Recall & Precision versus The Bookmaker

David M. W. Powers (powers@computer.org)
School of Informatics & Engineering
The Flinders University of South Australia
GPO Box 2100 Adelaide 5001 Australia

Abstract

In the evaluation of models, theories, information retrieval systems, learning systems and neural networks we must deal with the ubiquitous contingency matrix of decisions versus events. In general this is manifested as the result matrix for a series of experiments aimed at predicting or labeling a series of events. The classical evaluation techniques come from information retrieval, using recall and precision as measures. These are now applied well beyond this field, but unfortunately they have fundamental flaws, are frequently abused, and can prefer substandard models. This paper proposes a well-principled evaluation technique that better takes into account the negative effect of an incorrect result and is directly quantifiable as the probability that an informed decision was made rather than a random guess.

Introduction

Throughout the cognitive sciences we frequently deal with the ubiquitous contingency matrix of decisions versus events. While this is in its own right a matter for research in the study of gambling and risk taking and any theory of rational choices, we here focus on its practical applications. It is perhaps better known in its demeaned or standardized guises of the covariance and correlation matrices, but here we treat its practical manifestation as the result matrix for a series of experiments aimed at labeling a series of events.

The labeling may be performed by a human amateur or expert, or by a knowledge-based system or expert system. It may be the result of the application of a black box neural network, or a prediction of a formally specified theory. In all cases there is some event that has to be predicted or labeled, and some source of predictions or labels. We assume that the event is not a random variable but there is a theoretical basis of predicting or identifying it based on its context – which may without loss of generality precede, accompany or succeed the event. We will designate the label, predictor or prediction P and the true label, class or result R, as is traditional in designing and evaluating decision trees.

Specific examples include predicting the outcome of a horse race based on history and form, decision support systems that decide eligibility for a government allowance, parsers or taggers that give syntactic or semantic labels to the words of a sentence, neural nets that aim to identify a psychological condition from real-time single-trial electroencephalograms, search engines that aim to retrieve specified documents or web pages based on query words.

The first of these examples has inspired the approach to be presented in this paper, while the last has spawned the techniques and terminology that are presently used for evaluating the tabulated results.

Recall, Precision and Accuracy

In information retrieval, such as web search, search criteria such as keywords and the documents returned are then evaluated for relevance. The proportion of relevant documents that is returned is called *recall*. The proportion of documents returned that are relevant is called *precision*. This classic situation corresponds to a binary decision problem: *result* and *prediction* labels are either relevant or irrelevant, yes or no, + or −.

A supervised learning system has similar labels but actually trains with known result labels and compares the results against the predicted labels. An unsupervised learning or classification system automatically invents a number of classes and we can then associate these classes with their most common prediction and assess them in the same way. In each case both binary and multiple classifications are meaningful, although the unsupervised case has an additional issue in that a different number of classes, *K*, may be determined rather than the number of classes expected, *C*.

The search problem assumes that there are a specified number of documents (class 1) that satisfy the criteria and may be found by the search process, and the others don't (class 2). This binary task thus gives us a four cell result matrix: if +P indicates the number of times that we *predicted* a + label (success) and +R is the times the actual *result* was a + label, then the four cell contingency matrix counts are |+P+R| (P true and R true – also known as TP for True Positive), |−P−R| (P false and R false – also known as TN for True Negative), |+P−R| (False Positive/FP), |−P+R| (False Negative/FN) and it has marginal sums |+P|, |−P|, |+R| and |−R| where |+P| + |−P| = |+R| + |−R| = *N*:

	+R	−R	
+P	42	18	60
−P	28	12	40
	70	30	100

The *recall* (≡ *sensitivity* ≡ *tpr*) measure is simply the proportion of all such instances available that are identified correctly, whilst the *precision* is the proportion of all predictions that are identified correctly. The difference between these is in the denominator – precision is |+P+R|/|+P| ≡ $p_{\mathbf{R}}(c{=}1|l{=}1)$ and recall is |+P+R|/|+R| ≡ $p_{\mathbf{P}}(l{=}1|c{=}1)$. The problem here is that we have two measures of fitness rather than one, and neither of them incorporates any penalty for making an error: the |+P−R| (FP) and |−P+R| (FN) cells correspond to type 1 and type 2 error with rates *fpr* ≡ FP/|−R| (≡ *fallout*) and *fnr* ≡ FN/|+R/ resp.

