A Bayesian Metric for Evaluating Machine Learning Algorithms

Lucas R. Hope and Kevin B. Korb

School of Computer Science
and Software Engineering
Monash University
Clayton, VIC 3168, Australia

email: {lhope,korb}@csse.monash.edu.au

Abstract. How to assess the performance of machine learning algorithms is a problem of increasing interest and urgency as the data mining application of myriad algorithms grows. The standard approach of employing predictive accuracy has rightly been losing favor in the AI community. The alternative of cost-sensitive metrics provides a far better approach, given the availability of useful cost functions. For situations where no useful cost function can be found we need other alternatives to predictive accuracy. We propose that information-theoretic reward functions be applied. The first such proposal for assessing specifically machine learning algorithms was made by Kononenko and Bratko [1]. Here we improve upon our alternative Bayesian metric [2], which provides a fair betting assessment of any machine learner. We include an empirical analysis of various Bayesian classification learners, ranging from Naive Bayes learners to causal discovery algorithms.

Keywords: Evaluation metrics, Kullback-Leibler divergence, predictive accuracy, Bayesian evaluation, information reward.

1 Introduction

As academic and industrial interest in machine learning and data mining continues to grow, the problem of how to assess machine learning algorithms becomes more urgent. The standard practice for supervised classification learners has been to measure predictive accuracy (or its dual, classification error) using a fixed sample divided repeatedly into training and test sets, accepting a machine learner as superior to another if its predictive accuracy passes a statistical significance test. This account represents an improvement over historical practices, particularly when the statistical dependencies introduced by resampling are taken into account (cf. [3, 4]).

Nevertheless, there are a number of objections to the use of predictive accuracy, the most telling being that it fails to take into account the uncertainty of predictions. For example, a prediction of a mushroom's edibility with a probability of 0.51 counts exactly the same as a prediction of edibility with a probability of 1.0. We might rationally prefer to consume the mushroom in the second case, but not the first. Predictive accuracy shows no such discernment. According to common evaluation practice in machine learning and data mining every correct prediction is as good as every other. Hence, we advocate that classification learners should be designed, or redesigned, so as to yield probabilistic predictions rather than categorical predictions.

We believe a cost-sensitive assessment, favouring the machine learner which maximizes expected reward is, in principle, the best way of evaluating learning algorithms. Unfortunately, finding appropriate cost functions may be difficult or impossible. No expert may be available to provide a suitable cost function; or the algorithms being assessed may be applied across an open-ended variety of domains; or again the cost function may itself be evolving over time, as Provost and Fawcett point out [5]. Provost and Fawcett use receiver operating characteristic (ROC) convex hulls for evaluation independent of cost functions. This has the useful meta-learning feature of selecting the best predictor for a given performance constraint, in the form of a selected false

negative classification rate. Unfortunately, the ROC curves underlying this method again ignore the probabilistic aspect of prediction, as does predictive accuracy simpliciter.

Here we examine metrics that specifically attend to the estimated probability of a classification, but are also independent of cost, and so easier to apply than cost-sensitive metrics; namely information-theoretic measures and in particular, *information reward* (*IR*). We illustrate its application in some empirical results comparing Naive Bayes with other classification learners, contrasting *IR* with predictive accuracy assessments.

2 Kullback-Leibler Divergence for Classification

There are two fundamental ingredients to gambling success, and we would like our measure to be maximized when they are maximized:

Property 1: Domain knowledge, which can be measured by the frequency with which one is inclined to assert correctly $x_i = T$ or $x_i = F$ — i.e., by predictive accuracy. For example, in sports betting the more often you can identify the winning team, the better off you are.

Property 2: Calibration, the tendency of the bettor to put $P(x_i = T)$ close to the objective probability (or, actual frequency). That betting reward is maximized by perfect calibration is proven as Theorem 6.1.2 in Cover and Thomas's *Elements of Information Theory* [6].

With Property 1 comes a greater ability to predict target states; with Property 2 comes an improved ability to assess the probability that those predictions are in error. These two are not in a trade-off relationship: they can be jointly maximized.

If we happen to have in hand the true probability distribution over the target variables, then we can use Kullback-Leibler divergence to measure how different the model's posterior distribution is from the true distribution. That is, we can use:

$$\mathrm{KLD}(p, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \tag{1}$$

where X is a discrete random variable, and p and q are probability distributions.

Kullback-Leibler divergence has the two properties of betting reward in reverse: that is, by minimizing KLD you maximize betting reward, and vice versa. Clearly, divergence is minimized when $q = p$, which also implies that the model is perfectly calibrated. But it also implies Property 1 above: there is no more probabilistic information to be had once you know exactly the true probability distribution in the model. KLD, then, is an ideal metric for evaluating machine learning. It has the severe drawback in practice of requiring that the true probability distribution be available, which means there is no real learning problem. Nevertheless, KLD provides a useful benchmark for assessing other metrics.

3 Information-theoretic Metrics

3.1 Good's Information Reward

The original information reward (*IR*) was introduced by I.J. Good [7] as *fair betting fees* — the cost of buying a bet which makes the expected value of the purchase zero. Good's *IR* positively rewarded binary classifications which were informative relative to a uniform prior. *IR* is split into two cases: that where the classification is correct, indicated by a superscripted '+', and where the classification is incorrect, indicated by a superscripted '−'.

Definition 1. *The IR of a binary classification with probability p' is*

$$I^+ = 1 + \log_2 p' \qquad \textit{(for correct classification)} \tag{2a}$$

$$I^- = 1 + \log_2 (1 - p') \quad \textit{(for misclassification)} \tag{2b}$$

IR has the range $(-\infty, 1)$. For successful classification, it increases monotonically with p', and thus is maximized as p' approaches 1; for misclassification, *IR* decreases monotonically.

