Precision and recall

John Goldsmith

June 26, 2015

Document retrieval

Precision

How well do the documents that your system gives you actually satisfy what you are looking for?

Recall

How sure are you that you got back all of the documents you really wanted?

Document retrieval

Precision

$$\frac{\#(\text{appropriate documents returned})}{\#(\text{documents returned})}$$

Recall

$$\frac{\#(\text{appropriate documents returned})}{\#(\text{appropriate documents})}$$

Precision and recall

These terms have become the standard expectation of how a method is evaluated.

Precision and recall trade-off

Precision and recall trade-off

You can always get 100% precision, and you can always get 100% recall, but the cost is almost always too great, in both cases.

One solution: use the F-score: the reciprocal of the average of the reciprocals. $2 \times \frac{precision \times recall}{\text{precision + recall}}$.

$$\frac{1}{\frac{1}{2}(\frac{1}{a} + \frac{1}{b})} = \frac{1}{\frac{1}{2}(\frac{a+b}{ab})} = \frac{2ab}{a+b}$$

Precision and recall trade-off

Or you can give a chart of various precision/recall trade-offs produced by adjusting parameters of the algorithm.

Precision and recall

Gold standard:

	Yes	No	
Test says yes:	True positives	False positives	Precision
Test says no:	False negatives	True negatives	
	Recall		

More than one possible test: 1

Task: Find morphemes

Your algorithm wants to find morphemes (=word parts): anti-alias-ing

Measurement: find breaks

One way to measure this is by predicting which positions mark breaks: Gold standard truth is 0,4,9,14. Then antialias-ing is 0,9,14. Precision is $\frac{3}{4}$ and recall is $\frac{3}{4}$.

Baseline

Baseline

What is the precision and recall of a clever but useless algorithm: e.g., mark morphemes boundaries before the first and after the last letter?

Baseline

A clever but useless algorithm defines our *baseline*. Hopefully we have nowhere to go than up from there (though that is not guaranteed!).

Possible test 2:

Discover a list of morphemes

Suppose our goal is to “pullout” the morphemes of the language. Then if *ed* or *ing* is found in *any* word, that counts as 1 true positive.

If the algorithm cuts: *jump-ed walk-ed mov-e-d lov-ed raise-d* and the gold standard says *jump walk move love raise ed*, then there are 4 true positives (jump, walk, raise, ed) and 2 false negatives (move, love) (because they were *not found* by the algorithm), and 3 false positives (e,d, lov)(because they were found but they should not have been found).
Precision: 4 out of (4 + 3) = 0.571; recall is 4 out of (4 + 2) = 0.667.