准确测量机器学习模型的误差

作者 张洋 | 发布于 2013-08-16

机器学习 数据挖掘 翻译 误差分析

原文：Accurately Measuring Model Prediction Error

在机器学习模型的效果评估中，预测误差的分析是重中之重。对于现有的各种误差测量技术，如果使用不当，会得出极具误导性的结论。这些结论会误导模型设计者设计出过拟合的模型，过拟合是指训练出的模型对于训练集拟合的很好，但是对于新的样本集则预测效果极差。这篇文章描述了如何正确的测量模型误差，以避免此类问题。

误差测量

对于一个预测模型来说，最重要的是要能对**新出现**的数据样本准确进行预测。所以在测量误差时，必须着重考虑这一点。但是实际中很多模型设计者往往用训练数据的误差而不是新数据的误差来评估模型。这种错误的误差测量方式往往是导致模型质量不高的根源。

一般来说，模型总是倾向于更好的拟合训练数据。一个模型对于新数据的误差期望总是高于在训练数据上的误差期望。打个比方，例如我们抽取100个人，通过回归模型来预测财富高低对幸福程度的影响。如果我们记下模型对于训练数据进行预测的平方误差（squared error）。然后我们将模型应用于100个新的人进行预测，模型对于新样本的平方误差一般会高于在训练数据上的平方误差。

下面我们通过公式来更明确的表述这一事实。我们可以建立模型对于新数据的预测误差（我们应该真正关心的指标，也叫实际误差）和模型对于训练数据的预测误差（被很多模型设计者误用的指标）之间的关系。

$$\text{实际误差} = \text{训练集误差} + \text{乐观率}$$

这里_乐观率_表示相对于训练数据来说，模型在新数据上的表现要糟糕多少。这个指标越高，就表明我们的模型对于训练数据的误差在实际误差中所占的比率越小。

过拟合风险

或许我们可以认为对于一组固定的训练集，乐观率是一个常数。如果这个假设正确，那么我们就可以通过最小化训练集误差来最小化实际误差。也就是说，虽然训练集误差过于乐观，但是如果将其最小化我们仍可以得到一个实际误差最小的模型。因此我们可以忽略实际误差与训练集误差的差异。

不幸的是上面只是我们一厢情愿的想法。实际情况上乐观率是模型复杂度的函数：随着模型复杂度的增加，乐观率也会随之上升。因此上面的公式可以重新写成：

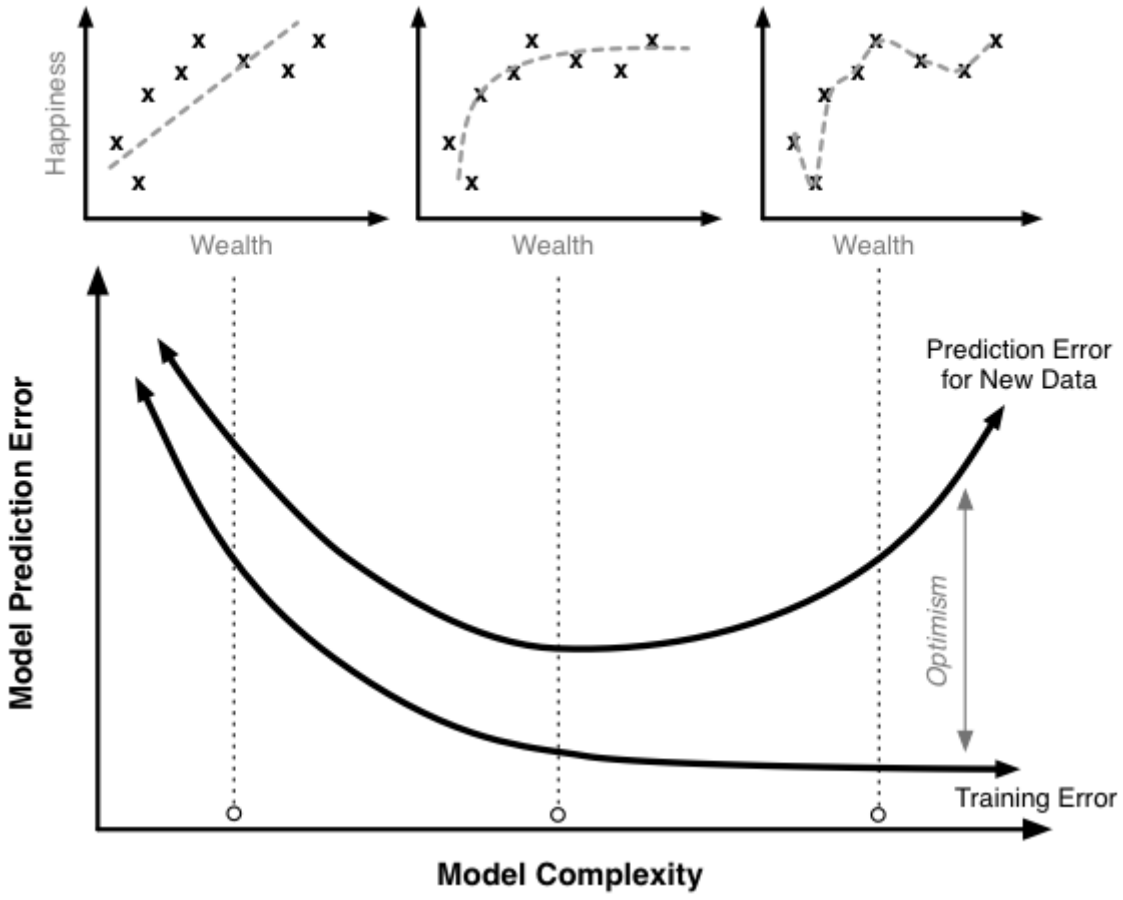
$$\text{实际误差} = \text{训练集误差} + f(\text{模型复杂度})$$

