Available online at www.sciencedirect.com

ScienceDirect

Engineering Applications of Artificial Intelligence 20 (2007) 875–885

Engineering Applications of
ARTIFICIAL
INTELLIGENCE

www.elsevier.com/locate/engappai

A lot of randomness is hiding in accuracy

Arie Ben-David*

Management Information Systems, Department of Technology Management, Holon Institute of Technology, 52 Golomb St. P.O. Box 305, Holon 58102, Israel

Received 3 November 2006; received in revised form 5 January 2007; accepted 10 January 2007
Available online 23 February 2007

Abstract

The proportion of successful hits, usually referred to as "accuracy", is by far the most dominant meter for measuring classifiers' accuracy. This is despite of the fact that accuracy does not compensate for hits that can be attributed to mere chance. Is it a meaningful flaw in the context of machine learning? Are we using the wrong meter for decades? The results of this study do suggest that the answers to these questions are positive.

Cohen's kappa, a meter that does compensate for random hits, was compared with accuracy, using a benchmark of fifteen datasets and five well-known classifiers. It turned out that the average probability of a hit being the result of mere chance exceeded one third (!). It was also found that the proportion of random hits varied with different classifiers that were applied even to a single dataset. Consequently, the rankings of classifiers' accuracy, with and without compensation for random hits, differed from each other in eight out of the fifteen datasets. Therefore, accuracy may well fail in its main task, namely to properly measure the accuracy-wise merits of the classifiers themselves.

Keywords: Accuracy; Kappa; "Chance driven" hits; Machine learning; AUC

1. Introduction

Accuracy measures the number of successful hits relative to the total number of classifications. It is by far the most commonly used metric for assessing the accuracy of classifiers for years (Lim et al., 2000; Alpaydin, 2004; Witten and Frank, 2005; Demsar, 2006).

This research deals with a very serious anomaly of the accuracy. Here is a simple example: Table 1 shows a binary confusion matrix with 1000 classifications.

The accuracy in the confusion matrix of Table 1 is 0.5; Fifty percent of the classifications were correct. But what can be said about the classifier that produced these predictions? One can hardly think of a worse classifier. This is due to the fact that a randomly tossed fair coin will produce approximately similar results. In other words, all the classifier's predictions of Table 1 may be due to mere chance. A good accuracy meter should explicitly measure the added value, if any, of a classifier relative to a random, or a majority-based, outcome. In this respect, the classifier that produced the confusion matrix of Table 1 has no added value at all. Saying that the accuracy is 50%, though arithmetically correct, does not explicitly convey this meaning. Similar examples can be given for any multi-class case.

The machine-learning community has long been aware of the fact that accuracy is far from being a perfect meter. Usually, several classifiers are competing against each other. Baseline classifiers (typically, majority based) are often used too. There would have been nothing wrong with this method provided that the effect of random hits was similar across all classifiers for any given dataset. However, this hidden assumption was never put to a real test. Consider the following hypothetical example: Classifiers A and B are applied to a single dataset. Classifier A scores on the average 80% success rate, and classifier B (which can be a baseline) only 70%. Assume further that a proper statistical test on accuracy has concluded that A is more accurate than B. This conclusion, however, would not

*Tel.: +972 3 7317977; fax: +972 3 5716481.
E-mail address: hol_abendav@bezeqint.net.

0952-1976/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2007.01.001

Table 1
A simple confusion matrix

Correct class	Predicted class		
	Good	Bad	Total
Good	250	250	500
Bad	250	250	500
Total	500	500	1000

make much intuitive sense, should one also knew that 50% of A's successes may be due to mere chance, and only 10% of B's. This research clearly shows that such scenarios are possible, because chance differently affects various classifiers, even when they are applied to a similar dataset. Classifiers' accuracy should be compared after compensating for random hits, and this compensation may vary with each classifier, even when a single dataset is used. By ignoring the effects of random hits, one unavoidably risks arriving at the wrong conclusions.

An alternative to accuracy, a meter that does compensate for random hits, is known for decades. It is called Cohen's kappa (Cohen, 1960). Cohen's kappa is routinely used in disciplines such as Statistics, Psychology, Biology and Medicine for a long period. However, for one reason or another, it has received only very little attention in machine-learning circles.

This research was focused at answering the following two questions:

A. Is the problem of counting random hits meaningful in the context of machine leaning?
B. Are rankings according to accuracy always identical to those that are arrived at using Cohen's kappa? In other words, can we arrive at different conclusions about classifiers accuracy when chance considerations are taken into account?

To answer these questions, an empirical study was conducted. Fifteen datasets were tested using five well-known classifiers. The results are quite interesting:

A. On the average, more than one third of the hits in the benchmark could be attributed to chance alone. Accuracy ignores this high proportion altogether.
B. The rankings by accuracy and via Cohen's kappa differed from each other in eight out of the fifteen datasets. Different rankings may lead to different conclusions.

The findings of this research strongly suggest that we, the machine-learning community, are traditionally using the wrong meter, namely accuracy. We do that without being fully aware of the fact that a significant portion of the so-called "accuracy" is merely the product of chance. In this respect, Cohen's kappa is a more accurate meter for measuring classifiers' own merits than accuracy.

2. Cohen's kappa and its very rare use in machine learning

Cohen's kappa (Cohen, 1960) was first introduced as a measure of agreement between observers of psychological behavior. The original intent of Cohen's kappa was to measure the degree of agreement, or disagreement, between two people observing the same phenomenon. Cohen's kappa can be adapted to machine learning, as shown in the example of Table 2.

