# 语义分割任务的评估指标

## 度量标准

**为何需要语义分割系统的评价标准？**

* 为了衡量分割系统的作用及贡献，其性能需要经过严格评估。并且，评估须使用标准、公认的方法以保证公平性。
* 系统的多个方面需要被测试以评估其有效性，包括：执行时间、内存占用、和精确度。
* 由于系统所处背景及测试目的的不同，某些标准可能要比其他标准更加重要，例如，对于实时系统可以损失精确度以提高运算速度。而对于一种特定的方法，尽量提高所有的度量性能是必须的。

### 执行时间

### 速度或运行时间是一个非常有价值的度量，因为大多数系统需要保证推理时间可以满足硬实时的需求。某些情况下，知晓系统的训练时间是非常有用的，但是这通常不是非常明显，除非其特别慢。在某种意义上说，提供方法的确切时间可能不是非常有意义，因为执行时间非常依赖硬件设备及后台实现，致使一些比较是无用的。

### 然而，出于重用和帮助后继研究人员的目的，提供系统运行的硬件的大致描述及执行时间是有用的。这可以帮助他人评估方法的有效性，及在保证相同环境测试最快的执行方法。

### 1.2 内存占用

内存是分割方法的另一个重要的因素。尽管相比执行时间其限制较松，内存可以较为灵活地获得，但其仍然是一个约束因素。在某些情况下，如片上操作系统及机器人平台，其内存资源相比高性能服务器并不宽裕。即使是加速深度网络的高端图形处理单元（GPU），内存资源也相对有限。以此来看，在运行时间相同的情况下，记录系统运行状态下内存占用的极值和均值是及其有价值的。

### 1.3 精确度

图像分割中通常使用许多标准来衡量算法的精度。这些标准通常是像素精度及IoU的变种，以下我们将会介绍常用的几种逐像素标记的精度标准。为了便于解释，假设如下：共有k+1个类（从到，其中包含一个空类或背景），表示本属于类i但被预测为类j的像素数量。即，表示真正的数量，而、则分别被解释为假正和假负，尽管两者都是假正与假负之和。

* Pixel Accuracy(PA，像素精度)：这是简单的度量，为标记正确的像素站总像素的比例。



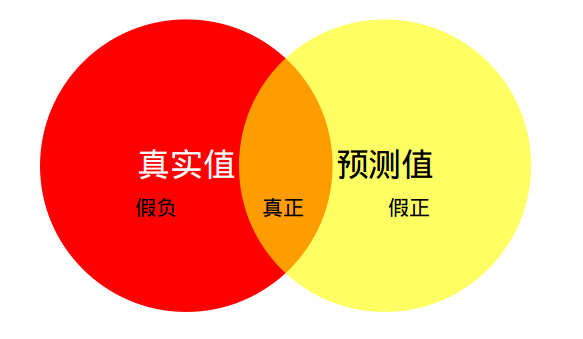
* Mean Pixel Accuracy(MPA，均像素精度)：是PA的一种简单提升，计算每个类内被正确分类像素的比例，之后求所有类的平均。，
* Mean Intersection over Union(MIoU，均交并比)：为语义分割的标准度量。其计算两个集合的交集和并集之比，在语义分割问题中，这两个集合为真实值（ground truth）和预测值（predicted segmentation）。这个比例可以变形为正真数（intersection）比上真正、假负、假正（并集）之和。在每个类上计算IoU，之后平均。



## 直观理解

如下图所示，红色圆代表真实值，黄色圆代表预测值。橙色部分红色圆与黄色圆的交集，即真正（预测为1，真实值为1）的部分，红色部分表示假负（预测为0，真实为1）的部分，黄色表示假正（预测为1，真实为0）的部分，两个圆之外的白色区域表示真负（预测为0，真实值为0）的部分。

* MP计算橙色与（橙色与红色）的比例。
* MIoU计算的是计算A与B的交集（橙色部分）与A与B的并集（红色+橙色+黄色）之间的比例，在理想状态下A与B重合，两者比例为1