These errors are often taken account of by limiting the acceptable range of type 1 error to less than 5%, and either ignoring type 2 error or limiting it to a an acceptable range of up to 20%, but there is no penalty attached to the errors. These figures correspond to 95% precision and 80% recall, but often cognitive science experiments yield results with much greater error ranges and much lower precision and/or recall, and the problem is that there is usually a trade off of recall against precision.

The most common solution in information retrieval, machine learning and neural network research is some sort of average accuracy. Here we discuss the standard approaches, and show that they fail to capture meaningfully the extent to which informed decisions are being made.

Diverse Definitions of Accuracy

The mean precision across all prediction labels and the mean recall across result labels both give us the same accuracy figure, $|{+}P{+}R| + |{-}P{-}R| / N$, since the cases must be correctly weighted by their marginal probabilities. This definition of accuracy is formally known in statistics as the Rand Index (Rand, 1971) and may also be extended to providing an accuracy figure for unsupervised clustering even when the number of clusters determined by the system does not match the evaluation model. It gives equal weight to correct identification of both cases and applies to any classification of classes.

A closely related measure of accuracy is the Jaccard or Tanimoto Index $|{+}P{+}R| / (N - |{-}P{-}R|)$ which treats positive and negative data asymmetrically and is designed for occurrence statistics in which non-occurrence is not regarded as being of interest.

The standard average of recall and precision (Manning & Schutze, 1999) is a harmonic mean which in its most general forms provide a weight $\alpha \equiv \beta^2$ to determine the bias towards recall, r, or precision, p, but is more commonly used with $\alpha = 0.5$ (which we will assume henceforth in this paper in weighting r and p):

$$\begin{aligned} F &= p \cdot r / (\alpha\, p + (1-\alpha)\, r) \\ &= 1 / (\alpha / r + (1-\alpha) / p) \\ &= 2\, p \cdot r / (p + r) \qquad (\alpha = 0.5) \end{aligned}$$

This so-called F-factor is usually viewed as providing a bias towards the lower of the precision and recall. These are relative frequencies and it is effectively taking the mean of their reciprocal intervals ('wavelengths') as distinct measures of error rate (per +-case and per +-decision). There is no clear justification for the use of an α other than 0.5. Rather it seems to be used as a bias to reflect the realities that most models have lower recall than precision whilst the 0.5 model gives them equal weight.

The complements of recall and precision, *inverse recall* (*specificity)* and *inverse precision*, may be used to assess the accuracy in predicting non-occurence, and an inverse or negative F-factor may be derived using the corresponding formula. The positive and negative F-factors may furthermore be combined analogously in terms of their average error rates to produce a single accuracy figure.

An alternate method of combining recall, precision, inverse recall and inverse precision figures is to use the geometric mean. Noting that these precision and recall figures are actually conditional probabilities, the geometric mean may be interpreted as a perplexity measure corresponding to an arithmetic mean of the conditional information, the usual weighted form of which is the conditional entropy based on log precision:

$$H(\mathbf{R}|\mathbf{P}) = -\sum_{l \in \mathbf{P}} p_{\mathbf{P}}(l) \sum_{c \in \mathbf{R}} p_{\mathbf{R}}(c|l) \log(p_{\mathbf{R}}(c|l))$$

Note that the conditional entropy measure does not assume that the model has correctly determined the number of classes, C. Rather, K, the number of classes induced, may differ from C, the number of pre-labeled classes (Cover and Thomas, 1990). The binary case is $C{=}K{=}2$, $\mathbf{R} = \mathbf{P} = \{+, -\}$ or $\{1, 2\}$. This gives us a direct information-theoretic measure of the goodness of the classification, which may be regarded as a measure of parsimony – it is a measure of the expected number of bits of information required in addition to the predictive model in order to correctly identify a case.

These measures of accuracy thus correspond to the (weighted) arithmetic mean of recall or precision, the harmonic F-factor corresponds to the (weighted) arithmetic mean of case and prediction error rate, and the geometric mean corresponds to the (weighted) arithmetic mean of the case and prediction conditioned information.

The correct use of any of these averages to give an overall accuracy figure requires weighting them by the expected number of instances or predictions per label.

Label Bias

Part of the problem with recall and precision is that it encourages model developers to bias their models. For example in part-of-speech (POS) tagging it is common to ignore the possibility that nouns without a distinct verb form (invention<−invent, speak−>speaker/speech) may be used as verbs (Entwisle and Powers, 1998).