The accuracy shown in Table 2 is 97% ((70 + 900)/1000). Can all these 97% be attributed to the sophistication of the classifier alone? Does chance have anything to do with it?

Cohen's kappa is defined as

$$K = \frac{p_0 - p_c}{1 - P_c}, \tag{1}$$

where P_0 is the total agreement probability, or accuracy, and P_c is the "agreement" probability which is due to chance.

For the data of Table 2 kappa is computed as follows:

$$P_0 = \frac{70}{1000} + \frac{900}{1000} = 0.97 \quad \text{(i.e., accuracy)},$$

$$P_c = \frac{80}{1000} \times \frac{90}{1000} + \frac{920}{1000} \times \frac{910}{1000} = 0.84$$

and the value of kappa is thus

$$K = \frac{0.97 - 0.84}{1 - 0.84} = 0.81.$$

According to the kappa statistic, the classifier that produced the confusion matrix of Table 2 has a less impressive "accuracy": 0.81 and not 0.97.

What the kappa statistic expresses can be explained in a nutshell as follows: kappa evaluates the portion of hits that can be attributed to the classifier itself (i.e., not to mere chance), relative to all the classifications that cannot be attributed to chance alone.

What about a case of a perfect agreement?

In this case, shown in Table 3, α, β are integers and C_1 and C_2 are class values.

$$p_0 = \frac{\alpha}{\alpha + \beta} + \frac{\beta}{\alpha + \beta} = 1,$$

$$p_c = 2\left(\frac{\alpha}{\alpha + \beta}\right)^2 \geqslant 0$$

Table 2
Another confusion matrix

Correct class	Predicted class		
	Good	Bad	Total
Good	70	10	80
Bad	20	900	920
Total	90	910	1000

Table 3
Prefect agreement confusion matrix

Correct class	Predicted class		
	C_1	C_2	Total
C_1	α	0	α
C_2	0	β	β
Total	α	β	$\alpha+\beta$

when $\alpha=\beta$

$$K=\frac{1-P_c}{1-P_c}=1.$$

Should a classifier and reality be in a perfect disagreement with each other, a confusion matrix similar to the one shown in Table 4 is expected:

Table 4
Prefect disagreement confusion matrix

Correct class	Predicted class		
	C_1	C_2	Total
C_1	0	α	α
C_2	β	0	β
Total	β	α	$\alpha+\beta$

Similar to the case of Table 3,

$$p_0=0,$$

$$P_c=\frac{2\alpha\beta}{(\alpha+\beta)^2}.$$

Therefore

$$K=\frac{-2\alpha\beta}{\alpha^2+\beta^2}$$

and when $\alpha=\beta$

$$K=-1.$$

Finally, in a random binary "classification" case, such as the one of Table 1, the confusion matrix of Table 5 is expected

Table 5
A random confusion matrix

Correct class	Predicted class		
	C_1	C_2	Total
C_1	α	α	2α
C_2	α	α	2α
Total	2α	2α	4α

$$p_0=2\left(\frac{\alpha}{4\alpha}\right)=0.5,$$

$$P_c=2\left(\frac{2\alpha}{4\alpha}\right)^2=0.5.$$

Therefore

$$K=\frac{0.5-0.5}{1-0.5}=0.$$

Cohen's kappa statistic, thus, ranges from −1 (total disagreement) through 0 (random classification) to 1 (perfect agreement). It can be shown that the above results do hold in any multi-class case; not only in binary-class problems.

The theoretical range of the kappa statistic (i.e., from −1 to +1) is different from that of accuracy (0–1). This may seemingly cause difficulties while trying to compare the two meters. Fortunately, most classifiers (at least those which are considered "reasonable" in machine-learning circles) do at least as good as random or as majority-based classifiers on most real-world datasets, so by definition they score kappa higher than zero. This observation, that was first made by Margineantu and Dietterich (1997), and is re-confirmed by the results this experiment, makes the actual comparison of both meters much easier than initially expected.

Sometimes it is more convenient to calculate kappa in terms of cell counts rather than probabilities.

$$K=\frac{N\,\Sigma_{i=1}^{I}x_{ii}-\Sigma_{i=1}^{I}x_{i.}x_{.i}}{N^2-\Sigma_{i=1}^{I}x_{i.}x_{.i}},\qquad(2)$$

where x_{ii} is the cell count in the main diagonal, N is the number of examples, I is the number of class values, and $x_{.i}$, $x_{i.}$ are the columns and rows total counts, respectively. Formula (2) is given in terms of counts, but by dividing both the nominator and the denominator of (2) by N^2 one gets probabilities.

The expected probability of a hit being the result of chance is

$$P_c=\frac{1}{N^2}\sum_{i=1}^{I}x_{i.}x_{.i},\qquad(3)$$

where N, I, $x_{.i}$, $x_{i.}$ are as in (2).

The kappa statistic has mainly been used in Social Sciences, Biology, Statistics, and in Medical Sciences for a couple of decades. However, within the context of machine learning, it has not received much attention. One of the few exceptions can be found in (Margineantu and Dietterich, 1997), where Cohen's kappa was used for pruning quite successfully. It was not used there, however, as a meter for measuring the resulting classifiers' accuracy.

The WEKA Machine Learning Project (WEKA) is another exception. Cohen's kappa statistic is calculated as one of the measures of classifiers' accuracy in WEKA for a couple of years by now. Yet, few, if any, scientific publications in machine learning do make any use of it as an accuracy meter for classifiers.