为什么会这样呢？因为随着模型复杂程度的提高（例如在线性回归中加入越来越多的参数）模型会更倾向于较好的拟合训练数据。这是统计学习模型的基本性质之一[1]。在我们的幸福指数预测模型中，如果我们将人的姓氏也当做一个特征，那么训练数据的误差相比之前就会下降。如果我们将一个已破产公司在1990年1月1日不同时段的股票价格作为特征加入，训练集误差也会下降。甚至我们通过掷骰子随机生成一列数据作为特征加入也可以降低训练集误差。无论增加的特征列与模型有没有关系，训练集误差总会随着特征的增加而下降。

同时，随着模型复杂度的增加，实际误差也会随之变化（我们实际关心的）。如果我们在幸福指数预测模型中加入了一个公司上世纪某一天的股票价格波动，那么可以预计模型的实际质量将降低。虽然股票价格可以_降低训练集误差_，但同时_拉高了模型对于新数据的预测误差_，因为加入股票价格特征后模型的稳定性变差了，因此在新数据上的预测表现随之下降。甚至就算你加入了一个与模型相关的预测变量，如果这个变量的信噪比较低，模型的实际误差依然会变大。

让我们通过一个实际的例子来直观感受一下上面的讨论。假设我们用线性回归模型来做幸福指数预测模型。刚开始我们可以使用最简单的模型：$Happiness = a + b\,Wealth + \epsilon$，然后我们逐渐增加多项式项，使得模型可以拟合非线性数据。每增加一项多项式项，模型复杂度也随之提高。通过这个过程，我们可以得到二次模型，如$Happiness = a + b\,Wealth + c\,Wealth^2 + \epsilon$，或者更复杂的高阶多项式模型：$Happiness = a + b\,Wealth + c\,Wealth^2 + d\,Wealth^3 + e\,Wealth^4 + f\,Wealth^5 + g\,Wealth^6 + \epsilon$。

下图展示了在不同复杂度下，模型的训练集误差、实际误差及乐观率间的关系。上面的散点图展示了不同复杂度下模型曲线对训练数据的拟合程度。



可见随着模型复杂度上升，训练集误差会随之下降。当模型复杂度很高时，我们的模型可以完美拟合训练集中的所有数据，训练集误差接近0。类似的，实际误差在开始的时候也是在下降。这是因为当模型没有引入高阶多项式项时看起来过于简单，不能很好的拟合训练数据。但是当复杂度到达一个临界点后，随着复杂度继续增加模型对训练数据的拟合越来越好，但是模型对于新数据的预测效果却在变差。