While the constant 1 in (2a) and (2b) is unnecessary for simply ranking machine learners, it makes sense in terms of fair fees. When the learner reports a probability of 0.5, it is not communicating any *information* (given a uniform prior), and thus receives a zero reward. Ignoring the constant 1, Good's *IR* has a clear information-theoretic basis: it reports (the negation of) the number of bits required in a message reporting an outcome of the indicated probability. Thus, a certain message requires no bits at all, whereas a certainly false message can never be communicated successfully, requiring an infinitely long message.[1]

Our work generalizes Good's to multinomial classification tasks, while also relativizing the reward function to non-uniform prior probabilities.

3.2 Kononenko and Bratko's Metric

The measure introduced by Kononenko and Bratko [1] also relativizes reward to prior probabilities. Furthermore, it too is nominally based upon information theory. This foundation is seriously undermined, however, by their insistance that when a reward is applied to a correct prediction with probability 1 and an incorrect prediction also with probability 1, the correct and incorrect predictions ought precisely to counterbalance, resulting in a total reward of 0. This conflicts with the supposed information-theoretic basis: on any account in accord with Shannon, a reward for a certain prediction coming true can only be finite, while a penalty for such a *certain* prediction coming false must always be infinite. Putting these into balance guarantees there will be no proper information-theoretic interpretation of their reward function.

We nevertheless agree that the kind of cost-neutral reward we are attempting to identify here needs to be relativized to prior probability. Otherwise, there is no way to avoid rewarding a learner which slavishly mimicks frequencies in a training set and no way to penalize algorithms which simply fail to learn from such frequencies.

Kononenko and Bratko introduce the following reward function, where p' is the estimated probability and p is the prior:

$$I^+_{KB} = \log p' - \log p \qquad \text{(for } p' \geq p\text{)}; \tag{3a}$$

$$I^-_{KB} = -\log(1-p') + \log(1-p) \qquad \text{(for } p' < p\text{)}. \tag{3b}$$

This measure is assessed against the *true* class only. Since the probabilities of other classes are not considered, in multinomial classification a miscalibrated assessment of the alternative classes will go unpunished. It follows that KLD fails to be minimized. For all these reasons we do not consider the Kononenko and Bratko function to be adequate.[2]

3.3 Bayesian Information Reward

The idea behind fair fees, that you should only be paid for an *informative* prediction, is simply not adequately addressed by Good's *IR*. Suppose an expert's job is to diagnose patients with a disease that is carried by 10% of some population. This particular expert is lazy and simply reports that each patient does not have the disease, with 0.9 confidence. Good's expected reward per patient for this strategy is $0.9(1 + \log_2 0.9) + 0.1(1 + \log_2 0.1) = 0.531$, so the expert is rewarded substantially for the uninformed strategy! The expected reward per patient we should like to see is 0, which the generalization below provides. Good's *IR* breaks down in its application to multinomial classification: any *successful* prediction with confidence less than 0.5 is penalized,

[1] Some have complained about infinite rewards rendering the resultant arithmetic of reward trivial. But it is not the arithmetic at fault in any such case, it is the predictor expressing an unfounded, miscalibrated and absolute confidence in its prediction which is at fault. There is no information-theoretic substitute for a negatively infinite reward in such cases.

[2] We did, however, include it in the empirical evaluation of [2]. See also Figure 1.

even when the confidence is greater than the prior. Good's fair fees are actually fair only when both the prior is uniform and the task binary.

Here is the Bayesian metric we presented in Hope and Korb [2]:

Definition 2. *The Bayesian IR for a classification into classes* $\{C_1, \ldots, C_k\}$ *with estimated probabilites* p'_i *and prior probabilities* p_i*, where* $i \in \{1, \ldots, k\}$*, is*

$$IR = \frac{\sum_i I_i}{k} \tag{4}$$

where $I_i = I_i^+$ *below for correct classes and* I_i^- *for incorrect classes:*

$$I_i^+ = 1 - \frac{\log p'_i}{\log p_i} \qquad \textit{(for correct classification)} \tag{4a}$$

$$I_i^- = 1 - \frac{\log (1 - p'_i)}{\log (1 - p_i)} \quad \textit{(for misclassification)} \tag{4b}$$

A non-uniform prior p can be obtained any number of ways, including being set subjectively (or arbitrarily). In our empirical studies here we simply use the frequency in the test set given to the machine learner to compute the prior.[3] This is because we have no informed prior to work with, and because it is simple and unbiased relative to the learning algorithms under study.

Bayesian information reward reflects the gambling metaphor more adequately than does Good's *IR*. Book makers are required to take bets for and against whatever events are in their books, with their earnings depending on the spread between bets for and against particular outcomes. They are, in effect, being rated on the quality of the odds they generate for all outcomes simultaneously. Bayesian *IR* does the same for machine learning algorithms: the odds (probabilities) they offer on all the possible classes are simultaneously assessed, extracting maximum information from each probabilistic classification.

Unfortunately this information reward fails to maximally reward perfect calibration, violating our own Property 2![4] We provide a restricted proof of this.

Theorem 1. *Given a batch classification learner (which does not alter its probability estimates after seeing the training data) and a binary classification problem, IR (per Definition 2 above) is* not *necessarily maximal when* $p'_i = f$*, where* f *is the frequency of the target class in the test set.*

Proof. Let S be a test set such that all data items have the same attributes (other than the binary target class value). If the actual test set is not of this kind, we can partition it so that each subset is; since the theorem holds of each subset, it will also hold for the union. To the classifier these items are indistinguishable. S is split into S_1 and S_2 with all items in S_1 belonging to the target class and all items in S_2 belonging to the complement class.