There are many possible sources for using the kappa statistic without bothering with tedious calculations. WEKA is clearly one option, but statistical packages such as SPSS and SAS can also be used. Other statistical packages, such as MedCalc (MedCalc), used mainly by the medical research community, also include kappa calculations. There are also many free Internet downloads, such as the version written by P. Bonnardel (Bonnardel), that can be used as a stand-alone application.

Kappa has some interesting properties that will not be discussed here in detail. For instance:

1. Majority-based classifiers that are traditionally used as baselines in machine-learning literature, always score zero kappa.
2. All random classifiers (i.e., those that randomly classify according to the class distribution) also score zero kappa.
3. Note that according to the kappa statistic, there is no added value in neither a majority-based nor in a random classifier, so kappa essentially estimates how (hopefully) better a classifier does relative to these two.
4. The value of P_c, the hit probability that is due to chance, can be computed for any class distribution, symmetric or skewed, by formula (3).
5. Complex loss functions can be incorporated into kappa, making its modified version, known as "weighted kappa", a good candidate meter for cost-sensitive applications.

Similar to any other statistic, kappa it has its own limitations. Many concerns were raised over the years and some remedies were suggested. The interested reader is referred to publications such as Maclure and Willett (1987), Thompson and Walter (1988), Cook (1998) and Cicchetti and Feinstein (1990) for more details.

3. Related accuracy meters

Many attempts to use meters other than accuracy for measuring classifiers' performance were made. Perhaps the most famous is Quinlan's use of entropy in ID3 (Quinlan, 1987). The concept of entropy was borrowed from communications. Entropy basically measures uncertainty. In a certain universe its value is zero. Later on, an extension, named Information Gain (Quinlan, 1993) was introduced. Both entropy and Information Gain were used for guiding a greedy search for the most explanatory attributes (i.e., those that minimize the entropy). Both were quite successful. However, entropy and Information Gain do not compensate for random hits. In this respect they are similar to accuracy. Despite of the fact that entropy and some entropy-based improvements thereafter proved valuable concepts for building decision trees, most publications did evaluate their resulting trees in terms of accuracy.

Compensating for random hits at the rule level was an integral part of Clark and Nibeltt's (1989) CN2. CN2 uses a likelihood ratio statistic that measures the difference between class probability distribution in the set of examples that are covered by a rule, versus that of the entire set of examples. A rule is considered significant if it locates regularity that is not due to chance. Rules that were selected this way were found to perform quite accurately. However, the indication for the usefulness of CN2 came by comparing the end concepts' accuracy; not the likelihood ratios which were used for rule selection.

The weighted relative accuracy (WRA), a concept introduced by Lavrac et al. (1999), is a single metric that trades off generality and relative accuracy. WRA combines precision and recall, two concepts that are widely used in information retrieval. It has been shown in Todorovski et al. (2000) that WRA generates more compact rule bases compared with those of CN2, on the expanse of slightly lower accuracy. Since only accuracy was used for classifier ranking, it is unclear if the more compact rule bases that were generated by WRA resulted in more or less random hits than those that were generated by CN2. This question was never raised nor tested.

All the above-mentioned meters were shown to be effective for attribute or rule selection. However, they were never universally accepted as measures for accuracy, even for their own generated concepts; accuracy has typically been used instead.

Provost and Fawcett (1997) brought the idea of using receiver operating characteristics (ROC) curves form the area of communications to machine learning. ROC curves are a very useful visualization tools for analyzing tradeoffs between true positives and false positives. The area under curve (AUC) is often used as a measure of accuracy. AUCs can be used to estimate the added value (if any) of a classifier relative to a random one, by subtracting the AUC of the latter from the AUC of the first.

ROC curves and AUCs are particularly useful in binary-class problems. They were rarely used in three-class problems too. However, using them in multi-class problems (e.g., problems with more than three classes), such as in most of the datasets that are used in this experiment, though technically possible by making (potentially many) binary splits, may be very cumbersome and not too practical; In particular, when many classes are involved. So far, the usefulness of ROC Curves and AUCs has not been demonstrated on real-world problems with more than three class values.

Clearly, being a scalar, Cohen's kappa is less expressive than ROC curves when applied to binary-class cases. However, for multi-class problems, kappa is a very useful, yet simple, meter for measuring classifier's accuracy while compensating for random successes.

4. The experiment

Fifteen classification datasets were used in this experiment. They are shown in Table 6.

Eight datasets were taken from the UCI machine-learning repository (UCI). One thousand examples were randomly selected from the original Nursery dataset to reduce learning and classification time. Four datasets were taken from the WEKA web site. These datasets are well documented, and will not be discussed here any further. EFE is a dataset that was collected in order to study which attributes of matriculation examination questioners mostly affect their quality. English Comprehension dataset is a part of an ongoing research project currently conducted in Holon Institute of Technology. This research aims at improving English comprehension of students. The Project Management dataset was collected in an attempt to identify the most influential factors in the success or failure of R&D projects in Israel. Fourteen of the datasets shown in Table 6 cover a wide range of real-world classification problems. Only one, Monks-3, is an artificial dataset.

Five well-known classifiers were selected: C4.5 (Quinlan, 1993), SMO (Platt, 1998), Naïve Bayes (Domingos and Pazzani, 1997), Logistic Regression, and Random Forest (Breiman, 2001). They represent different approaches to the building of classifiers. They all have good reputation as being relatively accurate. The experiment was done using WEKA 3.5.3. More references about these classifiers and details about their WEKA implementation can be found in (Witten and Frank, 2005). All the results that are reported here are based on stratified ten-fold cross validations, and the default parameter values of the respective classifiers.

5. Major findings

Table 7 shows the main results for each dataset. These results are averages of all the five classifiers.