这种现象叫做_过拟合（overfitting）_。在这种情况下，模型过于关注那些训练数据的细节变动，而这些细节变动并不是所有数据共性的东西。从图中可以看出，此时曲线在尽力拟合每一个训练数据，这样的模型显然与训练数据太过紧密的。

避免过拟合现象是构建精准、鲁棒模型的关键之一。当只关注训练数据时，过拟合现象非常容易发生。为了检测是否存在过拟合，应该将模型应用于新数据上以检测效果。当然了，一般不可能得到真正的实际误差（除非你能得到数据空间的全部数据），但是有诸多方式可以帮助我们对实际误差进行准确估计。本文的第二章节将介绍一些相关的误差估计方法。

实例：不合理的误差测量导致的悲剧

我们通过一个常见的建模流程展示使用训练集误差作为实际误差所带来的陷阱[2]。我们首先随机生成100个样本数据。每个样本数据有一个目标字段和50个特征字段。例如，目标字段是一种树的生长速率，而特征字段包括降水量、湿度、气压、经纬度等等。在这个例子中，每个样本数据都是完全独立随机生成的，因此这份数据的字段间毫无关系。

然后我们建立一个线性回归模型来预测生长速率。因为我们知道特征和目标字段没有相关性，所以我们期望得到的结果是R^2为0。不幸的是我们的模型最后报告R^2为0.5。这不科学啊！我们的数据明明只是一些噪声数据。不过别急，我们还可以通过_F_检验来对模型进行确认。这个可以衡量模型的显著性，用以识别回归出的相关关系是不是只是因为偶然性得到的。_F_检验的_p_值为0.53，这表明回归模型不显著。

如果到此为止，一切看起来没有问题；我们应该丢弃这个模型，因为模型并不显著（当然了，这只不过是一些噪声数据！）。不过很多人通常接下来不会彻底丢弃这个模型，而是丢弃那些不显著的特征，然后保留相对显著的特征再次做回归。让我们假设留下显著性水平低于25%的特征，在这个例子中有21个。接着我们再次训练回归模型。

在第二次训练后，我们得到：

- R^2为0.36
- _p_值为$5 * 10^{-4}$
- 6个参数的显著性水平达到5%

再强调一下，我们的数据完全是噪声；不可能有任何相关性。但是我们第二次却得到了一个高显著性的模型，证据就是有意义的R^2值（在社会科学领域这个值相对较高）和6个显著的参数！

这是一个令人困惑的结果，我们建模的过程似乎并无不妥，但是却得到了一个匪夷所思的错误结论。这个例子展示了在统计过程中如果不能准确测量误差，则会得到具有严重缺陷的模型。

误差的测量方法

使用Adjusted R^2

R^2被广泛用于衡量模型的拟合程度。它的计算非常简单。首先对模型进行训练，然后计算每个训练数据实际值与预测值的差，将这些差的平方和相加，然后与_零模型_的预测误差平方和做对比。零模型用训练数据集目标字段的的平均值作为预测值。零模型可以看做最简单的预测模型，以此作为基准来评价其它模型的效果。其数学表示如下：

$$R^2 = 1 - \frac{\text{模型的误差平方和}}{\text{零模型的误差平方和}}$$

R^2的意义非常直观。如果模型的效果并不比零模型号多少，则R^2接近0，而如果我们的模型效果远好于零模型，则R^2接近1。R^2作为一个直观易于理解的评价指标，广泛用于各种回归模型的效果检测。

