常见数据挖掘错误：识别及纠正

Identifying and Overcoming Common Data Mining Mistakes

（Doug Wielenga, SAS Institute Inc., Cary, NC）

2007-11

摘要

由于通常涉及海量数据，数据挖掘分析会激化不少常见建模问题，且不断涌现新问题。这些问题会大大增加开发有效模型的时间，并阻碍更高层次模型的开发。

本文讨论了如何识别并纠正若干常见建模错误。本报告先从数据准备中的几个常见问题出发进行深入探讨，接着，按照一个典型预测建模分析的数据流程，探讨如何借助SAS Enterprise Miner来进行设定变量角色、创建及使用数据分割(Data Partition)[1]、进行变量选择、替换缺失值、构建不同类型模型、比较结果模型，以及如何进行模型打分。本文还总结了聚类分析和关联/序列分析中常见问题的讨论。使用这些技巧，可以大大减少构建有效模型的时间，并提高模型的质量。

前言

我们将识别出几种常见的数据挖掘错误，并提供纠正这些错误的建议。这里罗列的错误与纠正措施不应看作是完整的列表，要对它们完全展开讨论，说上几天几夜都说不完。不过，这里罗列的的确是经常出现的错误，而且它们也是经常可以克服的。同时请注意一下，选择何种最佳方式是高度主观性的，很有可能本文推荐的建议并不太适合某种特定场合。毕竟，为具体分析选择最适当方法是分析员的职责，这里的讨论是要让大家对那些导致非预期结果的情况增加认识，并阐述如何应对那些情况的方式。

1 数据准备

数据准备所花费的时间通常比花在数据分析上的时间要多。但是不幸的是，由于项目期限的原因，人们总是希望通过捷径方式尽快找到答案，而不管答案是否是最优的。选择捷径的结果往往是减少在数据准备阶段上的时间，然而数据如果没有被充分准备好，会增加花费在数据分析阶段上的时间，反而削弱了捷径方法的效果。更重要的是，在这种情况下生成的模型与在充分数据准备下建立的数据模型进行比较，它导致的后果包括：变量考虑不足，不正确地处理分类型变量，以及不正确的处理数值型变量。

1.1 变量考虑不足

当需要分析全部数据的时候，人们要考虑采用哪些变量来建立模型。这种思维方法带来一些问题：人们常常以已知数据子集的特性作为参考来解决建模的问题。

我们来考虑一下几点：

1. 不是所有公司都有同样的变量。除了那些相似的变量外，每个公司都会找到很多其他公司没采用的变量。不同的定义和度量方法导致了不同的变量。
2. 削减建模过程，使用通用变量子集来代替，这样做会忽略公司数据的丰富性和独特性。如果所有的公司都使用同样的变量来建模，那么就有可能错过发现一个适合本公司的独特的模式。根据自己公司的数据来进行建模可以使你选择最适合自己公司状况的方法。
3. 对公司来说，充分利用数据建立模型的好处莫过于提高市场的份额。当你的竞争对手通

[1] 译注：Data partition是数据分割，指把样本数据分为训练、验证和测试数据三部分，有时也只分成前面两部分

过交叉销售来提高业绩时，你的竞争对手就总会赢得市场份额。

采用这种“一招鲜”的方式在大多数业务环境下没什么意义，对于建模当然也没什么意义。通过对很多大型和中型公司的相似问题建模，我发现从不同数据源经常能得到迥异的信息。分析数据的过程可以大大提升对企业客户的洞察。有时候，这些发现证实了通常的认识，也有些时候，这些常识会部分甚至全部被驳斥。这个过程还有个额外好处，就是帮你发现数据库的错误。

为了克服这个问题，在时间允许的条件下，尽可能利用更多的数据进行建模。在时间资源有限的条件下，只使用一部分数据进行建模也是很平常的事情。为了建模所花的时间对于未来模型分析阶段是很有用的。以前的建模经验对于分析人员来说，可以帮助他们判别关键变量和很多没有实际作用的变量。总的来说，建模时花费的时间比在将来修复模型所花费的时间要少的多。在模型重建的过程中，不可避免的会花费很多时间来重新考虑变量，从中发现有用的变量。同时当内部和外部环境变化时，分析人员也会经常重新考虑那些被筛掉的变量。