The average *IR* for machine learner M on items S is:

$$IR(M) = 1 - \sum_{i \in S_1} \frac{\log p'_i}{|S| \log p} - \sum_{i \in S_2} \frac{\log(1 - p'_i)}{|S| \log(1 - p)} \tag{5}$$

Since M is a batch machine learner, it will respond with the same probability p' to each item. So,

$$IR(M) = 1 - \frac{|S_1| \log p'}{|S| \log p} - \frac{|S_2| \log(1 - p')}{|S| \log(1 - p)} \tag{6}$$

The fractions $\frac{|S_1|}{|S|}$ and $\frac{|S_2|}{|S|}$ approximate the true probabilities of being in S_1 and S_2 respectively, so replacing these we get:

$$IR(M) \approx 1 - f \frac{\log p'}{\log p} - (1 - f) \frac{\log(1 - p')}{\log(1 - p)} \tag{7}$$

[3] We start the frequency counts at 0.5 to prevent overconfident probabilities.

[4] Thanks to David Dowe for pointing this out.

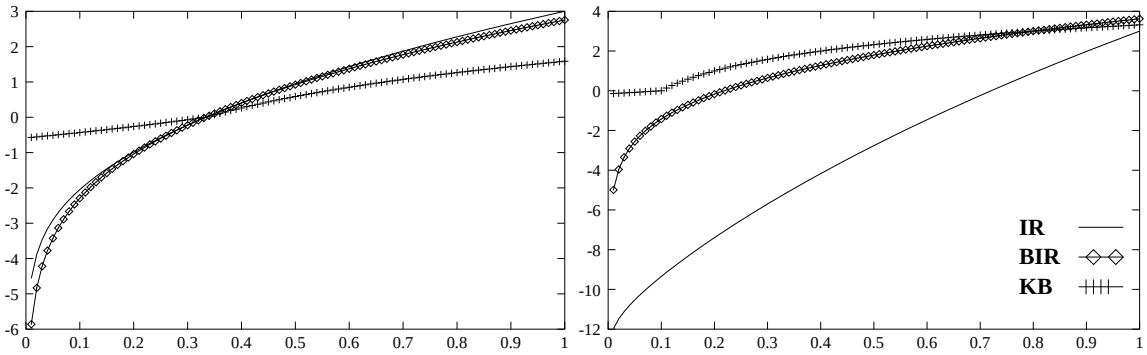

Fig. 1. Graphs comparing Information Reward (IR), new Information Reward (BIR) and Kononenko and Bratko's "Information Criterion" (KB) for a ternary class with (a) uniform prior and (b) prior of (.1, .45, .45) with the first being the true class.

Since we want to find the maximum reward, differentiate with respect to p' and set to 0:

$$\frac{dIR(M)}{dp'} = -\frac{f}{p' \log p} + \frac{1-f}{(1-p')\log(1-p)} = 0 \tag{8}$$

Thus:

$$\frac{f(1-p')}{p'(1-f)} = \frac{\log p}{\log(1-p)} \tag{9}$$

The only way f can equal p' is if p is 0.5, i.e., under a uniform prior, which is just the kind of restriction we were attempting to eliminate.

4 Information Reward Corrected

A new metric IR_B retains all the virtues introduced previously, while also being maximal under perfect calibration. For classification into classes $\{C_1, \ldots, C_k\}$ with estimated probabilities p'_i and priors p_i, where $i \in \{1, \ldots, k\}$:

$$IR_B = \frac{\sum_i I_i}{k} \tag{10}$$

where $I_i = I_i^+$ for the true class and $I_i = I_i^-$ otherwise, and

$$I_i^+ = \log \frac{p'_i}{p_i}$$

$$I_i^- = \log \frac{1-p'_i}{1-p_i}$$

Clearly, when $p' = p$, the reward is 0. IR_B also retains an information-theoretic interpretation: the measure is finitely bounded in the positive direction, since prior probabilities are never zero, and misplaced certainty (i.e., when the probability for the true value is 0) warrants an infinite negative reward. Finally, correct probabilities are now rewarded maximally in the long run. The proof of this is omitted; however, it is structurally similar to the proof in Section 3.3 and is available in [8, §10.8].

Figure 1 illustrates the score awarded by Information Reward, the new Bayesian Information Reward (BIR) and Kononenko and Bratko's "Information Criterion" (KB) for a ternary class. The horizontal axis shows the confidence in the true class (with the other two classes deemed equally likely). Note the kink in the KB metric where $p' = p$; this reflects its failure to support an information-theoretic interpretation. The old IR and IR_B are nearly indistinguishable given the uniform prior (in Figure 1a), but IR diverges radically from typical information measures under the non-uniform assumption of Figure 1b).

5 Empirical Evaluation

Our empirical evaluation focuses on machine learners that form Bayesian models, partially in response to recent work showing the surprising power of Naive Bayes learners and their relatives (e.g., [9–11]). We test learners with artficial data generated from models that specifically favour the simpler Bayesian learners. The machine learners were all implemented in Java, using the Weka data mining suite [12].

1. **Naive Bayes (NB)** is the simplest Bayesian learner; it assumes each attribute is independent of the others, given knowledge of the target class. Despite the strong assumption, it routinely performs well on many well known datasets [2]. We use an algorithm for Naive Bayes that incorporates Gaussian estimation for continuous attributes [13].
2. **Tree Augmented Naive Bayes (TAN)** [10] adds an additional tree-like dependency structure to Naive Bayes, thus each attribute may have one parent from amonst the other attributes, in addition to the target class. This method is complicated by the task of searching amongst attribute dependency structures. We use the algorithm presented by [14], with continuous attributes discretized using the technique of [15].
3. **Averaged One Dependence Estimators (AODE).** One dependence estimators are Naive Bayes variants where one attribute is nominated to be a parent of all the others, in addition to the standard naive dependence on the target class. Webb *et al.* [9] present a technique where an average over all possible ODEs is used for prediction. This introduces a favourable bias in the evaluation, since it is well known that averaging predictors perform better than predictors based upon selecting a single model; nevertheless, we ignore this bias here. Again, continuous attributes are discretized using the technique of [15].
4. **Causal MML (CaMML)** is a Bayesian network learner which uses Minimum Message Length [16] to rank causal structure [17]. We use CaMML's best model to predict the target class given a test instance's other attributes. Continuous attributes are discretized in the same way as TAN and AODE.
5. **J48.** We also include Weka's J48 [12], an implementation of Quinlan's C4.5 [18].