Many systems are poor at determining POS from syntactic cues alone and rely heavily on a lexicon to specify that (say) 'water' and 'shoulder' are nouns. If the error rate in determining the POS of a word is higher than the occurrence rate of a verbal usage of a noun, they can actually increase their accuracy by specifying that the word is always a noun. The occasional sentence where I 'water the garden' or 'shoulder someone aside' will have less impact on the accuracy figures than the impact of incorrectly labeling some of the nouns as verbs.

This problem is an instance of a more general label bias problem (Lafferty, McCallum and Pereira, 2002) which actually leads to a bias against cognitively, linguistically, and physiologically plausible models as illustrated above.

In an ideal model the distribution of predictions $|{+}P|{:}|{-}P|$ will reflect the *a priori* distribution of the data, $|{+}R|{:}|{-}R|$. Thus the marginal probabilities for the label $l \in \mathbf{P}$, $p_{\mathbf{P}}(l)$, and those for the corresponding actual class $c \in \mathbf{R}$, $p_{\mathbf{R}}(c)$, should be the same:

$$\begin{aligned} p_{\mathbf{P}}(1) \equiv |{+}P|/N &= p_{\mathbf{R}}(1) \equiv |{+}R|/N, \\ p_{\mathbf{P}}(2) \equiv |{-}P|/N &= p_{\mathbf{R}}(2) \equiv |{-}R|/N. \end{aligned}$$

If we consider the problem of correctly identifying a noun and know that 90% of the usages are nouns and 10% are verbs, but we always say 'noun', then we get the following contingency matrix:

	+R	−R	
+P	90	10	100
−P	0	0	0
	90	10	100

Note that with this trivial model the recall is 100% (90/90 nouns are identified correctly) and precision is determined by the prior probability of a noun as 90% (90/100 cases identified as nouns are correct).

An appropriately parameterized learning model or artificial neural network (ANN) should automatically tend to produce label predictions that mimic the observed distribution of classes. However many hand-crafted or hand-tuned models or systems take advantages of the weaknesses of recall and precision to achieve better “accuracy” by cheating as above.

In a supervised feedforward neural network (perceptron, backpropagation, etc.) the attribution of error to inputs in proportion to their influence, as given by their input weights, leads to stabilization at weights that give rise to an output distribution that matches the input distribution irrespective of whether there are learnable/learned patterns or not in the input. The same is true of unsupervised, self-organizing and associative networks based on Hebbian plasticity.

Sometimes a training set will be overtly “standardized” to avoid bias by the proportions of the different training examples either by equalizing the number of + and – cases or by discounting their weight. For example, if + and – occur with a 7:3 skew, + and – may be trained to output values of 1/7 and 1/3 respectively rather than 1, and negative cases where + or – doesn't occur to output values of −1/3 and −1/7 respectively rather than −1.

However, generally a learning system should aim to match the input distribution, and in a competitive learning system or ANN the thresholds can usually be adjusted to achieve such a match. A system that doesn't have a matching distribution cannot hope to achieve 100% accuracy across all conditions, but will have non-zero cross-correlations.

Bookmaker Odds

We now motivate an alternative accuracy measure using a betting scenario that contrasts with this recall and precision analysis by providing a penalty for errors based on fair (or rational) odds determined from the historical probability of each “horse” winning. This specifies what you win if you win as well as what you lose if you lose – in a ratio inverse to the ratio of probabilities. So an offer of odds of X:Y (or X/Y) for a horse means that if you (your horse) wins you win X, while if you lose you lose Y, and with fair odds this would indicate that the probability of winning is Y/(X+Y) and the probability of losing is X/(X+Y) and the expected gain is XY/(X+Y) – YX/(X+Y) = 0.

In empirical cognitive science, however, we do not in general know the costs of errors nor can we expect them to exactly follow the underlying probability distributions. For convenience we assume here that the odds are specified in percent rather than the usual reduced integer form, although we express them in reduced form in our spreadsheets.

While we have used a gambling rationale here, speculative investing fits exactly the same model and a basic principal of portfolio theory is to diversify in accordance with the perception of risk, with risk being used to estimate returns in technical analysis (as opposed to fundamental analysis based on external factors). On the other hand, failing to directly penalize errors – as with precision, recall, F-factor and conditional entropy – leads to arbitrage issues (in financial markets) or the possibility of a “Dutch book” (in gambling), violating the principal of rational choice.

This means that it is possible to adopt a strategy that guarantees winning some amount even in the absence of any “edge” or theory of the causal or historical factors involved.

