Week 6 Precision & Recall

In many real-world situations, accuracy is not an adequate measure of the quality of the model. Instead, precision and recall can be used. The example in the lecture involves automating the choice of positive reviews (sentences from positive reviews) to show on a website. The process of choosing and posting the review sentences is intended to be completely automated – in this case we want something we know will not deliver even a single bad result. So accuracy of the classifier is not the most important metric for this purpose; even if the classifier is 90% accurate, one out of 10 sentences may not be what we want.

What is good performance for a classifier?

Recall that a for a binary classifier and uniform data, a random guess would yield 50% accuracy. So we definitely want to do better than that.

What if our classifier has 90% accuracy? That sounds good, however, not all data is uniform. If 90% of reviews are negative, then a classifier that guessed negative 100% of the time would achieve a 90% accuracy. This is a problem with unbalanced data. In this particular example, we would never find a positive review, which is what we are trying to do in our scenario.

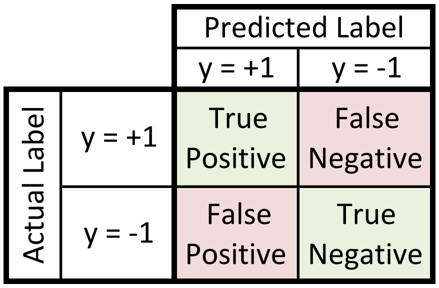
In our scenario, where we want to automate finding positive reviews and posting them to the restaurant's website, precision and recall are more useful. We never want to make a mistake and post a negative review on our own website – so precision is important – how well we do at eliminating errors from our results. Also, the kinds of sentences we are looking for may be rare, so we don't want to miss them – this is recall – the measure of how well we do at finding all examples of a category.

Precision- Fraction of positive predictions that are actually positive

Precision is generally measured as the fraction of positive predictions that are actually positive. So we if made 6 predictions but only 4 of those were really positive, then the precision would be 0.67. So precision is measuring the fraction of true positives.

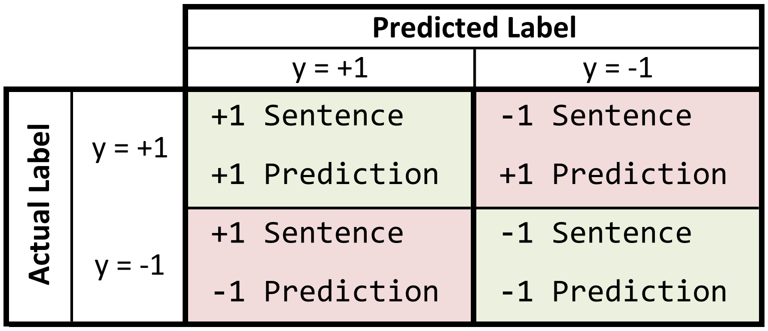
$$precision = \frac{(\hat{y} = +1) == (y = +1)}{(\hat{y} = +1)} = \frac{true\ positives}{positive\ predictions}$$

We can look at this with a truth table. We can see there are two kinds of errors; false positive and false negatives.



		Predicted Label	
		y = +1	y = -1
Actual Label	y = +1	True Positive	False Negative
	y = -1	False Positive	True Negative

You can see that if we predict a sentence to be +1, but the sentence is actually labeled -1, then this is a false positive – we incorrectly predicted positive. This kind of error is very bad for our scenario because false positives (actually negatives) could end up on our website because we predict it is positive,



		Predicted Label	
		y = +1	y = -1
Actual Label	y = +1	+1 Sentence +1 Prediction	-1 Sentence +1 Prediction
	y = -1	+1 Sentence -1 Prediction	-1 Sentence -1 Prediction

Recall - Fraction of positive data predicted to be positive

Recall is the fraction of positive data that we actually predicted as positive. It is the subset of truly positive data that we predict as positive.

$$recall = \frac{(\hat{y} = +1) == (y = +1)}{y = +1} = \frac{true\ postives}{actual\ positives}$$

So

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Precision-recall extremes

An overly optimistic classifier, one that classifies most data as positive, will have very high recall. At the extreme, a classifier that predicts all positives will have perfect recall because all labeled positive data will be predicted as positive.