通常R^2是根据模型在训练集上的效果计算的。如前文所示，即使是高R^2数据本身可能也只是一堆噪声。实际上对于样本容量为n、参数为p的纯噪声数据，R^2的期望存在一个解析表达式：

$$E[R^2] = \frac{p}{n}$$

根据这个公式可以在具体情况下判定R^2是否有意义。例如上例中，我们的模型有50个参数和100个训练数据，R^2的期望为$50/100$，也就是0.5。

R^2有一个变种指标叫做Adjusted R^2，这个指标会对模型的复杂度做出惩罚。随着参数的增加，Adjusted R^2对在R^2的基础上变小。Adjusted R^2的公式为：

$$AdjustedR^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

标准的R^2会随着模型复杂度增加而变大，而adjusted R^2克服了这个缺点，因此我们应该总是使用adjusted R^2而不是R^2。当然adjusted R^2也不能完美的评估实际误差。实际上adjusted R^2一般对模型复杂度的惩罚力度会有所欠缺，所以如果模型足够复杂，adjusted R^2也会失效。

因此adjusted R^2也会出线过拟合现象。另外，adjusted R^2的许多假设在实际中也不一定成立。这也会导致adjusted R^2给出错误的结论。

优点

- 便于应用
- 内建于许多分析程序中
- 计算速度快
- 解释性好[3]

缺点

- 通用性不高
- 仍然有过拟合风险

信息论方法

有一些方法可以用于评估相对于真实模型来说我们的模型_丢失了多少信息_。当然我们是无法获知真实模型的（真正产生训练数据的实际模型），但是在一些前提下我们仍然有办法估计模型的信息丢失程度。信息丢失越多，则模型误差越高，效果越差。

信息论方法假设模型是一个参数模型（parametric model）。在这个前提下，我们可以根据参数和数据来定义训练数据的似然率（likelihood），不严格的说，似然率是指观测到的这组训练数据出现的概率[4]。如果我们调整参数使得这组数据的似然率最大，则得到这组参数的最大似然估计。于是我们就可以利用信息论的方法来比较不同模型和它们的复杂度，以此确定哪个模型最接近真实模型。

最常用的信息论方法是Akaike信息准则（Akaike's Information Criteria，简称AIC）。AIC被定义为一个关于模型似然率及模型参数的函数：

$$AIC = -2ln(Likelihood) + 2p$$

如同其它误差评价准则，我们的目标是最小化AIC。AIC的公式很简洁。第一部分（$-2ln(Likelihood)$）可以被视为训练集下的误差率，第二部分（$2p$）可视为对模型乐观性的惩罚。

除了AIC外还有许多基于信息论的判定准则。下面列举两个其它信息准则，与AIC相比其区别在于对乐观性的惩罚方式不同，下面两个准则对乐观性的惩罚还与样本容量n有关。

$$AICc = -2ln(Likelihood) + 2p + \frac{2p(p+1)}{n-p-1}$$

$$BIC = -2ln(Likelihood) + pln(n)$$

如何选择合适的信息准则是非常复杂的，涉及大量理论、实践甚至是哲学因素。实际中决定选用哪个准则要具体情况具体分析，甚至带有一定信仰成分。

优点

- 便于应用
- 内建于许多高级分析程序中

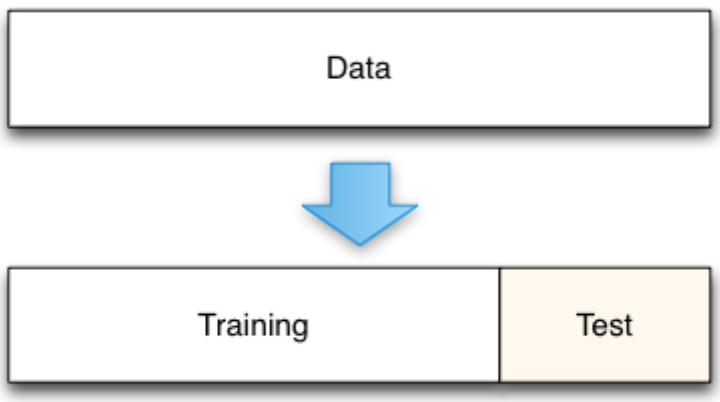
缺点

- 需要具体情况具体分析
- 需要模型能够计算似然率[5]
- 学术界对于这种方法的理论基础还存在诸多争议

测试集

上面提到的方法都只能用于参数模型，并且对模型有一些理论假设。如果这些假设不成立，则上面的方法效果将会很查。还好，实践中还有一些其它类型的方法，这些方法对模型没有任何假设，仅仅通过对数据集做处理来估计实际误差。