As we illustrated with our POS example, this is the trick that enables speech recognizers and parsers to quote unrealistic error rates, recall and precision – if you are trying to decide between two spellings or two part of speech tags for a word, and always choose the more common one, you will get a higher precision than mere guessing. Viz. if X>Y and you choose X you will win X−Y>0 percent of the time.

We show in this paper that our Bookmaker evaluation technique assesses guessing (random choice) as giving us 0 gain, perfectly correct performance as giving us maximum gain and perfectly incorrect performance as giving us maximum loss. Moreover making a perfect correct decision *G%* of the time and guessing otherwise gains us *G%* of our maximum gain. Conversely making a perfectly incorrect decision *G%* of the time and guessing otherwise loses us *G%* of our maximum loss. Moreover this generalizes from the binary case to the *K* choice case where there are actually *K*−1 wrong labels for any case – noting that in this case the maximum loss will in general be less than the maximum gain as the penalty is different according to precisely which incorrect choice is chosen. Also we can recover independent gain/guess factors *G*(*l*) = *B*(*l*) for each decision label *l*.

Bookmaker thus provides us with a measure of *informedness* – what percentage of the time we are making an informed decision versus guessing. It also tells us when we use information to choose incorrectly: training to noise, superstitious learning or overtraining decreases *B*(*l*).

The Bookmaker measure has been implemented in Excel for the binary and ternary cases, and a general version for any number of cases has been implemented in Matlab and has been used to evaluate research results in information retrieval, EEG, vision processing and speech processing experiments. (Electronic form of Fig.1 is active spreadsheet. Matlab/Octave code is available from author on request.)

Analysis of the Binary Case

The paradigmatic use of odds is in horse racing where a Bookmaker offers you odds like 2:1 or 1:2 on a particular horse in a race that will in general have more than two horses. When she offers you odds of 2:1 (2/1) it means she thinks your horse is at least twice as likely to lose as to win, and so you will receive twice your bet plus your original stake if it wins. Odds of 1:2 (1/2, or 2:1 against) means that she thinks the odds are 2:1 against her and that your horse is twice as likely to win as lose so you will receive half your bet plus your original stake if it wins.

For the binary case we assume there are just two horses the probability of horse 1 winning (R+) is $W\%$ – so the odds we expect are $L{:}W$ where $L\% = 100\% - W\%$. On the track this would be reduced to a similar but smaller ratio involving smaller integers. We will also assume a fixed reduction factor k. If the odds are fair and we are guessing our expected gain/loss for guessing is $0.

We initially assume that we will always bet $$W$ on horse 1 if our system predicts +P, a win for horse 1 – and we win $$L$ if we are right. Otherwise (−P) we will bet $$L$ on horse 2 as the system predicts horse 2 will win and horse 1 will lose – and win $$W$ if we are right.

This means we have a payoff of $$L$ if we correctly bet on horse 1 and otherwise we lose our stake of $$W$. Conversely, we win $$W$ if we correctly bet on horse 2 and otherwise lose our stake of $$L$. In terms of our contingency table for the decision as to whether horse 1 will win or not, we have the following payoffs and each event has the same amount riding in the pool (we have anted up $$L$ and the bookie $$W$ or vice versa, which add absolutely to $100):

	+R	−R	
+P	$$L$	($$W$)	100
−P	($$L$)	$$W$	100

Our net actual winnings or losses will reflect the percentage of the time we are using *definitive* information to choose the *correct* horse (row) rather than just guessing:

$$
\begin{array}{llll}
 & B & = \sum_{l\in \mathbf{P}} \mathrm{p}_{\mathbf{P}}(l) \sum_{c\in \mathbf{R}} \mathrm{p}_{\mathbf{PR}}(l,c)\, \mathrm{w}(c|l), & \\
\text{where} & \mathrm{w}(c|l) & = +(1- \mathrm{p}_{\mathbf{R}}(c)) / k & (c = l), \\
 & & = -(1- \mathrm{p}_{\mathbf{R}}(\mathrm{c})) / k & (c \neq l).
\end{array}
$$

If negative this indicates the extent to which we are using available information to make an *incorrect* decision, rather than guessing. We will set k so that the expected payoff is $1 for correct bets, a loss of $1 for incorrect bets, and an expected $0 overall for guessing as we will verify below.