5.1 Empirical Study: Bayesian Models

For this experiment, we use artificially generated data from a series of Bayesian model types. Three model types are chosen, each designed to favour a particular machine learner: Naive Bayes, TAN or AODE. Thus, we see how the learners perform when their assumptions are broken, in comparison with when those assumptions are exactly matched.

Out of our set of machine learners, CaMML generates models with greatest complexity. Given the standard convergence results for Bayesian learners, CaMML must better or equal every other machine learner in the limit. Similarly, AODE's and TAN's models are more complex than the Naive Bayes model, and given sufficient data they should perform at least on par with Naive Bayes. This implies a converse phenomenon. At low levels of data, and *if the learner's representations include the true model*, the simpler learner should outperform the more complex, because complex machine learners converge to their optimum models slower, due to a larger search space.

To test the threshold at which a simpler model outperforms the more complex, we generate three sets of models, based upon the underlying models of Naive Bayes, TAN and AODE respectively. We also systematically vary the amount of training data given to each learner.

Below we describe the experimental design in detail, then discuss the individual differences between model types and analyse the results.

Experimental method For statistical analysis, we regard each model type as a separate experiment, with the experimental design identical for each model type. For each experiment we sample the space of appropriate models (the exact details are described below). Each model has 4–8 attributes (including the target attribute), with each attribute having 2–5 values. The probabilities in each attribute are determined randomly. We sample forty models and perform a two-factor

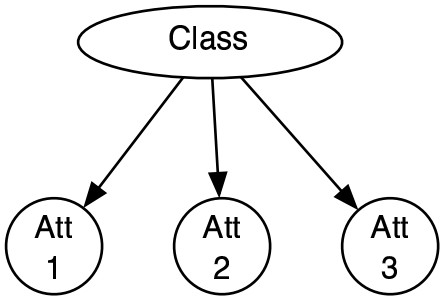

Fig. 2. An example Naive Bayes model.

repeated measures ANOVA, in order to provide a statistical test independent of our Bayesian assumptions. The two factors are (1) machine learner applied (we test Naive Bayes, TAN, AODE and CaMML) and (2) amount of training instances we pass to each learner (the amounts are 50, 500 and 5000). The training set sizes are chosen to represent small, medium and large datasets, while keeping the size of the analysis to a minimum. It is advantageous to use a repeated measure ANOVA because this design controls for the individual differences between samples (where each model is considered a sample).

We use information reward on a single test set of 1000 instances for each model to measure the 'treatment' of each machine learner at different 'doses' (amounts of training data). We don't report accuracy nor Kononenko and Bratko's measure, for the reasons we argued in Sections 1 and 3.2. Where we report confidence intervals, these have been adjusted by the Bonferroni method for limiting the underestimation of variance [19].

Naive Bayes models Naive Bayes models (shown in Figure 2) follow the assumptions governing the Naive Bayes learner; each attribute is conditionally independent of each other, given the target variable. This is the simplest model type we use in this evaluation, so we expect that all learners will perform reasonably.

Figure 3 shows the performance of the machine learners for each amount of training data. Note that Naive Bayes, TAN and AODE perform similarly for each level. Unsurprising, as they all share the (correct) assumption that the target class is a parent of all other attributes in the model. For small amounts of data, CaMML performs significantly worse than the other learners: it cannot reliably find the correct model. As more data become available, it finds the correct model and achieves a similar score to the other learners.

Tree Augmented Naive models TAN models (see Figure 4) are formed by creating a tree-like dependency structure amongst the (non-target) attributes, then making them all directly dependent upon the target class. This is more complicated than the Naive Bayes model above. Each model we generate has a random tree structure amongst the non-target attributes.

Surprisingly, the results in Figure 5 show that TAN is not the best learner with low amounts of training data: AODE stands superior for low amounts of training instances. This is likely because AODE has a richer representation than Naive Bayes (i.e., with averaged predictions), yet doesn't need to search for the tree structure. Once there are enough data both TAN and CaMML seem to find the right structure and thus both outperform AODE. This demonstrates the added difficulty of model selection. Although TAN assumes the correct model type, it still has to find the particular tree structure for each model, thus TAN's performance is dependent on its search capabilities.

Naive Bayes, with its inaccurate assumptions, is clearly inferior to the other learners once an adequate amount of training data is given.