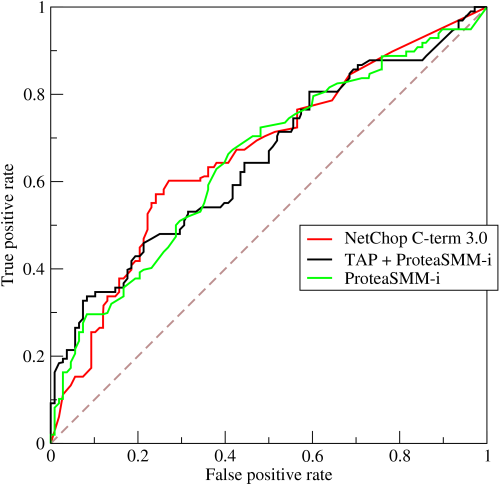
# ROC与AUC

http://alexkong.net/2013/06/introduction-to-auc-and-roc/#fnref:1

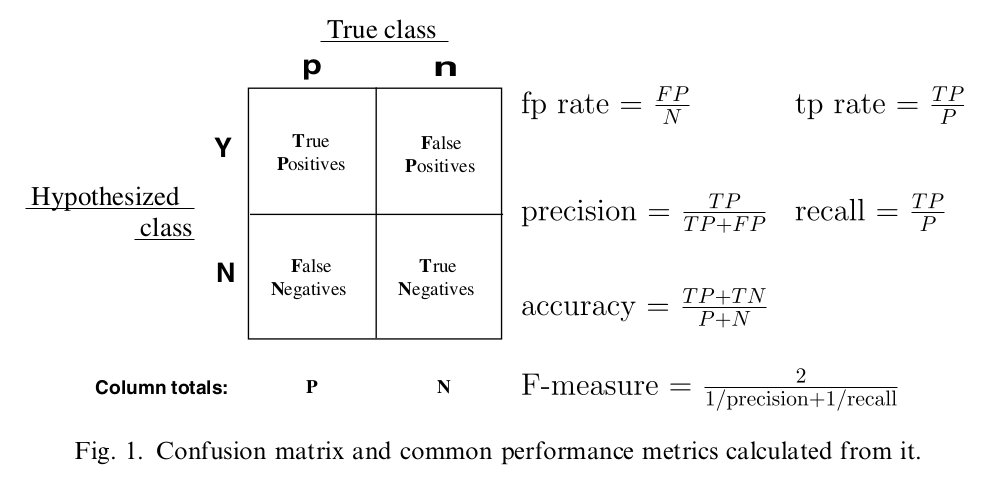
ROC（Receiver Operating Characteristic）曲线和AUC常被用来评价一个二值分类器（binary classifier）的优劣。

## ROC

需要提前说明的是，我们这里只讨论二值分类器。对于分类器，或者说分类算法，评价指标主要有precision，recall，F-score，以及我们今天要讨论的ROC和AUC。下图是一个ROC曲线的示例。



正如我们在这个ROC曲线的示例图中看到的那样，ROC曲线的横坐标为false positive rate（FPR），纵坐标为true positive rate（TPR）。下图中详细说明了FPR和TPR是如何定义的。



接下来我们考虑ROC曲线图中的四个点和一条线。第一个点，**(0,1)**，即**FPR=0, TPR=1**，这意味着FN（false negative）=0，并且FP（false positive）=0。Wow，这是一个完美的分类器，它将所有的样本都正确分类。第二个点，(**1,0)**，即**FPR=1，TPR=0**，类似地分析可以发现这是一个最糟糕的分类器，因为它成功避开了所有的正确答案。第三个点，**(0,0)**，即**FPR=TPR=0**，即FP（false positive）=TP（true positive）=0，可以发现该分类器预测所有的样本**都为负样本**（negative）。类似的，第四个点**（1,1）**，分类器实际上预测所有的样本**都为正样本**。经过以上的分析，我们可以断言，**ROC曲线越接近左上角，该分类器的性能越好。**

下面考虑ROC曲线图中的虚线y=x上的点。这条对角线上的点其实表示的是一个采用随机猜测策略的分类器的结果，例如(0.5,0.5)，表示该分类器随机对于一半的样本猜测其为正样本，另外一半的样本为负样本。