其中最简单的方法便是测试集方法。我们首先将样本数据集分为两份。一份用于模型训练；另一份用于效果评测。例如我们有1000个数据，我们可以用700个训练模型，剩下的300个评估模型。



这个方法可以说是测量模型误差的标准方法。模型实际误差被定义为模型对于新数据的预测误差。而通过预留测试集，我们可以直接测量这个误差。

测试集方法的代价是要减少一部分训练数据。例如上面我们从训练集中移除了30%的数据。这意味着相比于使用全量集合训练来说，我们的模型会存在更大的偏差。在标准的流程中，评价完模型效果后，我们会用全量数据重新训练来得到最终的模型。因此在这种情流程下，测试集的误差评价结果是偏_保守_的，因为模型的实际误差要比报告的误差低一些。在实际中这种保守的误差估计要比乐观的误差估计更有效。

这种技术的一个要点是在得到最终模型前不能以任何方式分析或使用测试集。一个常见错误是在效果评估后重新调整模型然后再次训练评估。如果在一次建模中你重复使用一份测试集，这份测试集就被污染了。由于测试集参与了模型调整，它就不能再给出模型误差的一个无偏估计了。

优点

- 对模型没有假设
- 数据足够多时，准确度较高
- 易于实现和使用
- 易于理解

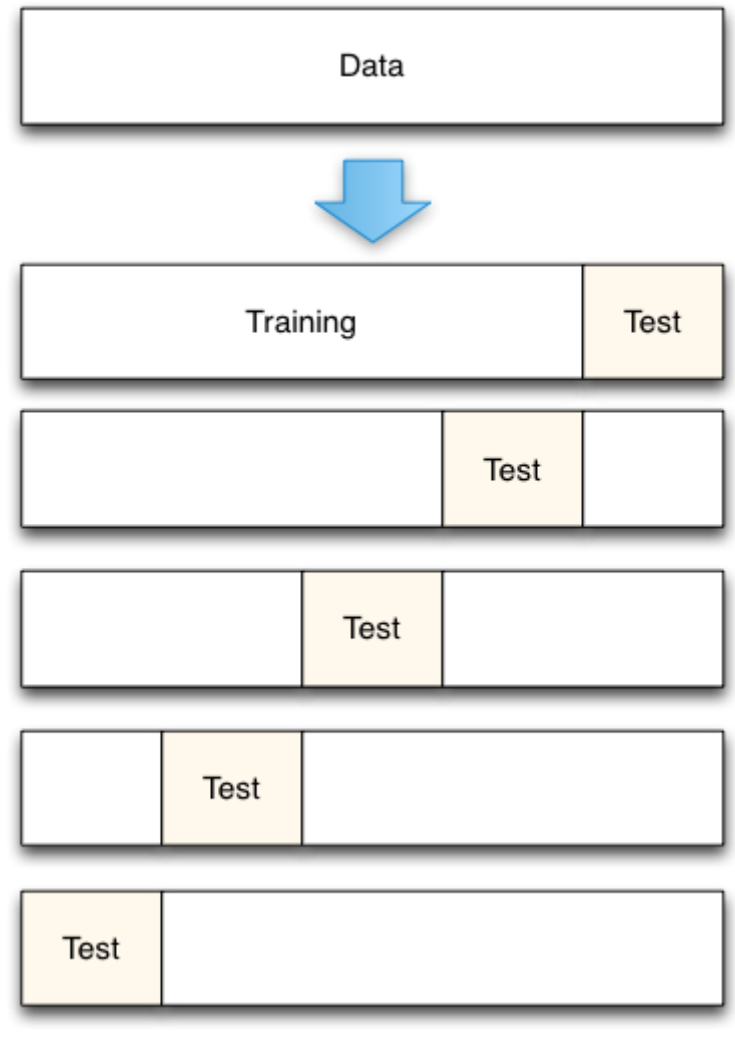
缺点

- 估计偏保守
- 一次使用即被污染
- 需要确定测试集比例（一般在70%-30%之间）

交叉验证及重新取样

有时，对于模型训练来说保留一部分数据作为测试集的方式有些代价过高。这时一些基于重新取样的方法如交叉验证就比较有用了。

交叉验证将数据集平均分成n份。例如我们将100个数据分成5份，每份20个数据点。然后我们重复做5轮误差测量。在每轮中，取其中4份（共80个数据点）训练模型，剩下的1份检验模型。然后将5轮测得的误差取平均值，最为对实际误差的估计。