1.2 分类型数据处理不当

大多数不好的模型的问题根源来自于没有很好的处理分类型变量。对分类变量处理方法不当有三种类型：包含了太多的分类层次（Level）；包含很少有值的分类层次；或者包含一类数据，这类数据占整体数据的很大比例。

1.2.1 包含太多的分类值

分类变量包含过多的分类值往往会导致性能问题。这就是为什么大部分建模过程缓慢甚至中断的原因。最好的办法就是对分类变量进行评估。一般来说，一个 k 层的分类变量在模型中需要 k-1 个参数来表示，而一个连续性变量只需要一个参数来表示，所以一个 k 层分类变量需要 k-1 个连续性变量对应。参数越多需要的数据量也越大，对性能的影响也越来越大。

为了解决这个问题，回答这些问题：

- 能否用一组具有少量值的变量来代替这个变量?
- 用多层的变量进行建模是否合适?

考虑这样的场景：在模型中使用邮政编码（zip code）。Zip code 是一个包含多层次的结构，我们从 Zip code 的用途进行考虑，考虑客户的地域、人口统计、经济状态，用这些指标来代替 zip code。或者使用 MSA（都市统计区）、STATE 把 zip code 分成组，用更少的取值和更细的粒度来代替 zip code。使用这些方法可以在建模的时候避免用在一个类别数据中用多层次结构来表示。

1.2.2 某些分类层次观测值过少

这个问题不至于过于影响性能，但也是低效率的潜在因素。在一个分类变量中，少部分分类层次对应了大量的数据，而其他绝大部分层次却对应了很少的数据，在这部分层次上，只有很少的观测值，对模型几乎没有什么实际意义。

我们以股东大会为例，每个股东根据自己持股比例进行投票。其中 5 个股东持有 96%的股票，其他 1000 个股东持有余下 4%的股票。在统计投票结果的过程中，把注意力放在每个股东身上会浪费很多时间和精力，因为投票的结果很大程度上取决于那 5 个股东。那些小股东在总体上对投票结果有影响，但是他们每个人对投票结果的影响微乎其微。当我们在预测模型中设计多层次的变量，并且每个层次又没有足够多的数据来影响预测结果时，就会发生这种问题。

为了解决这个问题，考虑合并一些分类层次。你可以通过合并那些对应数据少的类别数据，组成一个新类别，如：“其他”类别。还可以把那些对应数据少的层次与对应数据多的层次进行合并。这样就组成了粒度较粗的结构。这样的好处是模型比较稳定且易于管理。

1.2.3 包含几乎所有数据的层次

在这种情况下，所有的数据都包含在一个层次中。这种情况和前面所述的情况类似，这种变量对模型分析没有任何作用。模型的作用是否成功在于模型能否发现数据之间的差别。正向前面说的那样，太少的数据对于预测模型没有影响。如果所有的数据都属于一个层次的话，那么就和没有变量一样，不能为发现不同结果之间的差异性提供信息。

为了解决这个问题，需要把这种变量设计成为非单一层次。如果只有一层，那么对于任何模型它都是无用的，因为无法用它来发现大量数据之间存在的差异。

这种情况不要与下面的情况进行混淆：就是数据中包含大量的缺失值。对一个事件来说，事件发生可以用代码“1”来表示，如果事件没有发生的话，可以不给事件赋予任何值。在 SAS Enterprise Miner 中我们用变量 UNARY 来表示这种情况，因为 UNARY 只包含一个非空的层次。如果数据包含缺失值的话，可以添加一个新的层次，包含缺失值的观察值对应到新增的层次中。这时，这个变量就能够为整个结果的输出提供信息了。