Given $k = \mathrm{p}_{\mathbf{R}}(c)\cdot(1- \mathrm{p}_{\mathbf{R}}(c))$ we find $\mathrm{w}(c|l) = \pm 1 / \mathrm{p}_{\mathbf{R}}(c)$ and we can see that we bet $1/$L$ to win $1/$W$ for horse 1, and $1/$W$ to win $1/$L$ for horse 2, consistent with their respective odds of $L{:}W$ and $W{:}L$. Since the probability of a win for horse 1 is W, expected winnings are $W \cdot \$1/W = 1$ for model one (perfect play) and similarly for horse 2 we expect $L \cdot \$1/L = 1$ when betting correctly with model one. Thus the model is set up so that with perfect play (model one) we stand to win $1 on a given horse on average.

This is illustrated in Figure 1 (but odds are shown in a reduced form that add to 10 rather than as percentages that add to 100) and follows directly from the linearity of the payoff formula. Figure 1 shows four distinct decision models. The first chooses randomly, the second makes perfect decisions, the third follows model two $G\%$ of the time (perfect play) and model one the rest of the time (random guess). The total percentage of cases in each cell in the decision matrix is the sum of the $G\%$ of the decisions that follow model two and the $100-G\%$ of the decisions that follow model one. Due to the linearity of the payoff calculation the same applies to the payoff matrix, so that $B = G\%$ – given guessing has an expected payoff of $0 whilst the perfect model has an expected payoff of $1 as ensured by setting k appropriately.

Since our bookmaker odds are defined so as to be zero sum, guessing gives no advantage to either party and the expected (long term average) gain is $0. The expected winnings on horse 1 is $+W \cdot \$1/W$ and the expected loss is $-L \cdot \$1/L$ (when we bet on 1 but 2 wins) which sum to $0

Thus the percentage of bets we win is reflected directly in the dollars we win and also directly reflects the proportion of the time, $G\%$, that we are betting in accordance with a perfect play model.

In model three the expected payoff for each decision we make is the same for each horse and reflects our making an informed correct decision $G\%$ of the time in each case, with each row of the payoff matrix showing the same profit margin of $G(\mathrm{l}) = B$, the average across all decisions. In the binary case, our formula simplifies to $B = \mathit{recall} - \mathit{fallout} = \mathit{tpr} - \mathit{fpr} = G(0) = \mathit{invrecall} - \mathit{invfallout} = \mathit{tnr} - \mathit{fnr} = G(1)$.

The fourth model used in Figure 1 makes an *incorrect* decision (labels reversed) $H\%$ of the time, showing a loss $B = -H\%$ (due to the symmetry of the binary case).

Note the similarity between the Bookmaker payoff formula and the conditional entropy formula that uses information as its currency, with $\mathrm{w}(c|l) = \log(1/\mathrm{p}_{\mathbf{R}}(c))/\mathrm{p}_{\mathbf{P}}(l)$. This weighting is however uniformly non-negative, so does not exhibit analogous properties.

Analysis of the General Case

The extension of the bookmaker evaluation formula to more than two choices, $C{=}K{>}2$, is complicated by the fact that there are multiple wrong choices, but according to the bookmaker-odds metaphor, the penalty for losing is independent of which other horse wins.

