

mental job expected of a loss function: they give maximum reward to predictors that are capable of predicting the true probabilities accurately. However, there are some objective differences between the two that may help you form an opinion.

The quadratic loss function takes account not only of the probability assigned to the event that actually occurred, but also the other probabilities. For example, in a four-class situation, suppose you assigned 40% to the class that actually came up and distributed the remainder among the other three classes. The quadratic loss will depend on how you distributed it because of the sum of the p_j^2 that occurs in the expression given earlier for the quadratic loss function. The loss will be smallest if the 60% was distributed evenly among the three classes: an uneven distribution will increase the sum of the squares. The informational loss function, on the other hand, depends solely on the probability assigned to the class that actually occurred. If you're gambling on a particular event coming up, and it does, who cares how you distributed the remainder of your money among the other events?

If you assign a very small probability to the class that actually occurs, the information loss function will penalize you massively. The maximum penalty, for a zero probability, is infinite. The gambling world penalizes mistakes like this harshly, too! The quadratic loss function, on the other hand, is milder, being bounded by

$$1 + \sum_j p_j^2,$$

which can never exceed 2.

Finally, proponents of the informational loss function point to a general theory of performance assessment in learning called the *minimum description length (MDL) principle*. They argue that the size of the structures that a scheme learns can be measured in bits of information, and if the same units are used to measure the loss, the two can be combined in useful and powerful ways. We return to this in Section 5.9.

5.7 Counting the cost

The evaluations that have been discussed so far do not take into account the cost of making wrong decisions, wrong classifications. Optimizing classification rate without considering the cost of the errors often leads to strange results. In one case, machine learning was being used to determine the exact day that each cow in a dairy herd was in estrus, or “in heat.” Cows were identified by electronic ear tags, and various attributes were used such as milk volume and chemical composition (recorded automatically by a high-tech milking machine), and milking order—for cows are regular beasts and generally arrive in the milking

shed in the same order, except in unusual circumstances such as estrus. In a modern dairy operation it's important to know when a cow is ready: animals are fertilized by artificial insemination and missing a cycle will delay calving unnecessarily, causing complications down the line. In early experiments, machine learning methods stubbornly predicted that each cow was *never* in estrus. Like humans, cows have a menstrual cycle of approximately 30 days, so this “null” rule is correct about 97% of the time—an impressive degree of accuracy in any agricultural domain! What was wanted, of course, were rules that predicted the “in estrus” situation more accurately than the “not in estrus” one: the costs of the two kinds of error were different. Evaluation by classification accuracy tacitly assumes equal error costs.

Other examples in which errors cost different amounts include loan decisions: the cost of lending to a defaulter is far greater than the lost-business cost of refusing a loan to a nondefaulter. And oil-slick detection: the cost of failing to detect an environment-threatening real slick is far greater than the cost of a false alarm. And load forecasting: the cost of gearing up electricity generators for a storm that doesn't hit is far less than the cost of being caught completely unprepared. And diagnosis: the cost of misidentifying problems with a machine that turns out to be free of faults is less than the cost of overlooking problems with one that is about to fail. And promotional mailing: the cost of sending junk mail to a household that doesn't respond is far less than the lost-business cost of not sending it to a household that would have responded. Why—these are all the examples of Chapter 1! In truth, you'd be hard pressed to find an application in which the costs of different kinds of error were the same.

In the two-class case with classes *yes* and *no*, lend or not lend, mark a suspicious patch as an oil slick or not, and so on, a single prediction has the four different possible outcomes shown in Table 5.3. The *true positives* (TP) and *true negatives* (TN) are correct classifications. A *false positive* (FP) occurs when the outcome is incorrectly predicted as *yes* (or positive) when it is actually *no* (negative). A *false negative* (FN) occurs when the outcome is incorrectly predicted as negative when it is actually positive. The *true positive rate* is TP divided

Table 5.3 Different outcomes of a two-class prediction.