1.3 处理连续型变量不当

连续型变量也经常令人头疼。由于连续型变量的转换能力很强，导致在有些情况下转换的价值被忽略。在变量选择过程中，有时需要考虑一个变量的转换版和非转换版。有四种连续预测变量的错误方法需要注意：极度偏斜、其中层次有峰值、个别层次经常出现，或者时间元素被忽视。这些问题在预测方法中也会出现。

1.3.1 偏度较大的预测变量

偏度较大的预测变量可能引起问题，因为用来预测目标的观察值数在输入值范围上变化太大。分布在两端的记录可能会对拟合模型产生严重的影响。由于大多数预测不是基于这些极端值的，对于很多重要的预测变量来说，结果模型拟合就可能不是太理想。这类问题同时也可能导致预测变量远比实际上重要（或不重要）的多。为解决该问题，至少有两种方法可供使用：

1. 转化原始预测变量，稳定变量取值的方差，并在取值范围内产生更一致的支持
2. 选择合适的分箱转化[2](Binning Transformation)，从而避免生成过的多（或过少）箱，为预测变量在每个比例范围内赋予适当的权重。

很多情况下，要同时使用这种方法，这样预测变量的连续形式和类别形式都能用于变量选择。变量转化为分类型后，在回归模型的响应中可以为非线性的。而变量转化为连续型的变量后，可以提高模型拟合预测变量在取值范围内的稳定性。

应当注意的是非线性转化（比如，Log 转化，平方根转化、多次方转化等），这些转化会使结果理解比较困难，因为相对于原变量，转化后的变量不是很直观。例如，在分析结果时，Log（dollars）在理解上就不如 dollars 直观。因为通常不会去想它的 Log 形式的取值 。然而，如

[2] 译注：SAS Enterprise Miner中，Binning Transformations指把一个测量尺度为区间尺度（interval scale）的变量转换成一个次序尺度（ordinal scale）变量。

果模型目标是预测，通过变量转化能够获得更稳定的模型。同时也要注意，如果转化了目标变量，也可能同时转化了导致模型结构上的不同假设的错误。从而，经过变量转化的最优的模型拟合就可能不同于没有做变量转化的最优的模型拟合。

1.3.2 峰值和分布

当预测变量的一个取值经常出现，但是其他值却服从一定的分布时，就会出现个问题。如果忽视了该变量的二元性，就可能导致低估或高估它的重要性。例如，一个变量可能对于目标变量非常重要，但这种重要关系在观测值为峰值时被掩盖了。峰值甚至可能发生在远离变量分布的一个值，结果就会是拟合效果显得比实际上更强（或者更弱）。如果同时模型拟合两部分，就不能如实地体现预测变量和目标变量直接的关系。

为解决这个问题，可以为该变量生产两个新的变量。一个布尔型变量，用来说明这个值是否为峰值；另一个预测变量的取值来自于分布。对于后者，把值为峰值的记录设置为缺失。你可以在回归模型或神经网络模型填补该缺失值。选择一种可以消弱影响预测变量和目标变量关系的补缺策略。布尔型变量包含关于取值是否为峰值的信息，而第二个变量包含用来评估除峰值以外的取值和目标变量之间关系信息。当然，你可能会考虑为两种情况分别拟合一个模型。很多情况下，分别拟合在性能上要比同时考虑两种情况的模型要更好。然而，如果存在这种很多变量，使用该方法是不现实的。

你也可能会考虑为该变量生产一个具有三个类别的缺失值指示变量，用该变量来说明以下三种情况的不同，一、预测变量的哪部分是峰值，哪部分是转化为缺失的部分；二、预测变量出现但没有被修改；以及预测变量的原始取值是缺失的情况。当考虑的预测变量有大量缺失值时，这个具有三个类别值的指示变量就会非常有用。最后，预测变量的连续部分的用处可能不大，考虑把该变量通过分箱处理转化为一个普通的预测变量来最优地预测目标变量。