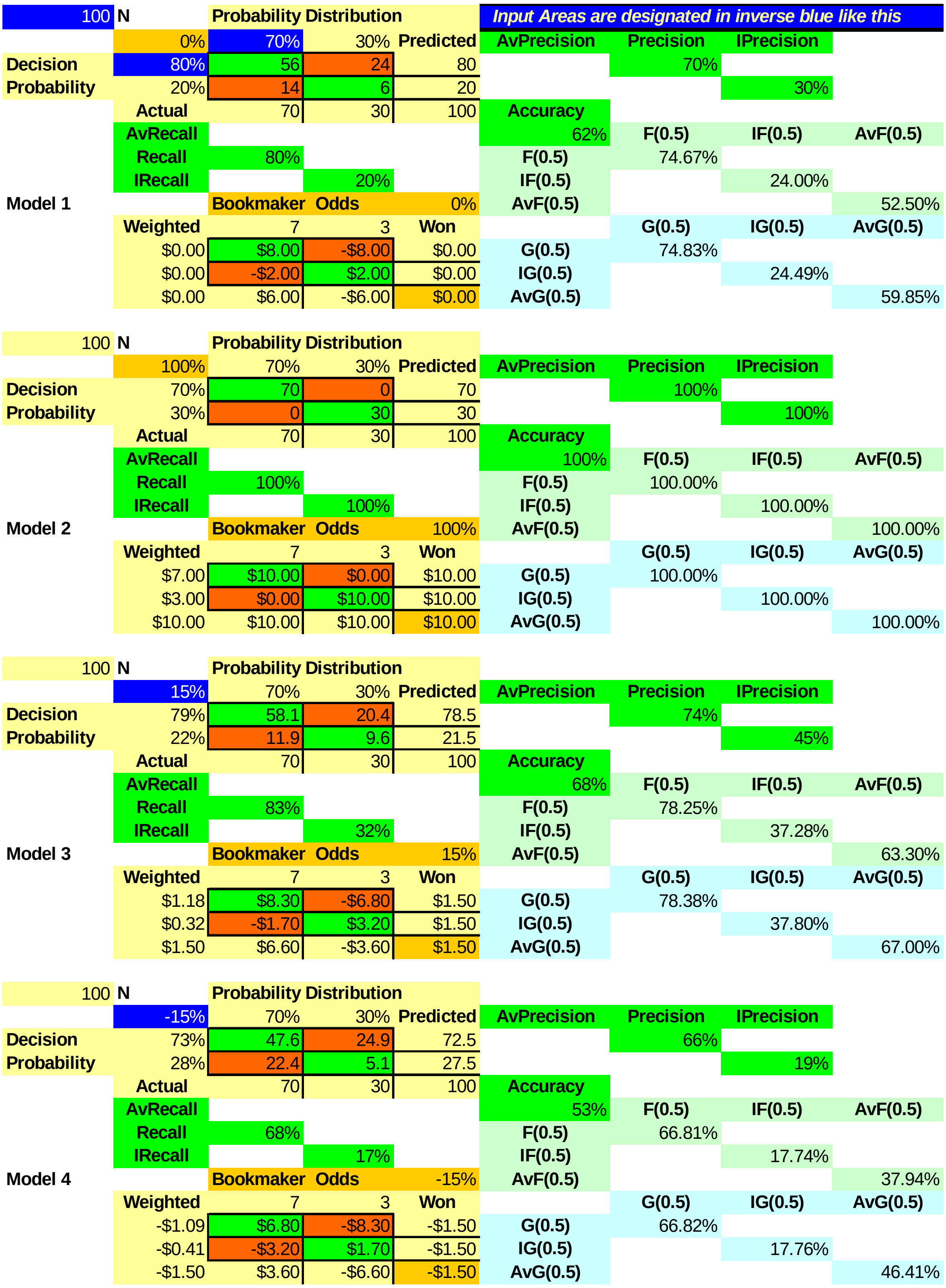
The generalized Bookmaker payoff formula is thus:

$$
\begin{array}{llll}
 & B & = \sum_{l\in \mathbf{P}} \mathrm{p}_{\mathbf{P}}(l) \sum_{c\in \mathbf{R}} \mathrm{p}_{\mathbf{PR}}(l,c)\, \mathrm{w}(c|l), & \\
\text{where} & \mathrm{w}(c|l) & = +1 / \mathrm{p}_{\mathbf{R}}(l) & (c = l), \\
 & & = -1 / (1 - \mathrm{p}_{\mathbf{R}}(l)) & (c \neq l).
\end{array}
$$

Note that this defines the same weighting for a binary decision as the previous formula, and in general a profit $B{\geq}0$ continues to estimate $G\%$, the percentage informed correct decision. This is because the penalty for making an incorrect decision is applied irrespective of which incorrect decision is made and fair bookmaker odds reflecting the distribution probabilities are designed to be zero sum – that is to say there is no advantage to either party from guessing or strategy. Any consistent gain $G\%$ is thus due to an edge – making good use of available information: *informedness*.

In the binary case, G was independent of the predicted label l so that $G(l) = \sum_{c\in \mathbf{R}} \mathrm{p}_{\mathbf{PR}}(l,c)\, \mathrm{w}(c|l)$ was constant independent of the chosen label. This is not necessarily going to be the case for C>2 as it may be that some classes are noise-affected or more confusable than others so we derive less information and depend more on chance in allocating these labels. We illustrate this later.