可以看到，交叉验证与测试集方法很类似。不同之处在于交叉验证中每个数据既参与模型训练又参与模型检测，只不过不在同一轮里。当数据集不是很大时，交叉验证比测试集方法要更值得推荐一些，因为交叉验证不需要移除训练数据。交叉验证同时还能给出误差估计的稳定性度量，这是一个非常有用的指标。不过如果主要目标是衡量估计的稳定性，一些其它的重新取样方法如Bootstrapping更值得一试。

交叉验证的一个最大问题是确定分组数。一般来说，分组数越小则估计偏差越大（往往偏保守，也就是报告的误差比实际误差要大）但是方差越小。极端情况下，你可以每一个样本点分一个组，这叫做Leave-One-Out-Cross-Validation。此时对误差的估计基本没有偏差，但是方差会很大。理解偏差-方差权衡对于确定分组数是非常重要的。另一个需要关注的点是计算效率。对于每一个分组，你都要训练一个新的模型，所以如果训练过程比较慢的话，还是分少点组为好。最后说一下，根据经验一般把分组数定为5或10是比较合适的选择。

优点

- 对模型没有假设
- 数据足够多时，准确度较高
- 易于理解

缺点

- 计算效率低
- 需要确定分组数
- 估计偏保守

做出选择

总结一下，我们一共讨论了下列测量模型误差的技术：

- Adjusted R^2
- 信息论方法
- 测试集方法
- 交叉验证及重新取样

作为模型设计者，首先要决定是否依赖前面两个方法对模型的假设条件。如果不是的话，则可以选择后面两个模型。

一般来说，基于假设条件的模型更便于使用，不过选择这种易用性的同时要付出一些代价。首先就是，对于实际情况来说这些假设都不是完全成立的。至于是否近似成立要具体情况具体分析。很多时候这些假设基本是成立的，不过一旦实际情况与假设出入较大，那么这些方法所得出的结论就完全不可信了。

就我个人的经验来说，我更偏好交叉验证。因为交叉验证不需要对模型的假设，而且估计效果较好。对于交叉验证来说最主要的消耗是计算资源，不过随着现在计算机计算能力越来越强，这一点可以不用过多担心。对于需要假设的模型来说，虽然实际中很多模型都是参数模型，但是并

没有一个有效的方法去判断模型是否符合假设。因此使用这些方法时心里总是存在一点疑虑。而交叉验证虽然计算资源消耗多一点，但是其结论总是更让人放心。

Footnote

1. 仅对于损失函数是凸的（没有局部最大值和最小值）统计模型来说是这样。如果损失函数存在局部最大值或最小值，增加参数会令模型无法收敛到全局最优值，从而导致训练集误差也会变大。不过对于一些常见的模型（如线性回归及逻辑回归）其损失函数都是凸函数。
2. 这个例子取自Freedman, L. S., & Pee, D. (1989). Return to a note on screening regression equations. The American Statistician, 43(4), 279-282。
3. 虽然adjusted R^2与R^2是不同的统计量，不过两者有类似的直观解释。但是与标准R^2相比，adjusted R^2可以是负数（表示这个模型比零模型效果更差）。
4. 这个定义是不严格的，因为对于连续随机变量，获得这组数据的概率为0。如果让你从0到1之间随机取一个数，则你取到0.724027299329434...的几率为0。你无法准确写出这个数因为其小数部分是无穷的。似然率是通过让模型的概率密度函数取特定值计算出来的。要获得真正的概率，你需要对概率密度函数在一个区间上求积分。因为似然率不是一个概率值，所以它可能大于1。尽管如此，将似然率看做“给定数据集出现的概率”对于直观理解其意义是有帮助的；不过心里要清楚意识到，这在数学上是不准确的！
5. 这一点限制了信息论方法的适用范围，诸如随机森林与人工神经网络等模型均无法应用此方法。