1.3.3 单个类别频繁出现

这类问题是峰值和分布问题的一个极端情况，当其余值的峰值出现在最左或最右端时，这种情况就会频繁出现。在这个例子中，数据除峰值外分布实际上是很平坦的，而且这部分数据占的比例也相对比较小。任何可以识别的关系基本上都是有一些很极端的值所造成。因为这么多点出现峰值处，以及最大值（拟合线变化对它们影响比较大的值）具有很大的杠杆作用，当预测变量具有有限的取值时，它就可能会出现与目标变量强相关的情形。

为解决这种问题，可以通过对原始连续变量做分箱处理来生成一个新变量。很多情况下，因为只有小部分取值散布在峰值外，你可能会只生成一个二值预测变量。但是一定要记住，具有分类型的预测变量，而该变量的类别上实际上没有数据时，一般是无利用价值的，同时也可以根据峰值以外数据的分布生成多个类别。一个普通类别也能代表一组类别，从而使用数据的更大粒度的类别实际上克服通过生成太多无用类别的问题。

1.3.4 忽视或误用与时间相关的信息

这类通常发生在有时间戳的或按规律发生的（如，每月）数据出现时。在预测模型中融入历史信息会有显著效果，但是必须使用恰当。许多事务型数据的原始形式不能直接用来建模，可以将它们处理为能保留这些事务或时期信息的形式，同时这些信息对建模也很有用处。

通常，日期变量能被转换为用来衡量自从某件事情发生至今的时间（如账户开通时间的长短）或某事发生之前的时间（如，合同结束之前时间的长短）。当有周期型概要统计时，通常使用滑动窗口比较好。当使用固定的日期概要值（例如 Year-to-date(YTD) 统计）来为将来的观察值评

分时，会产生误解。很多情况下，与 YTD 的值相比，年初的评分将会很低，而年末的就会很高。滑动窗口考虑了相对与现阶段的行为。通常，考虑最近的行为和距离现在比较近的行为对决定是否发生了什么改变。例如，可以为 3 个月到 6 个月的每月的统计生成 Lag 变量，来查看 3 个月到 6 个月以前时间的总体平均。当要说明每季度的情况时，该方法就很有价值，这样季度变化就不会被当作是独立的变化。

假如现有业务数据可以概括为阶段型数据，每行对应一条观察记录，而每列对应于一个特定时期的概括统计。当使用基准信息和最近信息去判断基准信息的变化对预测目标变量是否有预测性价值时，这种方法就显得很实用。变化被识别后，可以把这些变化用作预警信号，接着就可以采取干预措施来阻止意外结果发生。考虑评估最后若干时间段和基于更远时期的基准信息段的价值。观察这两类的信息可能会增加进入模型中的变量数目，所以明智的做法是要对这些变量进行变量选择。检查数据中是否存在季节性数据可能会非常有用。调整季节性数据能够提高整体的拟合效果，并能更好地理解整体的关系。

2 定义变量角色、抽样，和定义目标变量轮廓（target profile）

在可用数据被评估之后，并且分析者已经确定了如何准备数据，接下来应该考虑用其中的多少数据用于分析。以往因为数据量是十分有限，则需要分析每个观察所得数据，但是数据挖掘主要应用于大规模数据集的情况下。基于所有数据建立的模型看起来是令人满意的，但相对于一个选择良好的样本，分析所有数据时间的花费经常超过它带来的益处。这里面临的挑战就是如何确定一个合适的样本，以便用它来分析总体数据集的情况。剩余数据用于验证建立的模型。任何抽样策略需要考虑到目标变量的性质和预测变量的数量以特征。在选择变量并确定样本之后，必须先评估目标变量以保证建模的策略是适当的。在目标变量为分类型变量的情况下，有必要建立一个目标变量轮廓来获得一个有用的模型。特别是当我感兴趣的分类层次相对于总体数据或更大数据集取值较少时。

