based on the paired outcomes of r-times 10-fold cross validation. Comparing this to Table 2 (for corrected resampling) the consistency is almost everywhere as good and often better (assuming the same computational effort in both cases): the column with 1 run in Table 3 should be compared with the 10 runs column in Table 2, the column with 2 runs in Table 3 with the column with 20 runs in Table 2, etc. The same can be said about the replicability measure R. This indicates that repeated cross validation helps to improve replicability (compared to just performing random subsampling).

To ensure that the improved replicability of cross-validation is not due to stratification (which is not performed in the case of random subsampling), we performed an experiment where resampling was done with stratification. The replicability scores differed only very slightly from the ones shown in Table 2, suggesting the improved replicability is not due to stratification.

Because the corrected paired *t*-test based on 10-times 10-fold cross validation exhibits the best replicability scores, we performed an experiment to see how sensitive its replicability is to the significance level. The results, shown in Table 4, demonstrate that the significance level does not have a major impact on consistency or the replicability measure R. Note that the latter is greater than 0.9 in every single case, indicating very good replicability for this test.

	#Runs			
	1	2	5	10
NB vs C4.5				
consistent	16	20	21	24
almost consistent	18	21	23	25
replicability (R)	0.821	0.889	0.928	0.962
NB vs NN				
consistent	18	20	23	23
almost consistent	19	21	23	24
replicability (R)	0.858	0.890	0.939	0.942
C4.5 vs NN				
consistent	13	19	22	22
almost consistent	18	23	24	24
replicability (R)	0.814	0.904	0.928	0.928

Table 3. Replicability for corrected rx10 fold cross-validation test.

	Significance level			
	1%	2.5%	5%	10%
NB vs C4.5				
consistent	22	23	24	21
almost consistent	23	24	25	22
replicability (R)	0.927	0.936	0.962	0.915
NB vs NN				
consistent	23	24	23	23
almost consistent	24	27	24	23
replicability (R)	0.939	0.978	0.942	0.939
C4.5 vs NN				
consistent	23	24	22	20
almost consistent	23	24	24	24
replicability (R)	0.943	0.953	0.928	0.919

Table 4. Replicability of corrected 10x10 fold cross-validation test for various significance levels.

4.4 Simulation experiment

To study the effect of the observed difference in accuracy on replicability, we performed a simulation study. Four data sources were selected by randomly generating Bayesian networks over 10 binary variables where the class variable had 0.5 probability of being zero or one. A 0.5 probability of the class variable is known to cause the largest variability due to selection of the training data [5]. The first network had no arrows and all variables except the class variables were independently selected with various different probabilities. This guarantees that any learning scheme will have 50% expected accuracy on the test data. The other three data sources had a BAN structure [8], generated by starting with a naive Bayes model and adding arrows while guaranteeing acyclicity.

Using stochastic simulation [9], a collection of 1000 training sets with 300 instances each was created. Naive Bayes and C4.5 were trained on each of them and their accuracy measured on a test set of 20,000 cases, generated from each of the data sources. The average difference in accuracy is shown in Table 5 in the row marked Δ accuracy, and it ranges from 0% to 11.27%.

Each of the tests was run 10 times on each of the 4×1000 training sets. Table 5 shows, for each of the tests and each data source, the percentage of training sets for which the test is consistent (i.e., indicates the same outcome 10 times). The last column shows the minimum of the consistency over the four data sources.

Again, 5×2 cross validation, 10 times resampling, and 10 fold cross validation show rather low consistency. Replicability increases dramatically with 100 times resampling, and increases even further when performing 10 times repeated 10 fold cross validation. This is consistent with the results observed on the UCI datasets.

Table 5 shows that the tests have fewer problems with data sources 1 and 4 (apart from the 5×2 cv test), where it is easy to decide whether the two schemes differ. The 5×2 test has problems with data source 4 because it is a rather conservative test (low Type I error, high Type II error) and tends to err on the side of being too cautious when deciding whether two schemes differ.

Source	1	2	3	4	
Δ accuracy	0.0	2.77	5.83	11.27	min.
5x2 cv	72.3	71.2	63.5	16.9	16.9
10 x resampling	65.5	44.0	26.0	48.8	26.0
100 x resampling	90.9	73.2	66.8	97.2	66.8
10-fold cv	49.7	47.6	33.2	90.8	33.2
corrected 10x10 fold cv	91.9	80.3	76.7	98.9	76.7

Table 5. Results for data sources 1 to 4: the difference in accuracy between naive Bayes and C4.5 (in percent) and the consistency of the tests (in percent).

5 Conclusions

We considered tests for choosing between two learning algorithms for classification tasks. We argued that such a test should not only have an appropriate Type I error and low Type II error, but also high replicability. High replicability facilitates reproducing published results and reduces the likelihood of oversearching. In our experiments, good replicability was obtained using 100 runs of random subsampling in conjunction with Nadeau and Bengio's corrected resampled *t*-test, and replicability improved even further by using 10-times 10-fold cross-validation instead of random subsampling. Both methods are acceptable but for best replicability we recommend the latter one.

Acknowledgments

Eibe Frank was supported by Marsden Grant 01-UOW-019.

References

1. I.H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, 2000.
2. R.R. Bouckaert. Choosing between two learning algorithms based on calibrated tests. Proc 20th Int Conf on Machine Learning. Morgan Kaufmann, 2003.
3. R.R. Bouckaert. Choosing learning algorithms using sign tests with high replicability. Proc 16th Australian Joint Conference on Artificial Intelligence. Springer-Verlag, 2003.
4. C.L. Blake and C.J. Merz. UCI Repository of machine learning databases. Irvine, CA, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html
5. T. G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation, 10(7) 1895–1924, 1998
6. C. Nadeau and Y. Bengio. Inference for the generalization error. In Machine Learning 52:239–281, 2003
7. R. Quinlan. C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
8. N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian Network Classifiers. In Machine Learning 29:131–163, 1997
9. J. Pearl: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, 1988.