The average accuracy, Cohen's kappa statistic, and P_c, the seemingly "agreement" probability that can really be attributed to chance alone, are shown at the respective columns. The latter is labeled "chance". To their right are the average half widths of two-sided, t distribution, 95% confidence intervals. The bottom row shows a simple average of each column. Every dataset, regardless of its

Table 6
Main characteristics of the datasets

	Name	Source	Size	No. of attributes	No. of classes
1	Balance	UCI	625	4	5
2	Car	UCI	1728	6	4
3	Contraceptive	UCI	1473	9	3
4	Credit	UCI	690	15	2
5	EFE	Israel ministry of education	124	8	4
6	Engish_Comp	HIT—Project	91	13	6
7	ERA	WEKA	1000	4	9
8	ESL	WEKA	488	4	9
9	Housing	UCI	506	12	4
10	LEV	WEKA	1000	4	5
11	Monks-3	UCI	432	6	2
12	Nursery	UCI	1000	8	5
13	Post Operative	UCI	90	8	3
14	Proj. Man.	Beer Sheva U	89	10	7
15	SWD	WEKA	1000	10	4

Table 7
Average results for each dataset

	Dataset	Accuracy	95% CI	Kappa	95% CI	Chance	95% CI
1	Balance	0.8419	0.0211	0.7078	0.0383	0.4189	0.0338
2	Car	0.9148	0.0151	0.8111	0.0336	0.5447	0.0065
3	Contraceptive	0.4566	0.0229	0.1544	0.0350	0.3571	0.0056
4	Credit	0.8484	0.0324	0.6948	0.0646	0.5037	0.0033
5	EFE	0.5813	0.0788	0.2020	0.1490	0.4726	0.0437
6	Engish_Comp	0.2069	0.0951	−0.0014	0.1222	0.2077	0.0208
7	ERA	0.2676	0.0329	0.1442	0.0381	0.1442	0.0033
8	ESL	0.6569	0.0409	0.5665	0.0528	0.2080	0.0057
9	Housing	0.7057	0.0414	0.5393	0.0622	0.3621	0.0149
10	LEV	0.6096	0.0343	0.4418	0.0502	0.3001	0.0077
11	Monks-3	0.9944	0.0041	0.9889	0.0081	0.0699	0.0345
12	Nursery	0.9026	0.0133	0.8570	0.0197	0.3181	0.0028
13	Post Operative	0.6733	0.0571	−0.0186	0.1040	0.6788	0.0411
14	Proj. Man.	0.3192	0.0924	0.0641	0.1176	0.2708	0.0423
15	SWD	0.5708	0.0381	0.3490	0.0553	0.3409	0.0078
	Average	0.6367	0.0413	0.4334	0.0634	0.3465	0.0182

size, has an equal weight in that average. Again, all the results that are shown in Table 7 are averages over all the five classifiers mentioned above.

The average accuracy was about 64%. The average kappa was about 43%. On the average, kappa had a higher variance than accuracy. This fact is reflected by its wider average 95% confidence interval. Most surprisingly, and perhaps the most important finding shown in Table 7, is the fact that on the average, about 35% of all the hits (34.65% to be exact) could be attributed to chance alone. In other words, on the average, more than one third of the hits could not be attributed to the classifiers' sophistication. This finding raises the question whether accuracy, that totally ignores this phenomenon, is a trustworthy meter.

Up to this point, question A. of Section 1 was dealt with. Let us turn our attention now to the second question, namely, can the use of kappa affect model rankings relative to those that are obtained by accuracy?

Formula (1) suggests that some positive correlation does exist between accuracy and kappa. But to what degree can changes in accuracy explain changes in kappa? To answer this question, a Linear Regression was carried out. kappa was the dependent variable, and accuracy was the independent one. There have been 150 such couples in this run, ten for each dataset, generated by the ten-fold cross validation trails. The main results are shown in Table 8.

Table 8 shows that, on the average, variations in accuracy can explain about 78% of those in the kappa statistic. Although a positive correlation between accuracy and the kappa statistic was found (as expected), on the average, 22% of the latter's variations remained unexplained. Consequently, it has been hypothesized that accuracy and kappa-based rankings can sometimes differ from each other. This hypothesis was tested here as well.

Appendix A shows the results of the stratified ten-fold cross validation for each classifier. The tables of Appendix A are similar in their structure to those of Table 7, so no further explanations about them will be given. Based on the results that are shown in Appendix A, the classifiers were ranked by accuracy as well as by kappa. It turned out that in eight out of the fifteen datasets (53%) the rankings according to the kappa statistic differed from the rankings that were based on accuracy. The rankings for these eight datasets are shown in Table 9, where 1 stands for "the most accurate", 2 for the "second most accurate", etc. Identical rankings of datasets via the two meters are not shown in Table 9.

