Confusion Matrix

Alex Sverdlov
alex@theparticle.com

1 Introduction

We wish to automate a decision making process. Decisions often lead to actions. Actions have costs (also known as utility, or economic gain/loss). Bad decisions lead to incorrect actions, which often have much greater costs. Obviously what we want to do is make decisions that minimize expected costs.

2 Confusion Matrix

A classifier assigns measurement instances to one of $c_1, \ldots, c_N$ classes. A confusion matrix is a probability table:

		Assigned				
		c_1	$\cdots$	c_j	$\cdots$	c_N
True	c_1	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	c_i	$\cdots$	$\cdots$	$P_{TA}(c_i, c_j)$	$\cdots$	$\cdots$
	$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	c_N	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Such a table can generally be estimated by running the classifier on a test dataset. The diagonal elements represent correct classification. We can get a probability of correct assignment by summing the diagonal elements:

$$P_{correct} = \sum_{i=1}^{N} P_{TA}(c_i, c_i)$$

3 Economic Gain

A measure of correctness isn't very useful without some measure of how costly the errors are. Similar to the confusion matrix, there's also the economic gain matrix, which assigns a value to each pair:

True \ Assigned	c_1	$\cdots$	c_j	$\cdots$	c_N
c_1	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
c_i	$\cdots$	$\cdots$	$e(c_i, c_j)$	$\cdots$	$\cdots$
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
c_N	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

The value can be positive or negative. Using the economic gain matrix along with the confusion matrix we can determine the expected economic gain across all pairs:

$$E_{gain} = \sum_{i=1}^{N} \sum_{j=1}^{N} e(c_i, c_j) P_{TA}(c_i, c_j)$$

This gets us a single number for the decision rule.

4 False Positives/Negatives

If there is a target category that we are particularly interested in then terminology changes a bit. For example, if we're trying to detect fraud: c_1, vs normal activity c_2, then we get:

True \ Assigned	c_1	c_2
c_1	true positive	false negative
c_2	false positive	true negative

5 Decision Rules

We can picture a decision rule as a table that maps inputs to outputs:

Input \ Output	c_1	$\cdots$	c_j	$\cdots$	c_N
d_1	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
d_i	0	0	1	0	0
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
d_M	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

The above decision rule maps d_i to c_j.

A bit more abstract, we can flatten this table. Since all the entries are 0 or 1 we end up with a tuple of $N \times M$ bits. Meaning a decision rule is a corner point on a unit hyper-cube, of $N \times M$ dimensions.

Any such point represents a decision rule and a linear combination of these decision rules is also a decision rule (a stochastic one).

6 Receiver Operating Characteristic Curve

TODO