However, such a classifier has low precision, because it predicts many false positives.

An overly pessimistic classifier, one that only predicts positive if it is very very sure, will have high precision. However, because they predict large numbers of false negatives, they will have low recall. In the extreme, in no positive predictions are made, then recall is 0.

In practice we want to balance precision and recall. In our scenario where we want to automatically post positive reviews, we want to emphasize precision, but we don't want to miss good reviews either.

Generally;

- Pessimistic Classifiers have higher precision and lower recall.
- Optimistic Classifiers have higher recall and lower precision.

Of course, a perfect classifier that never gets anything wrong has precision of 1.0 and recall of 1.0, but that is not realistic using real-world data.

Trading off precision and recall

It turns out to be very easy to transition between optimistic and pessimistic models. This is done using the prediction probability that the classification algorithms produce. If $P(\hat{y} = +1)$ is very high, then we have high confidence that the positive prediction is correct. If $P(\hat{y} = +1)$ is close to 0.5, then we have low confidence in our classification prediction.

In our prior work in logistic regression, we used the probability that the data is classified as positive to make the prediction;

$$\hat{y} = \begin{cases} +1 \; if \; P(\hat{y} = +1) > 0.5 \\ -1 \; if \; P(\hat{y} = +1) \leq 0.5 \end{cases}$$

So we used as a threshold the probability of 0.5 to make the decision about the classification. We can simply choose a threshold other that 0.5 to make the tradeoff between precision and recall.

Remember that a very pessimistic classifier will tend to predict fewer positives (lower recall) but have high confidence in them so they are more likely to be true positives (high precision). We could create a very pessimistic classifier by making our threshold 0.999;

$$\hat{y} = \begin{cases} +1 \; if \; P(\hat{y} = +1) > 0.999 \\ -1 \; if \; P(\hat{y} = +1) \leq 0.999 \end{cases}$$

Now, only points with probability > 0.999 will be classified as +1, everything else will be classified as -1.

We could make a very optimistic classifier by making our threshold 0.001;

$$\hat{y} = \begin{cases} +1 \; if \; P(\hat{y} = +1) > 0.001 \\ -1 \; if \; P(\hat{y} = +1) \leq 0.001 \end{cases}$$

Now any point with a probability of being positive that is > 0.001 will be classified as positive; so most points will be classified as positive. That means we will likely capture most of the actual positives (high recall), but have many false negatives (low precision).

So we can navigate between optimistic and pessimistic classifiers using the probability threshold t.

$$\hat{y} = \begin{cases} +1 \; if \; P(\hat{y} = +1) > t \\ -1 \; if \; P(\hat{y} = +1) \leq t \end{cases}$$

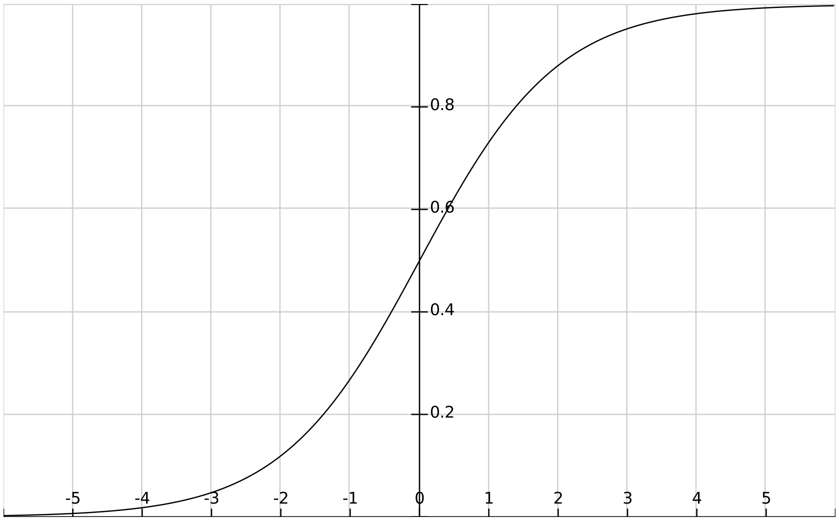
t is a probability that ranges from 0 to 1.