As hypothesized, kappa is likely to result in different accuracy-wise rankings than accuracy; in more than one half of the datasets the rankings did change. Sometimes, the best ranking classifier according to accuracy was ranked also the best via kappa (i.e., the ranking of the remaining classifiers changed). In other cases, such as in the Housing and LEV datasets, the best ranking classifier via accuracy ranked only the second according to kappa. In Post Operative and

Table 8
Regression results

	Model	R^2	p-value
1	C4.5	0.7467	<0.00001
2	SMO	0.7654	<0.00001
3	Naïve Bayes	0.7687	<0.00001
4	Logistic	0.8180	<0.00001
5	Random forest	0.7924	<0.00001
	Average	0.7782	<0.00001

Table 9
Different rankings by accuracy and kappa

	Dataset	Model	Ranking by accuracy	Ranking by kappa
3	Contraceptive	Logistic	1	2
3	Contraceptive	N Bayes	2	1
3	Contraceptive	SMO	3	3
3	Contraceptive	C4.5	4	5
3	Contraceptive	Random forest	5	4
5	EFE	N Bayes	1	1
5	EFE	Random forest	2	2
5	EFE	C4.5	3	4
5	EFE	SMO	4	5
5	EFE	Logistic	5	3
6	Engish_Comp	C4.5	1	1
6	Engish_Comp	Random forest	2	3
6	Engish_Comp	Logistic	3	2
6	Engish_Comp	SMO	4	4
6	Engish_Comp	N Bayes	5	5
8	ESL	Logistic	1	1
8	ESL	SMO	2	2
8	ESL	N Bayes	3	4
8	ESL	C4.5	4	3
8	ESL	Random forest	5	5
9	Housing	SMO	1	2
9	Housing	Random forest	2	3
9	Housing	Logistic	3	1
9	Housing	C4.5	4	4
9	Housing	N Bayes	5	5
10	LEV	Random forest	1	2
10	LEV	SMO	2	1
10	LEV	Logistic	3	3
10	LEV	C4.5	4	4
10	LEV	N Bayes	5	5
13	Post Operative	C4.5	1	3
13	Post Operative	N Bayes	2	2
13	Post Operative	SMO	3	5
13	Post Operative	Logistic	4	1
13	Post Operative	Random forest	5	4
14	Proj. Man.	N Bayes	1	3
14	Proj. Man.	SMO	2	2
14	Proj. Man.	Logistic	3	1
14	Proj. Man.	Random forest	4	4
14	Proj. Man.	C4.5	5	5

Table 10
Results of the Post Operative dataset

	Dataset	Model	Accuracy	Ramk	Kappa	Rank	Chance
13	Post Operative	C4.5	0.7000	1	−0.0174	3	0.7049
13	Post Operative	N Bayes	0.6889	2	−0.0109	2	0.6901
13	Post Operative	SMO	0.6778	3	−0.0509	5	0.6938
13	Post Operative	Logistic	0.6556	4	0.0157	1	0.6481
13	Post Operative	Random Forest	0.6444	5	−0.0295	4	0.6568

Table 11
Highest and lowest chance of random hits

	Dataset	Lowest chance	Highest chance	Chance: relative difference
11	Monks-3	The other four models	N Bayes	Very high
1	Balance	Logistic	N Bayes	80.9%
14	Proj. Man.	Logistic	Random forest	28.3%
5	EFE	Logistic	Random forest	14.7%
6	Engish_Comp	SMO	C4.5	10.7%
7	ERA	N Bayes	C4.5	8.9%
13	Post Operative	Logistic	C4.5	8.8%
3	Contraceptive	N Bayes	C4.5	8.6%
9	Housing	N Bayes	SMO	6.7%
15	SWD	N Bayes	C4.5	6.0%
8	ESL	Logistic	N Bayes	5.7%
2	Car	SMO	N Bayes	5.3%
10	LEV	SMO	N Bayes	4.7%
12	Nursery	Logistic	N Bayes	2.5%
4	Credit	SMO	N Bayes	2.0%

Project Management datasets, the best ranking classifier according to accuracy ranked only third via kappa.

We turn our attention now to the values of P_c. Table 10 is also based on the data of Appendix A. It shows, as an example, the values of P_c for each classifier when applied to the Post Operative dataset, sorted in decreasing order of accuracy. In this case, C4.5 was the classifier with the highest P_c (0.7049) and Logistic with the lowest (0.6481). The difference between these two values relative to the lowest was 8.8% $\left(100\frac{0.7049-0.6481}{0.6481}\right)$. We later refer to this value as the "relative difference" in P_c values.

The relative difference in P_c values, as shown for the data of Table 10, was computed for each dataset in the benchmark, and the main results are shown in Table 11. Table 11 shows the name of the model that resulted in the lowest and the highest P_c, and the relative difference between them as explained above. Table 11 is arranged in decreasing order of relative differences of P_c for the entire benchmark.

The first row of Table 11 indicates a "very high" relative difference between Naïve Bayes and the rest of the models. This is due to the fact that the value of P_c was 34.96% for Naïve Bayes versus zero for the other classifiers. Recalling that Monks-3 was the only synthetic dataset in the benchmark, one can consider it an exception. However, it is evident from the rest of the fourteen datasets that:

A. In some datasets, such as Monk-3, Balance, and Project Management—the odds of a hit being the result of mere chance varied quite significantly with the classifier that was tested. In other datasets, those at the bottom of the Table 11, these odds were quite similar to each other.
B. Logistic, Naïve Bayes and SMO appear in the column "lowest chance" in Table 11 in approximately similar frequencies. Random Forests and C4.5, on the other hand, are absent from that column. They do appear, however, in the column "highest chance", in which Naïve Bayes dominates. While it is not possible to draw any definite conclusion from this benchmark alone, it seems that some classifiers are more likely to generate random hits than others. A comprehensive investigation of this topic, however, per classifier, is outside the scope of this research.

6. Discussion

The results shown above demonstrate an important property: While taking chance into consideration via the kappa statistic, classifier rankings may differ from those that rely on accuracy. In eight out of the fifteen datasets, the rankings by accuracy and by kappa were not identical. Different rankings may result in different statistical conclusions.