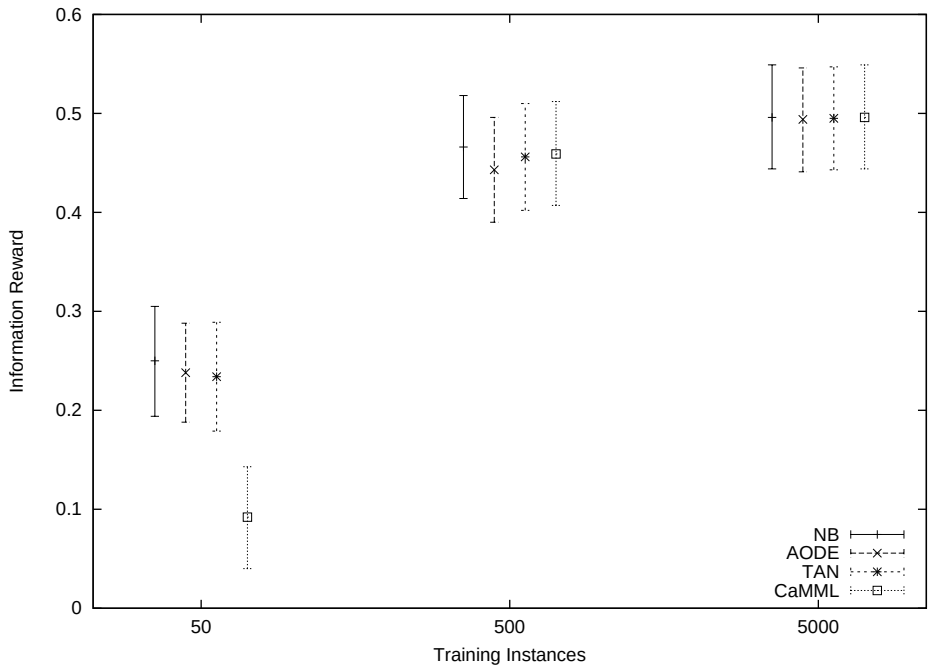

Fig. 3. Summary of results for the Naive Bayes experiment for dataset sizes 50, 500 and 5000 (confidence intervals at 95% are shown).

Averaged One-Dependence models AODE forms a series of n models where, in the ith model, attribute i is the parent of each other (non-target) attribute. Similar to Naive Bayes, each attribute is also directly dependent on the target (shown in Figure 6). Thus, each AODE model is a hybrid of several ODE models, with each model having equal chance to be selected from when sampling the model for data.

Judging from the results in Figure 7, this hybrid model seems to be very difficult for the machine learners to learn, with the information reward ranging from -0.3 to 0.1. Recall that a reward of zero corresponds to a machine learner which finds no associations amongst the non-target attributes, returning the observed frequency of the target class as its estimate, and it takes more than 50 training instances to achieve a score higher than zero! The likely reason is that the process for selecting a model to sample from is hidden from the learners: it's a hidden (or confounding) variable, of significant impact.

It is still puzzling that the machine learners other than CaMML score worse than zero for low amounts of training data. The answer seems to lie in each learner's assumptions: Naive Bayes, TAN and AODE each assume a model where all attributes depend on the target, regardless of whether this model decreases performance. CaMML is not beholden to any particular model, and thus is free to choose no association at all. This conservatism wins out, even against Naive Bayes with small datasets. After enough training data, AODE (the only learner that can really model the data properly) obtain an advantage over the other learners.

5.2 Empirical Study: UCI Archives

To supplement the evaluation on artificial data, we now evaluate a host of datasets from the standard UCI archives, and also from the Naive Bayes research literature (e.g., [9]). We choose these datasets so our results can be more easily compared with other empirical evaluations in the literature.

There are numerous pitfalls to avoid when performing a large empirical evaluation involving real-world datasets (see [20, 3] for commentaries). Briefly, the standard use of paired t-tests to

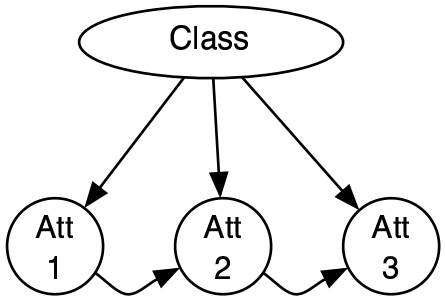

Fig. 4. An example TAN model.

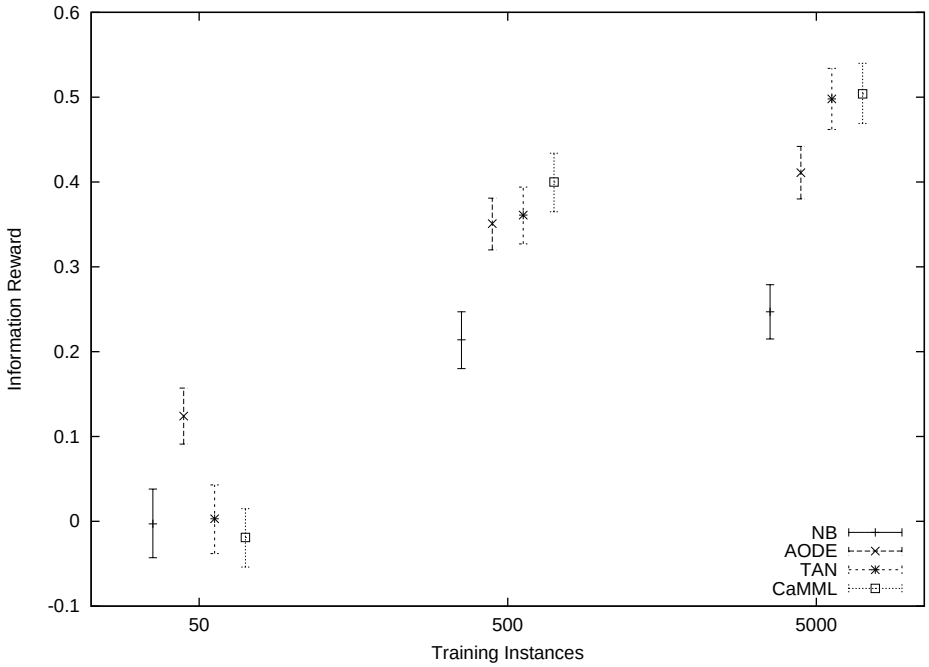

Fig. 5. Summary of results for the TAN experiment for dataset sizes 50, 500 and 5000 (confidence intervals at 95% are shown).

determine significant difference is undermined by dependencies introduced by resampling data. Since resampling is an unfortunate necessity, we use the 5×2cv test [4] which reduces Type 1 error to an acceptable level. This method introduces a special paired-t formula, calculated by repeating a two-fold cross-validation five times. We slightly alter the technique by stratifying the cross-validation folds, thereby increasing its statistical power. All significance tests are performed at the two-tailed $p < 0.05$ level.