In the same way, a loss $B{<}0$ no longer directly estimates $H\%$, the percentage informed incorrect decision. This arises because we must make a further (random) choice different from the correct choice as determined by the informed model, and each of these possibilities has a lower (but non-zero) probability and so a different penalty for an incorrect



100	N	Probability Distribution			*Input Areas are designated in inverse blue like this*			
	0%	70%	30%	Predicted	AvPrecision	Precision	IPrecision	
Decision	80%	56	24	80		70%		
Probability	20%	14	6	20			30%	
	Actual	70	30	100	Accuracy			
	AvRecall				62%	F(0.5)	IF(0.5)	AvF(0.5)
	Recall	80%			F(0.5)	74.67%		
	IRecall		20%		IF(0.5)		24.00%	
Model 1		Bookmaker	Odds	0%	AvF(0.5)			52.50%
	Weighted	7	3	Won		G(0.5)	IG(0.5)	AvG(0.5)
	$0.00	$8.00	-$8.00	$0.00	G(0.5)	74.83%		
	$0.00	-$2.00	$2.00	$0.00	IG(0.5)		24.49%	
	$0.00	$6.00	-$6.00	$0.00	AvG(0.5)			59.85%

100	N	Probability Distribution						
	100%	70%	30%	Predicted	AvPrecision	Precision	IPrecision	
Decision	70%	70	0	70		100%		
Probability	30%	0	30	30			100%	
	Actual	70	30	100	Accuracy			
	AvRecall				100%	F(0.5)	IF(0.5)	AvF(0.5)
	Recall	100%			F(0.5)	100.00%		
	IRecall		100%		IF(0.5)		100.00%	
Model 2		Bookmaker	Odds	100%	AvF(0.5)			100.00%
	Weighted	7	3	Won		G(0.5)	IG(0.5)	AvG(0.5)
	$7.00	$10.00	$0.00	$10.00	G(0.5)	100.00%		
	$3.00	$0.00	$10.00	$10.00	IG(0.5)		100.00%	
	$10.00	$10.00	$10.00	$10.00	AvG(0.5)			100.00%

100	N	Probability Distribution						
	15%	70%	30%	Predicted	AvPrecision	Precision	IPrecision	
Decision	79%	58.1	20.4	78.5		74%		
Probability	22%	11.9	9.6	21.5			45%	
	Actual	70	30	100	Accuracy			
	AvRecall				68%	F(0.5)	IF(0.5)	AvF(0.5)
	Recall	83%			F(0.5)	78.25%		
	IRecall		32%		IF(0.5)		37.28%	
Model 3		Bookmaker	Odds	15%	AvF(0.5)			63.30%
	Weighted	7	3	Won		G(0.5)	IG(0.5)	AvG(0.5)
	$1.18	$8.30	-$6.80	$1.50	G(0.5)	78.38%		
	$0.32	-$1.70	$3.20	$1.50	IG(0.5)		37.80%	
	$1.50	$6.60	-$3.60	$1.50	AvG(0.5)			67.00%

100	N	Probability Distribution						
	-15%	70%	30%	Predicted	AvPrecision	Precision	IPrecision	
Decision	73%	47.6	24.9	72.5		66%		
Probability	28%	22.4	5.1	27.5			19%	
	Actual	70	30	100	Accuracy			
	AvRecall				53%	F(0.5)	IF(0.5)	AvF(0.5)
	Recall	68%			F(0.5)	66.81%		
	IRecall		17%		IF(0.5)		17.74%	
Model 4		Bookmaker	Odds	-15%	AvF(0.5)			37.94%
	Weighted	7	3	Won		G(0.5)	IG(0.5)	AvG(0.5)
	-$1.09	$6.80	-$8.30	-$1.50	G(0.5)	66.82%		
	-$0.41	-$3.20	$1.70	-$1.50	IG(0.5)		17.76%	
	-$1.50	$3.60	-$6.60	-$1.50	AvG(0.5)			46.41%

Figure 1: Spreadsheet showing 0%, 100%, +15% and - 15% chance models comparing Bookmaker Profit with Precision, Recall, Rand Accuracy, F-factor and unweighted Geometric Mean.
(Within Word double click to use spreadsheet, right click to import into Excel.)

Table 1: All 500 occurrence of *train* and *dog* in the Brown corpus, with sense tf and df counts.

A. 127:37 *dog* as animal	B. 2:2 negative *dog* metaphor	C. 2:2 top *dog*	D. 1:1 *dog*leg
E. 16:14 *dog*ma/*dog*matic	F. 1:1 *dog*trot	G. 4:3 hot *dog*	H. 9:1 *dog* as cam
I. 5:5 *dog*ged/follow *dog*gedly	J. 2:2 *dog* in plant name	K. 11:3 *dog* in place name	L. 1:1 *dog* in product name

M. 229:87 *train* as teach	N. 75:42 railway *train*	O. 8:4 wagon *train*	P. 5:1 *train* of dress	Q. 2:1 *train* as aim

choice. Their loss is maximized if they back the most probable incorrect horse, since the favorite has the least favorable odds and gives the most profit to the bookmaker when we lose, as we must.