Remember that for logistic regression, the score is given by;

$$score = w^T h(x)$$

We then use the sigmoid function to map these score values to between 0 and 1;

$$probability = \frac{1}{1 + e^{-w^T h(x)}}$$

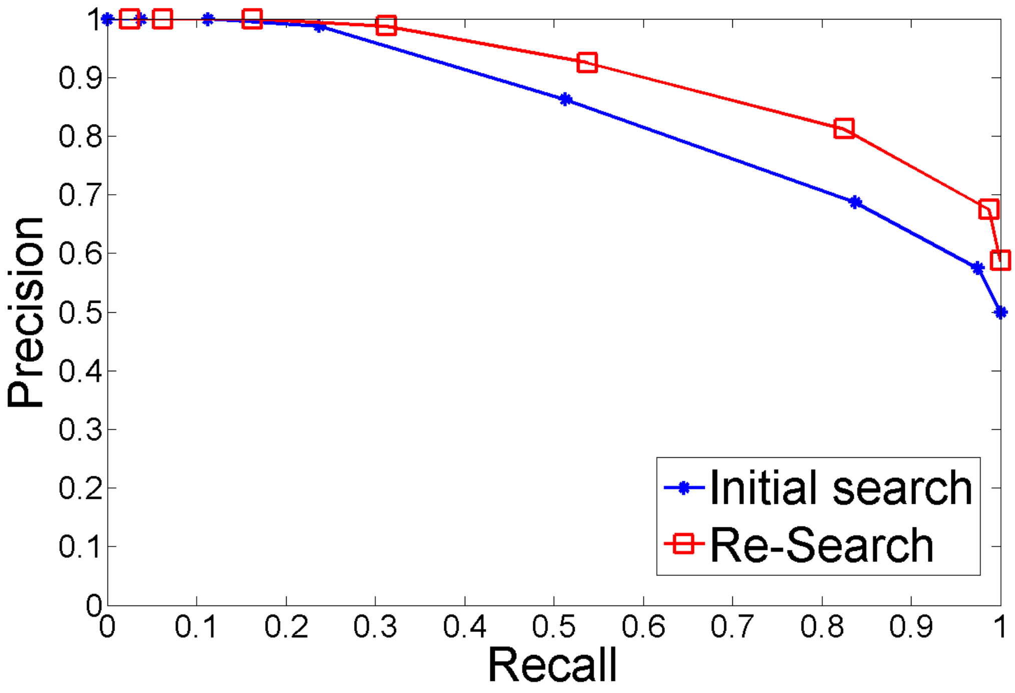


Probability vs score

Precision-recall curve

A precision-recall curve is a plot of precision (y-axis) vs recall (x-axis) with each point calculated at a given probability threshold. This can be used to compare classifiers. Given two classifiers, always choose the classifier that has better precision for the same recall. You can see this in the precision-recall curve; the more desirable classifier will have a curve above the less desirable (higher precision for the same recall).

For instance, here are precision-recall curves for an image matching classifier (see http://www.cs.cmu.edu/~hebert/indexing.html for the research)



We can see that the Re-Search classifier has the same or higher precision for all recall values.

It is not always this clear. In some cases, the precision-recall curves for two classifiers may cross. This creates regions in which one classifier is better (has higher precision for the same recall) than the other.

It is common to use area-under-the-curve (AUC) measures to choose one classifier over another. Area under the curve calculates a measure for a range of t, so that we can pick a classifier that works best over a range.

The F1 measure is also used to make decisions between classifiers based on the precision-recall curve.

Comparing Classifiers – Precision at k

Another measure is Precision-at-k. Rather than a range, this measures the precision of the top k most probably positive points. So if we make predictions for our data and sort by probability, then we choose the top k predictions. The precision-at-k is then the fraction of those that are actually positive.

So in our scenario, if we want to show 5 reviews on our site, we would pick k=5. We can then measure our classifiers against this by making predictions on training data, then sorting by probability, then choosing the top 5. If 4 of 5 of these are actually positive, then our classifier has precision-at-k of 0.8. Of course, in our application we want this to be closer to 1, so we might increase the probability threshold of the classifier to increase precision.