The raw results are lengthy, so we show them in Appendix A. Confidence intervals are omitted because the 5×2cv test only provides pairwise significance comparisons. These comparisons are shown in Tables 1 and 2. Columns show individual machine learners with each row devoted to a dataset. The cells of the table contain the names of the machine learners deemed significantly *inferior* to that column's machine learner.

Upon initial analysis of the results, it seems quite clear that Naive Bayes is clearly inferior to the other Bayesian learners. Also, when Naive Bayes beats J48, it is almost certain that the other Bayesian learners will also be superior. In fact, the information reward results invariably show this property amongst the datasets presented. Perhaps this is because the underlying model for these datasets is different from that assumed by decision trees. For decision trees the target class is

Dataset	Naive Bayes	J48	CaMML	AODE
adult			AODE, TAN	
anneal				
balance-scale	J48, AODE, TAN			TAN
bcw	J48			
bupa				
chess				
cleveland	J48			
crx			TAN	TAN
echocardiogram				
german	J48			
glass				
heart				
hepatitis				
horse-colic			TAN	AODE
house-votes-84				
hungarian			NB	
hypothyroid			AODE	
ionosphere			TAN	TAN
iris		CaMML		
labor-neg				
led	J48, AODE			
letter-recognition				TAN
lung-cancer			AODE, TAN	
mfeat-mor			TAN	
new-thyroid	J48			
pendigits				TAN
post-operative			AODE, TAN	
promoters	J48			
ptn	J48, CaMML			
satellite			AODE, TAN	
segment				
sign			AODE, TAN	
sonar			TAN	TAN
syncon	J48			
ttt			AODE, TAN	TAN
vehicle				
wine				

Table 1. Information reward results for all learners on the datasets. Columns represent winners, cells losers. E.g., NB performed significantly worse than all other learners on the "hypothyroid" dataset.

Dataset	Naive Bayes	J48	CaMML	AODE
adult		AODE	AODE	
anneal				
balance-scale	J48, CaMML, AODE, TAN			
bcw				
bupa				
chess	CaMML	CaMML		
cleveland	J48			
crx				
echocardiogram	J48, CaMML, AODE, TAN			
german	J48, CaMML			
glass				
heart	J48			
hepatitis	J48, TAN			
horse-colic			TAN	TAN
house-votes-84				
hungarian	J48, CaMML			
hypothyroid				
ionosphere				TAN
iris		CaMML		
labor-neg				
led	J48			
letter-recognition		CaMML, TAN		TAN
lung-cancer				
mfeat-mor	TAN	CaMML, TAN		
new-thyroid	J48			
pendigits				TAN
post-operative		AODE, TAN	AODE, TAN	
promoters	J48			
ptn	J48, CaMML			
satellite				
segment				
sign		CaMML, AODE, TAN	AODE, TAN	
sonar				
syncon	J48			
ttt		AODE, TAN	AODE, TAN	TAN
vehicle		CaMML, AODE		
wine				

Table 2. Accuracy results for all learners on the UCI archives. Columns represent winners, cells losers. E.g., NB performed significantly worse than all other learners on the "sign" dataset.

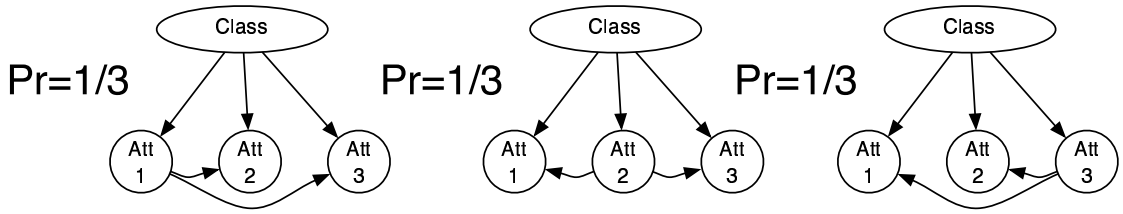

Fig. 6. An example AODE model.

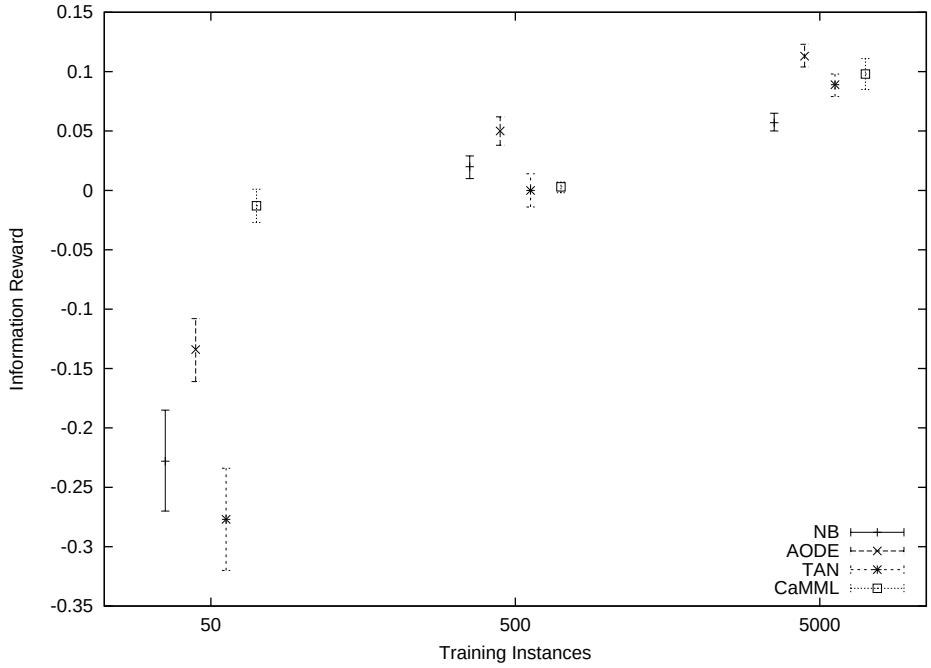

Fig. 7. Summary of results for the AODE experiment for dataset sizes 50, 500 and 5000 (confidence intervals at 95% are shown).

dependent (in a symptomatic sense) on all the other attributes. For Naive Bayes, TAN and AODE the opposite is true: the target class is seen as the root cause for the other attributes. CaMML has the advantage in that it can choose whichever model is more appropriate.