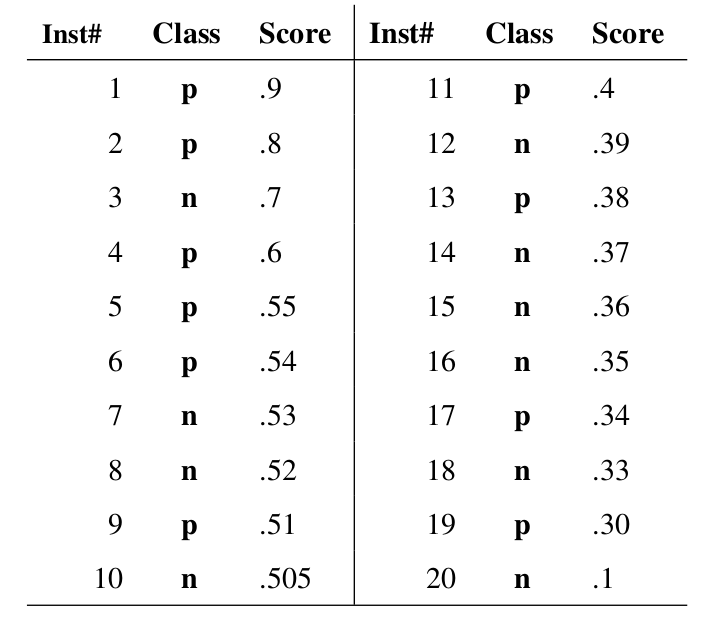
## 如何画ROC曲线

对于一个特定的分类器和测试数据集，显然只能得到一个分类结果，即一组FPR和TPR结果，而要得到一个曲线，我们实际上需要一系列FPR和TPR的值，这又是如何得到的呢？我们先来看一下[Wikipedia](http://en.wikipedia.org/wiki/Receiver_operating_characteristic)上对**ROC曲线的定义**：

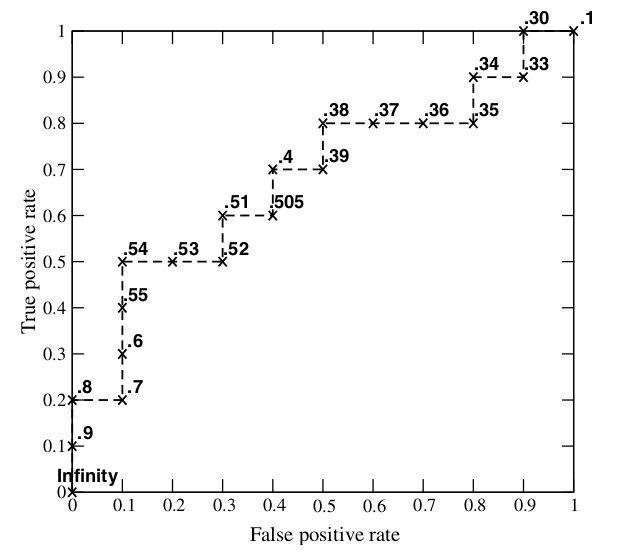
“In signal detection theory, a receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied.”

问题在于“as its discrimination threashold is varied”。如何理解这里的“discrimination threashold”呢？我们忽略了分类器的一个重要功能“**概率输出**”，即表示分类器认为某个样本具有多大的概率属于正样本（或负样本）。通过更深入地了解各个分类器的内部机理，我们总能想办法得到一种概率输出。通常来说，是将一个实数范围通过某个变换映射到(0,1)区间。

假如我们已经得到了所有样本的概率输出（属于正样本的概率），现在的问题是如何改变“discrimination threashold”？我们根据每个测试样本属于正样本的概率值从大到小排序。下图是一个示例，图中共有20个测试样本，“Class”一栏表示每个测试样本真正的标签（p表示正样本，n表示负样本），“Score”表示每个测试样本属于正样本的概率。



接下来，我们从高到低，依次将“Score”值作为阈值threshold，当测试样本属于正样本的概率大于或等于这个threshold时，我们认为它为正样本，否则为负样本。举例来说，对于图中的第4个样本，其“Score”值为0.6，那么样本1，2，3，4都被认为是正样本，因为它们的“Score”值都大于等于0.6，而其他样本则都认为是负样本。每次选取一个不同的threshold，我们就可以得到一组FPR和TPR，即ROC曲线上的一点。这样一来，我们一共得到了20组FPR和TPR的值，将它们画在ROC曲线的结果如下图：