2.1 不适当的元数据

建立正确的元数据对建模过程至关重要。元数据决定每个变量如何使用。SAS Enterprise Miner 根据变量名或变量值自动地指配建模的类型和变量角色。不幸的是，这个过程无法防止不适当的变量(例如,数字形式的 ID 变量) 被看作连续型输入变量,因为数值型数据经常有很大数级。证明信息、日期信息和许多分类变量经常以数字形式出现且以数值变量形式存储。分析过程中，不恰当地使用这些变量，很容易产生不可预知的结果。

为了解决这个问题，在这些数据提交到建模节点时要考察每个变量。您将经常发现在上面章节描述的许多问题，必须重视以使数据在建模中发挥最大效用。这可能极其费时，因为数据挖掘经常涉及上百甚至数以万计的变量。

2.2 不充足或过多输入数据

一些分析师认为对他们的数据抽样会导致低劣的结果，因此他们宁愿分析整个数据集。虽然在一些情况下这也许是事实，但并非总是如此，即当样本被合适的选择时。在选择样本建立候选模型后,剩余数据可用来比较竞争模型和评估最后的模型。

当为偶发事件建模时，抽样是几乎必要的，目的是获得一个性能优于零模型（null model[3]）的模型。在零模型中，对分类型目标变量，每个观测值分配到最为频繁出现的组，对连续型目标变量，则分配到它的均值。在偶发事件情况下，虚无模型能非常精确。例如，考虑一个二分型目标变量，我们感兴趣的分类层次（例如是否违约中的“是”）只占 1%，那么零模型可以把所有的观测都不归到我们感兴趣的目标事件中，从而获得 99%的准确率。（违约的只有 1%，把所有观测都归为不违约，只有 1%的误差）

在其它情况，大量数据的出现为试验验证适合的任何模型提供了机会。由数据的一部分(训练数据)调整所建模型，由备用样本(或校验集数据) 来比较结果模型 。这些样本能被用来评估经过调整模型的执行效果。选择一个在训练和校验集上都运作良好的模型能提供保护措施寻找数据间的偶然关系。另外，如果足够的数据是能允许分割数据集成第三个集合来测试最后模型，用户用这些数据，即测试集，有机会获得最终模型执行的无偏评估。

观测样本很少所产生的后果很容易理解，但样本过多也许会增加计算时间，而对预测结果没有很大影响。这额外处理时间可能相当多，缩短了评估和改进中间模型的时间。使用过多数据作为训练/检验数据集也会导致只有少量甚至没有数据来用作测试集，导致获得模型执行效果的无偏估计十分困难。

对数据的欠抽样也是同样危险的。欠抽样频繁地发生在分析人员计划用一个相对不频繁的分类层次来为一个二元分类变量建模的时候。在二种可能结果和固定的样本大小的情况下，当从每一组取同样数量的观测值时，在假定同方差和其它参数不变的情况下效果最好。但是去除数据以获得这些相等大小的样本比起原始的数据各组样本规模不等时效果要差。所以，不应该为建立相等数量大小，而去除额外的数据。(Muller 和 Benignus 1992) 。

当对一个分类层次不均衡的二元目标变量建模时，分析者经常对数据取样，每种结果各取 50%的比例。除非结果概率被调整，否则结果概率是不可信的，因为他们反映的是样本数据而不是总体数据。在偶发事件的情况中，因为也许存在相对少量的感兴趣的结果，许多问题显露出来了。一个样本包含所有的偶发事件和同样数目的其他结果往往是一个很小的数据集。此外，少量抽样对于样本选择的结果很可能不足代表这一群体，这一群体代表了总体的大多数。

在许多情况下，按比例抽样并通过调整目标轮廓来处理比例层次不平衡的情况将更为恰当。在偶发事件或者预测变量与目标变量之间关系不是很显著的情况下按比例抽样也许不合适。在这种情况下，过抽样但是对每一群体采取相等的观测数量产生一个没有罕见事件不代表这一群体的样本是合情合理的。当过抽样数据时建立一个目标轮廓来产生所需要的决策规则是重要的。下面的部分将建议几个目标轮廓的策略。