		Predicted class	
		yes	no
Actual class	yes	true positive	false negative
	no	false positive	true negative

by the total number of positives, which is $TP + FN$; the *false positive rate* is FP divided by the total number of negatives, $FP + TN$. The overall success rate is the number of correct classifications divided by the total number of classifications:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Finally, the error rate is one minus this.

In a multiclass prediction, the result on a test set is often displayed as a two-dimensional *confusion matrix* with a row and column for each class. Each matrix element shows the number of test examples for which the actual class is the row and the predicted class is the column. Good results correspond to large numbers down the main diagonal and small, ideally zero, off-diagonal elements. Table 5.4(a) shows a numeric example with three classes. In this case the test set has 200 instances (the sum of the nine numbers in the matrix), and $88 + 40 + 12 = 140$ of them are predicted correctly, so the success rate is 70%.

But is this a fair measure of overall success? How many agreements would you expect *by chance*? This predictor predicts a total of 120 *a*'s, 60 *b*'s, and 20 *c*'s; what if you had a random predictor that predicted the same total numbers of the three classes? The answer is shown in Table 5.4(b). Its first row divides the 100 *a*'s in the test set into these overall proportions, and the second and third rows do the same thing for the other two classes. Of course, the row and column totals for this matrix are the same as before—the number of instances hasn't changed, and we have ensured that the random predictor predicts the same number of *a*'s, *b*'s, and *c*'s as the actual predictor.

This random predictor gets $60 + 18 + 4 = 82$ instances correct. A measure called the *Kappa statistic* takes this expected figure into account by deducting it from the predictor's successes and expressing the result as a proportion of the total for a perfect predictor, to yield $140 - 82 = 58$ extra successes out

Table 5.4 Different outcomes of a three-class prediction: (a) actual and (b) expected.

		Predicted class						Predicted class			
		a	b	c	Total			a	b	c	Total
Actual class	a	88	10	2	100	Actual class	a	60	30	10	100
	b	14	40	6	60		b	36	18	6	60
	c	18	10	12	40		c	24	12	4	40
	Total	120	60	20			Total	120	60	20	
(a)						(b)					

of a possible total of $200 - 82 = 118$, or 49.2%. The maximum value of Kappa is 100%, and the expected value for a random predictor with the same column totals is zero. In summary, the Kappa statistic is used to measure the agreement between predicted and observed categorizations of a dataset, while correcting for agreement that occurs by chance. However, like the plain success rate, it does not take costs into account.

Cost-sensitive classification

If the costs are known, they can be incorporated into a financial analysis of the decision-making process. In the two-class case, in which the confusion matrix is like that of Table 5.3, the two kinds of error—false positives and false negatives—will have different costs; likewise, the two types of correct classification may have different benefits. In the two-class case, costs can be summarized in the form of a 2×2 matrix in which the diagonal elements represent the two types of correct classification and the off-diagonal elements represent the two types of error. In the multiclass case this generalizes to a square matrix whose size is the number of classes, and again the diagonal elements represent the cost of correct classification. Table 5.5(a) and (b) shows default cost matrixes for the two- and three-class cases whose values simply give the number of errors: misclassification costs are all 1.

Taking the cost matrix into account replaces the success rate by the average cost (or, thinking more positively, profit) per decision. Although we will not do so here, a complete financial analysis of the decision-making process might also take into account the cost of using the machine learning tool—including the cost of gathering the training data—and the cost of using the model, or decision structure, that it produces—that is, the cost of determining the attributes for the test instances. If all costs are known, and the projected number of the

Table 5.5 Default cost matrixes: (a) a two-class case and (b) a three-class case.

		Predicted class				Predicted class		
		yes	no			a	b	c
Actual class	yes	0	1	Actual class	a	0	1	1
	no	1	0		b	1	0	1
(a)				(b)				
				c	1	1	0	