Taking results from amongst the three advanced Bayesian learners (TAN, AODE and CaMML), AODE seems to outperform the others a surprising amount of the time. TAN never beats AODE, and only beats CaMML on the 'ptn' dataset. A possible reason for this is the problem of model selection. TAN and CaMML must search large model spaces for the best fitting model, and thus will likely perform poorly if a suboptimal model is chosen. AODE provides good estimates by choosing a family of models, more complex than Naive Bayes yet simpler than TAN, and averaging over them, thus avoiding the model selection.

In comparing accuracy versus information reward, we can see that the results commonly differ, with the 'sign' dataset showing the largest reversal on significance results: accuracy reports that J48 is better than any other machine learner on this dataset; information reward rates J48 as worse than TAN, AODE and CaMML. For more comprehensive reports comparing these two measures empirically, see [21, 2].

6 Conclusion

We have reviewed a number of metrics for the evaluation of machine learners. Accuracy is too crude, optimizing only domain knowledge while ignoring calibration. Kullback-Leibler divergence, on the other hand, is ideal — so ideal it is inapplicable to real-world data. Other information-theoretic metrics were found wanting, including our prior information reward metric. We have developed a new metric which is shown to be maximized under the combination of domain knowledge and perfect calibration. This information reward evaluates learners on their estimate of the whole class distribution rather than on a single classification. In rewarding calibration, it provides a valuable alternative to cost-sensitive metrics when costs are unavailable.

We applied information reward to Bayesian machine learners using artificial data in order to find when one is superior to another. We found that more powerful learners such as CaMML can pay a performance penalty when there is a sparsity of data. Indeed, we found that under some conditions, it is better to not model the data at all: sometimes zero information reward is the best you can do. We also confirmed that avoiding model selection can lead to superior predictive results, e.g., AODE's superior performance with TAN models.

Finally, we compared accuracy and information reward on some real world datasets and found results can be reversed. The differing verdicts, and the theoretical superiority of information-theoretic metrics, make a good case for a general change in experimental practices in machine learning studies.

Future Work. While we have developed some telling theoretical criticisms of predictive accuracy, the Kononenko-Bratko metric and (implicitly) ROC measures, we are aware that many will remain unconvinced by theoretical arguments. We are developing a 'meta-metric' to evaluate these and other evaluation functions empirically — one which is not biased towards Bayesian metrics. This should allow us to test all these metrics and more empirically and, we hope, provide even more telling support for Bayesian Information Reward.

References

1. Kononenko, I., Bratko, I.: Information-based evaluation criterion for classifier's performance. Machine Learning **6** (1991) 67–80
2. Hope, L.R., Korb, K.B.: Bayesian information reward. In Mckay, B., Slaney, J., eds.: Lecture Notes in Artificial Intelligence. Volume 2557. Springer-Verlag, Berlin, Germany (2002) 272–283
3. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI. (1995) 1137–1145
4. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation **7** (1998) 1895–1924
5. Provost, F., Fawcett, T.: Robust classification for imprecise environments. Machine Learning **42** (2001) 203–231
6. Cover, T., Thomas, J.: Elements of Information Theory. Wiley, New York (1991)
7. Good, I.J.: Rational decisions. Journal of the Royal Statistical Society. Series B (Methodological) **14** (1952) 107–114
8. Korb, K.B., Nicholson, A.E.: Bayesian Artificial Intelligence. Chapman & Hall/CRC, (Boca Raton, Florida)
9. Webb, G.I., Boughton, J., Wang, Z.: Averaged One-Dependence Estimators: Preliminary results. In: Proceedings of the Australasian Data Mining Workshop, University of Technology, Sydney, Australia (2002) 65–73
10. Friedman, N., Goldszmidt, M.: Building classifiers using Bayesian networks. In: AAAI-96. (1996) 1277–1284
11. Zheng, Z., Webb, G.I.: Lazy learning of Bayesian rules. Machine Learning **41** (2000) 53–84
12. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000) http://www.cs.waikato.ac.nz/ml/weka/.
13. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: UAI 11, Morgan Kaufmann, San Mateo (1995) 338–345

14. Keogh, E., Pazzani., M.: Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In: Int. Workshop on Artificial Intelligence and Statistics. (1999) 225–230
15. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: IJCAI. (1993) 1022–1029
16. Wallace, C., Boulton, D.: An information measure for classification. The Computer Journal **11** (1968) 185–194
17. Wallace, C.S., Korb, K.B., Dai, H.: Causal discovery via MML. In: ICML'96, Morgan Kaufmann (1996) 516–524
18. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann (1993)
19. Keppel, G.: Design and Analysis: A Researcher's Handbook. Prentice-Hall, (San Mateo)
20. Salzberg, S.L.: On comparing classifiers: A critique of current research and methods. Data Mining and Knowledge Discovery **1** (1999) 1–12
21. Korb, K.B., Hope, L.R., Hughes, M.J.: The evaluation of predictive learners: Some theoretical and empirical results. In De Raedt, L., Flach, P., eds.: European Conference on Machine Learning (ECML'01). (2001) 276–287