没有什么简单的方式来确定一个最小的观测数量。大多数计算样本规模的策略集中在观测数量将需要获得一个特殊的置信水平，或者给定一系列假设达到某一个准确性的层次。实际上硬件条件的限制经常要求抽样的规模适用于给定的模型中。许多情况下，处理时间或磁盘空间需求所需成本的增长要远远快于增加样本规模带来好处的增长。

在变量个数特别多的情况下，使用一个较小的初步样本来确定哪些变量对预测结果作用不大，这对样本的选择是很有用的，此外其他的方法包括去除不重要的类别变量，连续变量可以被去除或者通过数据简化技术例如主成分分析简化数据。通过减少原始数据集的量，具有较大数量的观测值可以及时的被分析。

[3] 译注：Null model，是这么一个模型，它只有截距项，而没有任何自变量。怎么说？比如我们建立的模型采用了多少多少自变量，还有一个模型就是根本不采用任何自变量

2.3 对分类型目标变量不适当或者缺少目标变量轮廓

为了理解你做出的关于目标轮廓的选择的影响，去了解模型如何被评估是非常重要的。没有有效的指定一个目标轮廓等同于选择默认的目标轮廓。一旦分类层次不均衡或者对于误分类存在较大不同的代价，使用这个默认的轮廓将导致非优的结果。

对于一个目标类别，默认的目标轮廓选项对每一种预测准确结果赋予相同的权重。这种情况下决策规则很简单----即选择概率最大的结果。这个规则在除了样本比例不均衡的情况或结果的误分类代价有显著不同的情况以外是非常适用的。

2.4 目标变量的分类层次不均匀

在这种情况下，一个事件发生的频率比另外一个事件低。在预测模型中稀疏事件很平常，所以这些事件更容易发生。即使样本均衡，调整先验概率以反映一个不均衡的样本总体，利用重复取样来调整后验概率。这个默认的决策规则不太可能选择较低频率发生的结果层次。

假设一个二元目标，感兴趣的事件在总体事件中发生的概率是 10%(Pre(event))=0.1),并且非感兴趣事件发生的概率为 90%(Pre(nonevent))=0.9）。在建立一个预测模型后，假设一个观测事件被模型预测为 40%可能会发生，即概率 Pre(event))=0.4,同理 Pre(nonevent))=0.6。注意到虽然这个观测值与总体事件中随机的一个观测事件相比更可能发生的概率大 4 倍，这个观测事件发生的概率比非感兴趣事件的概率要高。如果对每一个结果做相等的准确预测的加权，我们将把这个观测事件最终预测为非感兴趣事件，因为它非感兴趣发生的概率更高。事实上这个目标事件被预测为目标事件的概率比不是目标事件的概率要大 5 倍。同样的，当使用默认的决策规则时这个目标事件被预测为目标事件的概率比不是目标事件的概率要小 5 倍。这种不平衡导致变量选择放弃所有的预测，因为没有模型可以利用这些可用的预测构建，这些预测能够识别出发生概率五倍于目标事件的观测变量。

为了克服这个问题，指定一个不同的有利事件正确的预测结果。在大多数情况下，目标变量是二元的，并针对目标事件比起替代事件发生的可能性更小。在不是二元目标的情况下，建模工具可以运用逻辑描述多个模型，其中每一个模型包括一个作为目标事件的时间和剩余的非目标事件。另外，这种逻辑可以被延伸到 k 级目标变量，但这，没有在本文提到。

假设响应变量 Y 取 0 或 1，1 表示对事件的影响。假设X_1, X_2, X_3...X_n表示感兴趣的输入变量。

假设：P 是基于模型的目标事件未校准的预测概率

p_{adj}是基于模型的目标事件校准过的预测概率

p_1 是样本中目标事件的比例

p_0=1-p_1 是样本中非目标事件的比例

τ是总体的目标事件的比例

$1-\tau$是总体的非目标事件的比例

则对一个特殊的观测的校准概率是

$$P_{adj} = \frac{(P * \tau_1 * p_0)}{[(P * \tau_1 * p_0) + ((1 - P) * \tau_0 * p_1)]}$$

