



链滴

机器学习 (1) ——几个基本要素

作者: [tionch](#)

原文链接: <https://ld246.com/article/1546960555728>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

学习算法

什么是学习算法，学习当然不是一个动词，学习算法最简单的理解便是能够从**数据学习**的算法，学习的解释根据 **Mitchell** 提出的定义“对于某类任务 **Task** 和性能度量 **Performance** ,如果一个计算机程序在任务中，性能能够随着经验 **Experience** 而自我完善，那么我们就称为**程序在经验中学习**。”
举一个非常简单的例子，加入你有一个女朋友，你们每次约好出去吃饭的时候，你总是兴冲冲的去等，但是每次都要等很久，你也不敢抱怨什么，但是也不想和个孤儿一样站着。于是你就开始了学习。的任务就是“缩小等待的时间”，然后你在约好时间后等待了五分钟再出门，结果你还是等了 15 分。下一次，你等待了十分钟再出门，你还是等了十分钟。后来你不断的调整等待的时间，这就是一个习的过程。那么有一次，你等了 25 分钟再出门，结果被臭骂了一顿，这也就是我们机器学习里面很常见的一个东西，叫做惩罚。当然学习也是有一个前提条件，就是任务必须要有**潜在的规律**，它不可以是一个无规律的事物。

学习任务

事实上很多人会有一种误解，认为我们学习的就是任务。请记住，我们学习的永远不是任务本，学习指的是**获得执行任务的能力**。比如，我们要实现人脸的自动识别，那么别就是任务，我们要获得识别的能力。说到底，机器学习就是试图让程序替代或者辅助人的智能行为

性能度量

我们在评估学习算法里面某项任务的好坏的时候，我们常常需要设计一些东西去量化它的性能比如人脸识别的任务重，我们需要衡量识别的正确率和准确率，通常我们称之为精度，当然还有错误。

为了衡量一个算法的好坏，我们常常需要人为的制作出数据，我们称为 **测试数据** (**test set of data**)，不过通常来说，测试数据也是已知数据的一部分，只是我们单独的划分开，只用于训练完成后的最终测试。那么平时我们进行参数的优化时所使用的据，我们称为**训练数据** (**train set of data**) 同时还有验证数据等等。

查全率与查准率

错误率和精度是最为常用的度量方式,但在一些任务中，比如分类任务，我们还需要一些额外的度量式。比如进行信息检索时，我们经常要关心分类中有多少是正确的以及分类正确的中有多少被选出来了这时候，我们引入**查准率** (**precison**)和**查全率** (**recall**)。

我们举一个二分类的问题，我们将分类器的预测结果分为一下四种结果：

(1) 真正例 (**True Positive,TP**) ,即预测为 1，真实列表为“1”的分类数据。

(2) 假正例 (**Flase Positive,FP**) ,即预测为 0，真实列表为“0”的分类数据。

(3) 真反例 (**True Negative,TN**) ,即预测为 0，真实列表为“0”的分类数据。

(4) 假反例 (**False Negative,FN**) ,即预测为 1，真实列表为“1”的分类数据。

总数据量=TP+FP+TN+FN，在这里，你可以列出一个淆矩阵方便你的阅读。

查准率 **P** 就是分类器预测结果为“1”中预测正确的比率。

$$P = \frac{TP}{TP + FP}$$

查全率 **R** 就是真实类标为“1”的数据中分类器预测正确的比率。

$$R = \frac{TP}{TP + FN}$$

通常来说，查全率和查准率是矛盾关系，当查准率很高的时候，查全率往往很低。比如讲一张片中的人全部查找出来，你可以将所有的动物都标记为人，那么你的查全率就是 1 了，但是查准率是很低。所以你需要有时候谨慎的选择，有时候宁可错杀一千也不放过一个，有时候你需要反其道行。