A Raw Scores

A.1 Information Reward

Here are the raw results for Information Reward. No bounds are given on the results because the 5x2cv method [4] only produces pairwise significance results. These are shown in Table 1.

dataset	Naive Bayes	J48	CaMML	AODE	TAN
adult	7.22980	7.83770	7.96496	7.87052	7.88208
anneal	6.78609	7.35020	7.30007	7.69314	7.66789
balance	5.62249	4.84739	5.43315	5.44021	5.41536
bcw	5.82035	5.51422	5.82631	5.91234	5.82281
bupa	3.88562	3.17644	4.07791	4.08645	4.08642
chess	4.33390	4.20745	4.18040	4.35184	4.37450
cleveland	4.46802	3.74239	4.57949	4.66777	4.56869
crx	5.50976	6.23944	6.41087	6.37981	6.24681
echocardiogram	3.15361	2.63586	3.06695	3.14662	3.11833
german	5.15109	4.32249	5.25952	5.28885	5.23247
glass	4.24639	4.70175	5.21894	5.33640	5.23926
heart	4.03220	3.40950	4.06098	4.16457	4.10956
hepatitis	1.60088	1.48358	1.62095	1.80062	1.66564
horse	4.28402	4.82144	4.82625	4.81158	4.49184
house	4.60104	5.08948	5.18538	5.13808	4.72888
hungarian	4.59802	4.34745	4.62318	4.77714	4.66284
hypothyroid	0.59064	0.67239	0.68895	0.66254	0.65027
ionosphere	3.31030	3.82483	4.10966	4.07039	3.83852
iris	5.86626	5.61783	5.77659	5.83336	5.82947
labor	3.60224	3.07112	3.10452	3.46798	3.72117
led	9.19016	8.78243	9.18342	9.17474	9.18229
letter	13.81702	14.07254	14.96533	15.22630	15.01995
lung	1.34683	1.71182	2.74738	1.35843	1.30984
mfeat	10.20241	9.89526	10.43757	10.39758	10.33721
new	3.92654	3.31574	3.87709	3.85632	3.87262
pendigits	13.21161	14.11380	14.42818	14.56372	14.46698
post	1.31190	1.25769	1.54332	1.26249	1.22681
promoters	4.02273	2.82232	4.12059	3.90486	4.02746
ptn	5.45322	4.33701	5.02789	5.48000	5.42786
satellite	9.84297	10.90947	11.80021	11.54662	11.08514
segment	9.82980	11.22429	11.39362	11.32126	11.25842
sign	10.36300	10.55098	11.01350	10.91561	10.87566
sonar	2.66956	2.09511	3.69163	3.61604	3.50905
syncon	8.88879	8.06741	9.16254	9.15553	8.92866
ttt	5.32365	5.18512	5.94521	5.43531	5.33428
vehicle	5.80541	7.13489	7.96226	7.94813	7.25499
wine	6.26350	5.56457	6.34221	6.35151	6.36883

A.2 Accuracy

Here are the raw scores for accuracy. No bounds are given on the results because the 5x2cv method [4] method only produces pairwise significance results. These are shown in Table 2.

dataset	Naive Bayes	J48	CaMML	AODE	TAN
adult	0.83152	0.85894	0.86604	0.85034	0.85998
anneal	0.80779	0.88351	0.86414	0.95256	0.96280
balance-scale	0.88000	0.77792	0.75294	0.75933	0.76413
bcw	0.97167	0.93019	0.97282	0.96623	0.97224
bupa	0.53813	0.63300	0.54368	0.55182	0.55182
chess	0.85336	0.88277	0.80290	0.85845	0.86027
cleveland	0.82571	0.75178	0.81121	0.82571	0.82042
crx	0.77623	0.85478	0.85971	0.86086	0.85304
echocardiogram	0.73571	0.62899	0.65039	0.64578	0.64424
german	0.73739	0.7014	0.7242	0.73639	0.7292
glass	0.60467	0.73177	0.67757	0.70560	0.69626
heart	0.83555	0.76148	0.80222	0.82370	0.82222
hepatitis	0.84495	0.80516	0.79881	0.83483	0.82960
horse-colic	0.78423	0.82989	0.81739	0.825	0.81304
house-votes-84	0.89928	0.95079	0.94894	0.94112	0.90939
hungarian	0.82993	0.78503	0.78435	0.83741	0.83401
hypothyroid	0.97824	0.99152	0.98969	0.98640	0.98406
ionosphere	0.81833	0.88436	0.90144	0.91057	0.90089
iris	0.95066	0.94133	0.94000	0.94133	0.93999
labor-neg	0.89815	0.80307	0.79236	0.82450	0.91551
led	0.7408	0.718	0.7404	0.7398	0.739
letter-recognition	0.64079	0.84472	0.82464	0.86247	0.83667
lung-cancer	0.39375	0.40625	0.28125	0.375	0.3875
mfeat-mor	0.692	0.70639	0.6865	0.69049	0.6914
new-thyroid	0.97017	0.90423	0.94792	0.94698	0.94698
pendigits	0.85693	0.95311	0.95160	0.97210	0.95918
post-operative	0.63555	0.67999	0.71111	0.62444	0.61777
promoters	0.87924	0.75283	0.88679	0.84905	0.87735
ptn	0.45606	0.37347	0.32803	0.46256	0.45312
satellite	0.79512	0.85292	0.85302	0.88397	0.86700
segment	0.79757	0.95272	0.93818	0.93567	0.93783
sign	0.50717	0.82628	0.74895	0.71967	0.72692
sonar	0.69615	0.69519	0.68942	0.73557	0.73173
syncon	0.943	0.86200	0.97066	0.96399	0.94333
ttt	0.70835	0.82045	0.89457	0.74175	0.71607
vehicle	0.44113	0.70141	0.63380	0.67848	0.63995
wine	0.96404	0.89550	0.97191	0.97640	0.97752