Both positive and negative values of *B* have been seen in supervised learning systems. However, the latter is very rare, occurring only with very poor learning models that behave worse than chance.

In the case of unsupervised learning or classifications, a negative value tells us the classes have not been labeled optimally. In the binary case this only affects the sign, $G\% = -H\%$, whilst in the general case the classes should be permuted to maximize the sum of the diagonal of the contingency matrix, or simply to maximize *G%*.

If the number of classes found, *K*, does not equate with the number expected, C, then combinations rather than permutations need to be explored.

Selection, Abstention and an Example Evaluation

In real life, it is important to know when to stay out of the market or not to take a bet. Rather than guessing when you don't have the information to make a good decision, it is better to refrain from doing so. In a clustering context, often the largest cluster will be a cluster which the available attributes were simply not able to tease apart. In this situation it desirable to have $K>C$ and ignore such catch-all classes.

The Bookmaker may be used in this case too – simply apply as usual to the cases but assign zero weight (cost or penalty) to cases classified in an ignored class. Alternatively ignored classes and labels that are only present in an ignored class may be omitted and the Bookmaker calculation executed on a contingency matrix of the remaining *n* cases, matching the best class for each retained label. The probabilities found will then all be conditioned on the criterion for inclusion, but the true recall (*r*) and informedness (*B*) may be found by multiplying them by n/N where *N* is the original number of cases.

Like the cheat of assuming the most likely answer, this acknowledges that there are some things we aren’t good at. But Bookmaker won’t be influenced by that form of guessing or any other. Rather, each prediction *c* has a value $G(c)$ that indicates how informed the decision is, whilst $1-G(c)$ indicates how much pure guesswork is involved.

Table 1 shows 17 sense classes for the words *dog* and *train*, possibly followed by other letters. Exactly 500 such occurences were found in the 500 2000 word document extracts that constitute the Brown corpus. An additional index document (501) and class (Z) were added for technical reasons – this corresponds to the introduction of an index context (many documents returned by multiword web searches tend to have the form of a dictionary or index). 21 words related to the primary meanings of *dog/N*, *train/N*, and *train/V* (classes A, M and N) were used to provide context to cluster these 501 occurences using AutoClass (Cheeseman and Stutz, 1995).

A classification into 18 classes was reduced to a 12x12 contingency matrix with 170 cases after finding which induced class had the highest probability of predicting which label, combining equally good classes, and eliminating the catchall classes that did not predict any label best. Bookmaker, recall, precision, and means were calculated for the 12 classes along with a weighted average:

%	A	B	E	G	H	I	K	M	N	O	P	Z	wav
r	100	3	33	7	13	10	33	100	100	8	13	20	34
p	35	100	100	33	25	100	25	33	24	25	100	100	
B	100	-24	32	-2	5	4	32	100	100	1	-2	18	86
F	52	5	50	11	17	18	29	50	39	13	22	33	18
G	60	16	58	15	18	32	29	58	49	14	35	45	31

Note that the *F* and *G* class means lie between *p* and *r* but that *B* is strongly influenced by *r*. The average, B_{wav}, shows that where a classification is made it is an informed decision 86% of the time. Since only 170 out of 500 cases (34%) were classified, *B* (and *r*) must be discounted by 34% so that overall informedness is estimated at 0.86*0.34 = 29%.

The negative values of $B(l)$ indicate where superstitious labelings dominate and we are doing *worse* than chance so should discard the classes (and/or guess) to improve B_{wav}.

References

Cheeseman, P. and Stutz, J. (1996). Bayesian Classification (AutoClass): Theory and Results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth., and Uthurusamy, R. (Eds.) *Advances in Knowledge Discovery and Data Mining*. Cambridge, MA: MIT Press/AAAI Press.

Cover and Thomas, (1990). *Elements of Information Theory.* New York, NY: John Wiley & Sons.

Entwisle, J., & Powers, D. M. W. (1998). The Present Use of Statistics in the Evaluation of NLP Parsers, *Proceedings of the NeMLaP3/CoNLL98 Joint Conference* (pp. 215-224), Somerset, NJ: ACL.

Lafferty, J., McCallum, A. & Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the 18th International Conference on Machine Learning* (282-289), San Francisco, CA: Morgan Kaufmann.

Manning, C. D., & Schutze, H. (1999). *Foundations of statistical natural language processing.* Cambridge, MA: MIT Press.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association, 66*, 846-850.