Table 12 shows three pair-wise rankings that are based on two-sided t-test at $\alpha = 0.05$. The first line of Table 12 compares the accuracy of Naïve Bayes (Model A) with that of C4.5 (Model B) on the Contraceptive dataset. According

Table 12
Different statistics yield different conclusions

Model A	Model B	Dataset	Ranking by accuracy	Ranking by kappa
Naïve Bayes	C4.5	Contraceptive	No signif. difference	A$\gg$B
Naïve Bayes	Random Forest	Contraceptive	No signif. difference	A$\gg$B
SMO	Naïve Bayes	LEV	No signif. difference	A$\gg$B

to accuracy—both classifiers were statistically indistinguishable at $\alpha = 0.05$. However, when the kappa statistic was used, the null hypothesis was rejected, so one can conclude that according to this particular statistical test, Naïve Bayes was found more accurate than C4.5 at $\alpha = 0.05$ on this dataset (A$\gg$B in the rightmost column of Table 12 indicates that classifier A is more accurate than B). The data that leads to this conclusion is shown on line 3 of Tables A.1 and A.3 in Appendix A. The remaining two lines of Table 12 are to be read similarly.

Table 12 shows three examples of statistical conclusions about model rankings that were meter dependent. The interested reader is invited to brows through Appendix A. in search of more such examples. It is possible that different statistical methods, if used, would have lead to different conclusions, of course. However, as has been demonstrated here, any selected statistical method may yield different conclusions should it rely upon accuracy or on the kappa statistic, and the results shown in Table 12 just demonstrate this fact.

Let us turn our attention now back to a point that was made in the Section 1, namely that the effect of chance on successful hits varies from one classifier to another, even when the classifiers are applied to a similar dataset. As can be seen in Table 10, when applied to the Post Operative dataset, C4.5 ranked first according to accuracy. However, it had also the highest value of P_c (0.7049), 8.8% higher than Logistics. For this reason, Logistic ranked first by kappa, and C4.5 only the third.

The Post Operative results shown in Table 10 demonstrate yet another small convenience of using the kappa statistic. Note that the kappa values of all the classifiers that were tested on this particular dataset were around zero. By definition, any majority-based classifier and any random classifier, if applied to any dataset, would have scored zero kappa. The actual computation of such classifiers as baselines (as many researchers do while using accuracy) for determining the added value (if any) of a classifier is not required here in the first place, since they will score zero kappa by definition. Since the kappa values of all the classifiers in Table 10 are close to zero—the immediate conclusion is that they all have failed to show any added value relative to a majority-based or a random classifier, should they were applied to this particular dataset.

On the other end of the spectrum one can point at the results of the Monks-3 dataset (Table 7, dataset number 11), where both accuracy and the kappa statistic indicate very good average classifiers' accuracy. This is an example of a case where the kappa statistic has a relatively small added value relative to accuracy. In Monks-3, accuracy was very high (above 99%), and only about 7% of the hits, on the average, could be regarded random. However, recalling that Monks-3 was the only artificial dataset in this experiment, one can rightfully argue that it does not represent real-world classification problems. As shown in Table 7, most of the datasets did result in a higher difference between the values of their accuracy and their kappa. Table 7 shows that, on the average, more than one third of the hits were actually "chance driven", so to speak. This is not an outcome of a magnitude that can be overlooked. Accuracy ignores this fact, while Cohen's kappa does not. The high proportion of "chance driven" hits is also reflected by a difference in the averages of Table 7: 0.6367 for accuracy, and only 0.4334 for kappa.

Classifier's properties such as bias, as well as characteristics of the datasets, affect P_c. This can clearly be seen from formula (3). Further investigations of these interesting topics were outside the scope of this work, and they are proposed for future research. Hopefully, such research will be able to identify characteristics of classifiers and datasets that tend to be less vulnerable to random agreements than others.

7. Conclusions

Two important findings were presented here for the first time:

A. In the context of machine learning, there are likely to be many potentially random hits hidden in the meter named "accuracy". More than one third successful classifications, on the average, were found in this experiment as hits that can be attributed to chance alone, and not to the merits of the classifiers that were tested. A result of such a magnitude cannot be overlooked in any scientific research or in any practical application.

B. The use of Cohen's kappa may influence classifiers' rankings relative to those that are based on accuracy. More than one half of the classifiers' rankings changed when the kappa statistic was applied, relative to the rankings that were based on accuracy.

There are also two practical lessons that should be learned from the results of this experiment:

1. When one uses accuracy in order to compare two or more classifiers (one of which may be a baseline classifier),

he/she must first verify that chance equally affects all the classifiers. This experiment shows that this is rarely the case.

2. If for whatever reason one does not use ROC curves or AUCs (for instance, in multi-class problems)—he/she should at least consider using kappa, since it compensates for random hits. Kappa makes intuitive sense, and it is very easy to compute and to interpret. As a small bonus—while using the kappa statistic, one does not resort to any majority-based or random classifier as baselines, for the simple reason that it is known that they score zero kappa by definition.

Some interesting questions were not included in this study. For example, what are the conditions, if any, under which only a small fraction of hits will be attributed to chance and vise versa? Characteristics of the datasets, as well as those of the classifiers, eventually play a role. Which one? How do they affect? What typifies problems with relatively high accuracy but low kappa? These and other interesting questions are left for future research.

It is not a simple matter to suddenly become suspicious of a meter one is accustomed to for decades. Hopefully, the findings of this research will convince the machine-learning community to adopt meters such as AUCs and Cohen's kappa, that take random successes into consideration, as a standard.

Appendix A. Detailed results

The results shown in this Appendix are of stratified ten-fold cross validations, done with WEKA 3.5.3. They are similar in structure to those of Table 7. The interested reader is referred to the explanations of that table in the text for further details (see Table A.1).