在评估多个算法的时候，我们可以作出 **P-R** 图，如果一个算法的 **P-R** 图将另一个算法的图完全包住，那么可以判定后者的性能更秀。如果它们发生了交叉，我们通常很难判定它们的性能，通常会进行比较 **BEP**

ong> (P=R 的点),不过 BEP 过于简陋了, 所以我们使用 F1 量:</p>
<div class="language-math">F1=\frac{2\times P\times R}{P+R}=\frac{2\times TP}{Sum+TP-T}</div>
<p><有的时候我们对查全率和查准率的要求不同, 比方说, 我们进行内容推荐的时候, 要尽可能降查全率提高查准率, 因为我们的要求是更少的打扰。我们使用 F1 度量的一般形式——F_{\beta}:</p>
<div class="language-math">F_{\beta} = \frac{(1+\{\beta^2\}) \times P \times R}{\{\beta\} \times \eta \times P + R}</div>
<p><其实我们可以发现, F1 是基于查准率和查全率的调和平均, 也就是\frac{1}{F1}=\frac{1}{2}(\frac{1}{P}+\frac{1}{R}),F_{\beta}则是广义的 F1, 也就是加权调和平均:</p>
<div class="language-math">\frac{1}{F_{\beta}}=\frac{1}{1+\beta^2}(\frac{1}{P}+\frac{\beta}{\eta \times R})</div>
<p><对于, 如果 \beta > 1, 展开表达式很容易发查全率会有更大的影响, 如果\beta < 1, 则查准率更影响。</p>
<h2 id="学习经验">学习经验</h2>
<p><在这里我会很粗略的讲一下监督学习(Supervised Learning)和非监督学习(Unsupervised Learning)。事实上机器学习的算法则是粗略的分为这两种。</p>
<监督学习算法: 简单地说就是我们试图将每一个数据都用一个相对应的类标(Label)进行关联, 就好比指着堆车子, 告诉你, 这种形状的车, 或者说, 以江西省为例, 所有籍贯江西的人的身份证号前两位都是 36, 这也是一个类标。我们一个小朋友去学习认识这些, 每次错误的时候, 我们相应的提醒纠正他, 他慢慢的就会调整自己, 不学习提高自己的识别能力。你可以把监督学习认为是一种映射关系, 对于每一个数据 x, 都会有一个对应的 y。

<非监督学习算法: 现实生活中, 大部分数据是没有标记的, 即使我们人为的它添加上标记, 这个工作量基本上是无法估量的。非监督学习就是在没有标记的情况下, 学习数据内自有的结构和规律。最常使用的非监督学习算法就是聚类, 例如我们听到了两毫无关联的声音, 我们在此之前也从来没有听过。但是我们能很清楚的分辨出这两种声音, 尽管我们不知晓这两种声音是什么。但是我们的任务只是将他们以最大的共同点进行分类就行了。

<监督学习费时费力, 半监督学习能力有限, 那么有没有什么是可以取到两种算法的优点呢? 这个时, 我们提出了半监督学习。

<半监督学习很好理解, 就是我们在面对大量无标记数据或者只有少部分有标记数据时, 我们将两种据一起运用上, 这就是半监督学习。

<不过对于一个算法的学习来说, 即使我们只标记一部分数据, 这样的工作量也是很大的。并且存在个问题, 很多标记使用人为的方式并不好, 标记通常都从环境中获取。那么我们怎么评判这些数据的记是否合适呢? 我们只需要提供某些评价机制(奖励和惩罚)。这一种学习方式我们称为化学习。不知道读者中养狗的多不多, 这种机制类似我们教狗上厕所, 小狗可能不懂你的思, 但是它乱撒尿我们打它一顿, 当它前往指定位置的撒尿我们奖励它一个食物。慢慢的它自然就懂。</p>
</blockquote>
<p>我的掘金:WarrenRyan</p>
<p>我的简书:WarrenRyan</p>
<p>欢迎关注我的博客获得第一时间更新 https://blog.tity.xyz</p>
<p>我的 Github:StevenEco</p>
</blockquote>