当我们将threshold设置为1和0时，分别可以得到ROC曲线上的(0,0)和(1,1)两个点。将这些(FPR,TPR)对连接起来，就得到了ROC曲线。当**threshold取值越多，ROC曲线越平滑**。

其实，我们并不一定要得到每个测试样本是正样本的概率值，只要得到这个分类器对该测试样本的“评分值”即可（评分值并不一定在(0,1)区间）。评分越高，表示分类器越肯定地认为这个测试样本是正样本，而且同时使用各个评分值作为threshold。我认为将评分值转化为概率更易于理解一些。

## AUC值的计算

AUC（Area Under Curve）被定义为ROC曲线下的面积，显然这个面积的数值不会大于1。又由于ROC曲线一般都处于y=x这条直线的上方，所以AUC的取值范围在0.5和1之间。使用AUC值作为评价标准是因为很多时候ROC曲线并不能清晰的说明哪个分类器的效果更好，而作为一个数值，对应AUC更大的分类器效果更好。

在了解了ROC曲线的构造过程后，编写代码实现并不是一件困难的事情。相比自己编写代码，有时候阅读其他人的代码收获更多，当然过程也更痛苦些。在此推荐[scikit-learn](http://scikit-learn.org/stable/)中关于[计算AUC的代码](https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/metrics/metrics.py" \l "L479)。

## AUC意味着什么

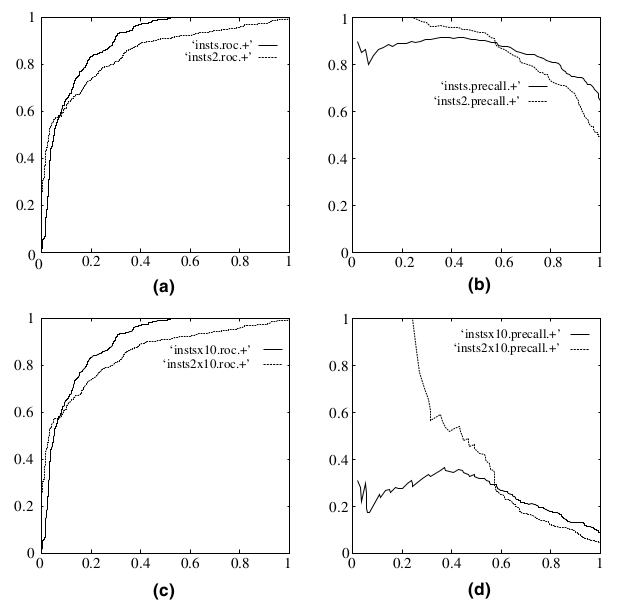
那么AUC值的含义是什么呢？AUC的值的含义是：

“The AUC value is equivalent to the probability that a randomly chosen positive example is ranked higher than a randomly chosen negative example.”

这句话有些绕，我尝试解释一下：首先AUC值是一个概率值，当你随机挑选一个正样本以及一个负样本，当前的分类算法根据计算得到的Score值将这个**正样本排在负样本前面的概率就是AUC值**。当然，**AUC值越大，当前的分类算法越有可能将正样本排在负样本前面，即能够更好的分类。**

## 为什么使用ROC曲线

既然已经这么多评价标准，为什么还要使用ROC和AUC呢？因为ROC曲线有个很好的特性：当测试集中的正负样本的分布变化的时候，ROC曲线能够保持不变。在实际的数据集中经常会出现类不平衡（class imbalance）现象，即负样本比正样本多很多（或者相反），而且测试数据中的正负样本的分布也可能随着时间变化。下图是ROC曲线和Precision-Recall曲线[5](http://alexkong.net/2013/06/introduction-to-auc-and-roc/" \l "fn:5)的对比：



在上图中，(a)和(c)为ROC曲线，(b)和(d)为Precision-Recall曲线。(a)和(b)展示的是分类其在原始测试集（正负样本分布平衡）的结果，(c)和(d)是将测试集中负样本的数量增加到原来的10倍后，分类器的结果。可以明显的看出，ROC曲线基本保持原貌，而Precision-Recall曲线则变化较大。