Table A.1
C4.5

	Dataset	Accuracy	95% CI	Kappa	95% CI	Chance	95% CI
1	Balance	0.6336	0.0361	0.3222	0.0645	0.4596	0.0047
2	Car	0.9236	0.0151	0.8345	0.0321	0.5390	0.0064
3	Contraceptive	0.4345	0.0256	0.1093	0.0418	0.3650	0.0068
4	Credit	0.8536	0.0347	0.7057	0.0693	0.5032	0.0042
5	EFE	0.5615	0.0660	0.1414	0.1439	0.4845	0.0363
6	Engish_Comp	0.2811	0.1205	0.0850	0.1472	0.2156	0.0235
7	ERA	0.2670	0.0363	0.1405	0.0429	0.1472	0.0031
8	ESL	0.6598	0.0335	0.5699	0.0444	0.2083	0.0083
9	Housing	0.6998	0.0321	0.5267	0.0495	0.3656	0.0188
10	LEV	0.6040	0.0451	0.4317	0.0655	0.3030	0.0097
11	Monks-3	1.0000	0.0000	1.0000	0.0000	0.0000	0.0000
12	Nursery	0.8930	0.0131	0.8428	0.0193	0.3190	0.0029
13	Post Operative	0.7000	0.0384	−0.0174	0.0393	0.7049	0.0376
14	Proj. Man.	0.2694	0.0547	−0.0315	0.0627	0.2917	0.0340
15	SWD	0.5650	0.0291	0.3336	0.0403	0.3476	0.0077
	Average	0.6231	0.0387	0.3996	0.0575	0.3503	0.0136

Table A.2
SMO

	Dataset	Accuracy	95% CI	Kappa	95% CI	Chance	95% CI
1	Balance	0.9056	0.0147	0.8255	0.0271	0.4590	0.0031
2	Car	0.9375	0.0121	0.8648	0.0263	0.5376	0.0074
3	Contraceptive	0.4664	0.0187	0.1610	0.0296	0.3640	0.0073
4	Credit	0.8551	0.0285	0.7115	0.0559	0.4981	0.0032
5	EFE	0.5545	0.0831	0.1335	0.1731	0.4807	0.0412
6	Engish_Comp	0.1733	0.0778	−0.0294	0.1096	0.1948	0.0209
7	ERA	0.2790	0.0328	0.1551	0.0379	0.1467	0.0025
8	ESL	0.6617	0.0449	0.5720	0.0575	0.2092	0.0050
9	Housing	0.7274	0.0324	0.5672	0.0480	0.3708	0.0141
10	LEV	0.6280	0.0305	0.4747	0.0430	0.2919	0.0059
11	Monks-3	1.0000	0.0000	1.0000	0.0000	0.0000	0.0000
12	Nursery	0.9150	0.0108	0.8755	0.0161	0.3169	0.0032
13	Post Operative	0.6778	0.0451	−0.0509	0.0587	0.6938	0.0374
14	Proj. Man.	0.3278	0.0847	0.1113	0.1047	0.2429	0.0371
15	SWD	0.5600	0.0494	0.3280	0.0733	0.3454	0.0093
	Average	0.6446	0.0377	0.4467	0.0574	0.3435	0.0132

Table A.3
Naïve bayes

	Dataset	Accuracy	95% CI	Kappa	95% CI	Chance	95% CI
1	Balance	0.9136	0.0098	0.8398	0.0180	0.4608	0.0018
2	Car	0.8553	0.0178	0.6661	0.0432	0.5662	0.0082
3	Contraceptive	0.4671	0.0215	0.1974	0.0303	0.3360	0.0057
4	Credit	0.8478	0.0306	0.6908	0.0617	0.5081	0.0018
5	EFE	0.6346	0.0640	0.3098	0.0953	0.4719	0.0506
6	Engish_Comp	0.1411	0.0961	−0.0857	0.1248	0.2081	0.0152
7	ERA	0.2530	0.0288	0.1363	0.0333	0.1351	0.0049
8	ESL	0.6599	0.0451	0.5667	0.0581	0.2146	0.0064
9	Housing	0.6521	0.0464	0.4670	0.0688	0.3474	0.0169
10	LEV	0.5660	0.0331	0.3748	0.0494	0.3055	0.0075
11	Monks-3	0.9721	0.0205	0.9443	0.0406	0.3496	0.1726
12	Nursery	0.8770	0.0165	0.8183	0.0247	0.3227	0.0017
13	Post Operative	0.6889	0.0503	−0.0109	0.1181	0.6901	0.0398
14	Proj. Man.	0.3569	0.1272	0.0966	0.1621	0.2889	0.0530
15	SWD	0.5740	0.0402	0.3667	0.0559	0.3280	0.0074
	Average	0.6306	0.0432	0.4252	0.0656	0.3689	0.0262

