Lecture notes: ROC evaluation

Evaluation: How should the results of an algorithm designed to detect an object be reported? For example, suppose we want to determine if each pixel is a blood vessel pixel, yes or no? Or suppose we take pictures of cars coming off the assembly line, and want to ask whether or not each car has 4 tires on it?

Ground truth: The first thing we need is the expected (correct) answer. In medical science, this is often called the "gold standard", which is generally considered to be the best treatment currently known. In imaging, it is called ground truth, from the experience of verifying what has been seen in aerial or satellite imagery. To do that an operative would go in "on the ground" to a remote or hostile area, and verify (create "ground truth") of what was seen in the imagery. The phrase has stuck in image processing and computer vision and is now often used to refer to the desired/actual answer.

In the case of blood vessel segmentation, an opthalmologist could provide us a ground truth labeling for each pixel that we assume is correct. The system response for each pixel can then be compared to the ground truth to evaluate system performance. In the case of car manufacturing, a person could observe a recording of 100 cars coming off the line and write down the correct answer. The computer vision system could then be tested on the recording, verifying its answers against the ground truth.

A **truth table** is defined as follows:

system response \ ground truth	yes	no
yes	true positive (TP)	false positive (FP)
no	false negative (FN)	true negative (TN)

TP and TN are correct performance, FP and FN are errors.

The true positive rate (TPR) and false positive rate (FPR) are given as

$$TPR = \frac{TP}{GT = yes} = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{GT = no} = \frac{FP}{FP + TN}$$

For example, if the opthalmologist says there are 10,000 total blood vessel pixels, and the system performs with TP=9,000 and FN=0, then the TPR=90%. If the ophthalmologist says there are 100,000 total non-blood vessel pixels, and the system performs with FP=2,000 and TN=0, then the FPR=2%.

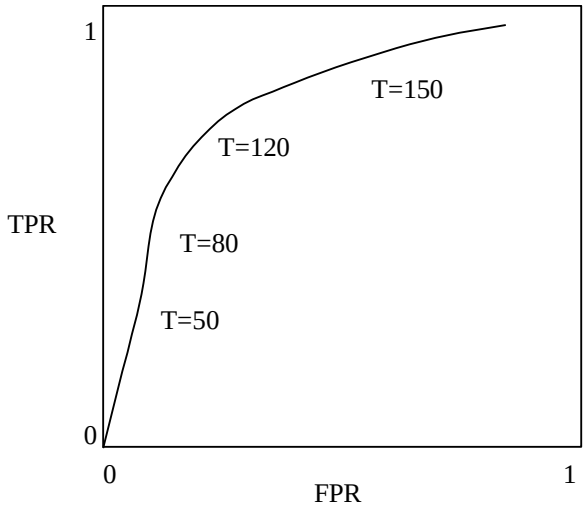
In the medical domain, the terms **sensitivity** and **specificity** are often used in place of TPR and FPR. Sensitivity is the same calculation as TPR, and specificity is the complement of FPR (specificity = 1-FPR = TN/(FP+TN)). This is used when the focus is on not only detecting true occurrences of a problem, but also making sure that when the problem is not there (i.e. no tumor), there is no operation (a true negative).

For some computer vision problems, the concept of a true negative is difficult to define or does not exist. For example, suppose we were tasked with detecting when a person takes a bite (places food into the mouth) during eating. Imagine watching a video and being asked to indicate the times when a person has taken bites. These are the "GT=yes". The computer vision system can similarly indicate specific times for the "system response=yes". However, all the other times are implicitly "GT=no" and "system response=no". It is not difficult to define FP's and TN's, as these are times where the GT or system said "yes" but the other said "no". However, it is impossible (or impractical) to define the intersection of them both saying "no" (true negatives) since there are infinitely many such times. For problems like this one, it is better to use the **positive predictive value** (PPV) as an indicator of how often the system reports a false positive.

$$PPV = \frac{FP}{system = yes} = \frac{FP}{TP + FP}$$

Varying performance. Most computer vision systems have thresholds and parameters that can vary their performance. In such cases, it can be useful to look at the TPR and FPR as the thresholds and parameters are changed. One can seek to determine the best possible values for the thresholds and parameters by finding a particular TPR vs FPR. This is done by plotting an ROC.

A **Receiver Operating Characteristic (ROC)** curve plots the TP vs FP as a function of a given variable. For example, with a matched filter, the system responds differently depending on how the output (match) threshold is selected. In general, more TP will be found as the threshold is decreased, but more FP will also be found. An ROC curve plots the TP vs FP as a function of the match threshold. For example:



Where is perfect performance? (At the top left, where FPR=0 and TPR=1.)

How is a "final" threshold selected? According to the application, depending on tolerances for the TPR and FPR. For example, for systems that detect medical problems we are usually willing to live with a high FPR to get as high a TPR as possible. On the other hand, some systems cannot survive FPs and live with as high a TPR as possible where FPR=0. In the average case, the threshold can be selected at the "**knee**" of the ROC curve for "**best tradeoff**" performance.

Cross confusion can happen when a system is tasked to identify multiple types of objects, instead of just a single yes/no. For example, suppose a computer vision system is tasked to identify different types of cars coming off the assembly line. If it mistakenly classifies a sedan as a truck, this is cross confusion.

A **cross confusion matrix** is a way of tabulating how often different things are misclassified as other things. For example, consider a system that classifies 5 different letters (i, l, m, n and x). Sometimes it may confuse one letter for another. A confusion matrix could be written as:

	ground truth					
system response		i	l	m	n	x
	i	80%	11%	1%	3%	1%
	l	15%	85%	1%	3%	2%
	m	2%	1%	85%	12%	0%
	n	2%	1%	12%	82%	2%
	x	1%	2%	1%	0%	95%

The number in each box indicates how often each letter was classified as the given letter by the system. Diagonal entries indicate correct classification, off-diagonal entries indicate errors. The relative size of the confusion entries indicates that i and l were often confused with each other, as were m and n.