Table A.4
Logistic regression

	Dataset	Accuracy	95% CI	Kappa	95% CI	Chance	95% CI
1	Balance	0.9856	0.0100	0.9750	0.0174	0.2548	0.1569
2	Car	0.9312	0.0164	0.8502	0.0361	0.5398	0.0053
3	Contraceptive	0.4813	0.0262	0.1840	0.0390	0.3645	0.0044
4	Credit	0.8333	0.0290	0.6639	0.0580	0.5044	0.0032
5	EFE	0.5455	0.0764	0.1920	0.1479	0.4313	0.0472
6	Engish_Comp	0.2189	0.0973	0.0166	0.1239	0.2057	0.0176
7	ERA	0.2690	0.0341	0.1446	0.0386	0.1456	0.0034
8	ESL	0.6701	0.0442	0.5858	0.0567	0.2030	0.0034
9	Housing	0.7236	0.0539	0.5687	0.0798	0.3605	0.0136
10	LEV	0.6190	0.0314	0.4560	0.0462	0.2994	0.0074
11	Monks-3	1.0000	0.0000	1.0000	0.0000	0.0000	0.0000
12	Nursery	0.9220	0.0106	0.8861	0.0158	0.3148	0.0036
13	Post Operative	0.6556	0.0696	0.0157	0.1877	0.6481	0.0292
14	Proj. Man.	0.3250	0.0859	0.1170	0.1196	0.2324	0.0425
15	SWD	0.5700	0.0410	0.3482	0.0620	0.3402	0.0081
	Average	0.6500	0.0417	0.4669	0.0686	0.3230	0.0231

Table A.5
Random forest

	Dataset	Accuracy	95% CI	Kappa	95% CI	Chance	95% CI
1	Balance	0.7714	0.0347	0.5763	0.0648	0.4602	0.0023
2	Car	0.9265	0.0139	0.8400	0.0301	0.5406	0.0051
3	Contraceptive	0.4338	0.0223	0.1205	0.0342	0.3562	0.0037
4	Credit	0.8522	0.0393	0.7023	0.0779	0.5044	0.0039
5	EFE	0.6103	0.1047	0.2332	0.1848	0.4948	0.0431
6	Engish_Comp	0.2200	0.0839	0.0067	0.1055	0.2145	0.0267
7	ERA	0.2700	0.0325	0.1447	0.0377	0.1465	0.0024
8	ESL	0.6330	0.0366	0.5383	0.0471	0.2048	0.0052
9	Housing	0.7255	0.0419	0.5670	0.0646	0.3660	0.0113
10	LEV	0.6310	0.0315	0.4719	0.0471	0.3008	0.0078
11	Monks-3	1.0000	0.0000	1.0000	0.0000	0.0000	0.0000
12	Nursery	0.9060	0.0155	0.8623	0.0228	0.3171	0.0024
13	Post Operative	0.6444	0.0821	−0.0295	0.1162	0.6568	0.0616
14	Proj. Man.	0.3167	0.1093	0.0272	0.1388	0.2982	0.0448
15	SWD	0.5850	0.0308	0.3684	0.0448	0.3431	0.0063
	Average	0.6350	0.0453	0.4286	0.0678	0.3469	0.0151

References

Alpaydin, E., 2004. Introduction to Machine Learning. MIT Press, Cambridge, MA.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Cicchetti, D.V., Feinstein, A.R., 1990. High agreement but low kappa, in two parts. Journal of Clinical Epidemiology 43 (6), 543–558.

Clark, P., Niblett, T., 1989. The CN2 induction algorithm. Machine Learning 3 (4), 261–283.

Cohen, J.A., 1960. Coefficient of agreement for nominal scales. Educational and Psychological Measurement, 37–46.

Cook, R.J., 1998. Kappa and its dependence on marginal rates. In: Armitage, P., Colton, T. (Eds.), Encyclopedia of BioStatistics. Wiely, New York, pp. 2166–2168.

Demsar, Janez, 2006. Statistical comparisons of classifiers over multiple datasets. Journal of Machine Learning Research 7, 1–30.

Domingos, P., Pazzani, M., 1997. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103–130.

Lavrac, N., Flach, P., Zupan, B., 1999. Rule evaluation measures: a unifying view. In: Dzeroski, S., Flach, P. (Eds.), Ninth International Workshop on Inductive Logic Programming (ILP'99). Springer, Berlin, pp. 174–185.

Lim, T.S., Loh, W.Y., Shih, Y.S., 2000. A comparison of prediction accuracy, complexity and training time of thirty-three old and new classification algorithms. Machine Learning 40, 203–229.

Maclure, M., Willett, W.C., 1987. Misinterpretation and misuse of the kappa statistic. American Journal of Epidemiology 126 (2), 161–169.

Margineantu, D.D., Dietterich, T.G., 1997. Bootstrap methods for the cost-sensitive evaluation of classifiers. In: Kaufmann, M. (Ed.), Proceedings of the Seventh International Conference on Machine Learning, pp. 582–590.

Platt, J., 1998. Fast training of support vector machines using sequential minimal optimization. In: Schoelkopf, Burges, C., Smola, A. (Eds.), Advances in Kernel Methods Support Vector Learning. MIT Press, Cambridge, MA, pp. 185–208.

Provost, F., Fawcett, T., 1997. Analysis and visualization of classifier performance—comparison under imprecise class and cost distribution. In: Heckerman, D., Mannila, H., Pregibon, D., Uthurusamy, R. (Eds.), Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, AAAI Press, CA.

Quinlan, J.R., 1987. Simplifying decision trees. In: Gaines, B., Boose, J. (Eds.), Knowledge Acquisition for Knowledge-Based Systems. Academic Press, New York, pp. 239–252.

Quinlan, J.R., 1993. Programs for Machine Learning. Morgan Kaufmann, Los Altos, CA.

Thompson, W.D., Walter, S.D., 1988. A reappraisal of the kappa coefficient. Journal of Clinical Epidemiology 41, 949–958.

Todorovski, L., Llach, P., Lavrac, N., 2000. Predictive performance of weighted relative accuracy. In: Zighed, D.A., Komorowski, J., Zytkow, J. (Eds.), Fourth European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2000). Springer, Berlin, pp. 255–264.

Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, second ed. Academic Press, New York.