进一步假设预测与实际的决策矩阵是

		Predicted	
		1	0
Actual	1	D_{tp}	D_{fn}
	0	D_{fp}	D_{tn}

其中

D_{tp} 准确预测的事件情况的利润（真阳性（True positive，TP）被模型正确预测的正样本）

D_{fp} 错误预测的事件情况的成本（假阳性（False positive，FP）被模型错误预测的负样本。）

D_{fn} 错误预测非事件的成本（假阴性（False negative，FN）被模型错误预测的正样本）

D_{tn} 准确预测非事件的成本（真阴性（True negative，TN）被模型正确预测的负样本）

SAS EM 基于众所周知的贝叶斯规则的分类器的扩展来分类每个观测，这个分类器使期望损失最小。 对以一个二元目标 1（正）和 0（负），如果后验概率

$$posterior\ \ probability > \frac{1}{1+\left(\frac{\text{cost of false negative}}{\text{cost of false positive}}\right)} = \frac{1}{1+\left(\frac{D_{fn}}{D_{fp}}\right)}$$

贝叶斯规则分类一个观测值为 1（正）。

在这种情况下，如果

$$P_{adj} > \frac{1}{1+\left(\frac{D_{tp} - D_{fn}}{D_{tn} - D_{fp}}\right)}$$

SAS EM 分类一个观测值为 1（正）。这个公式当$D_{tp} = D_{tn}$=0 的时候与贝叶斯规则是相等的。

例如默认情况下 ＳＡＳ　ＥＭ使用$D_{tp} = D_{tn}$= 1，$D_{fn} = D_{fp}$=0，当没有目标概要被建立时规则简化为：

$$P_{adj} > \frac{1}{1+\left(\frac{1-0}{1-0}\right)} = \frac{1}{1+\left(\frac{1}{1}\right)} = \frac{1}{2} = 0.5$$

被分配给任一个结果的相等的值准确的导致了选择一个调校概率值大于 0.5 的结果的决策规则。在这个例子中处理一个二元目标值，一个结果的概率可以被通过减去另外一个结果的概率来计算。所以这个规则与选择最大校准后验概率的结果是相等的。如果没有指定先验情况，并且假设随机取样，在这种情况下

$$E(p_0) = \tau_0$$
$$E(p_1) = \tau_1$$
$$P_{adj} = \frac{P*p_1*p_0}{(P*p_1*p_0)-((1-P)*p_0*p_1)} = \frac{P*p_1*p_0}{p_1*p_0*(P+(1-P))} = \frac{P*p_1*p_0}{p_1*p_0*1} = \frac{P*p_1*p_0}{p_1*p_0} = P$$

P^*表示上述模型预测结果是目标时间的概率，如下所示：

$$P^* = \frac{1}{1+\left(\frac{D_{tp} - D_{fn}}{D_{tn} - D_{fp}}\right)}$$

在这个简单但常见的二元目标的例子中，伴随一个我们感兴趣的分类层次发生不频繁的情况，我们可以通过明智的选择 Dtp, Dfn, Dfp, Dtn 控制临界阈值。实际上存在这些值的许多结合值将产生同样的决策，因为这些值的不同的比例控制了分母的值。

实际上经常花费很大的代价就是要预测目标变量的值。在一个直销市场中，预测某些不会购买的人群将不会向他们发送邮件。因为临界概率只依赖于差异的比例。假设 Dfp=Dtn=0

		Predicted	
		1	0
Actual	1	D_{tp}	0
	0	D_{fp}	0

在这个例子中，公示变为

$$P^* = \frac{1}{1+\left(\frac{D_{tp}-0}{0-D_{fp}}\right)} = \frac{1}{1-\left(\frac{D_{tp}}{D_{fp}}\right)}$$

所以选择合适的 Dtp, Dfp 产生所需的阈值。例如假设把被预测值为 1 的人定为目标客户，对于这些人市场营销人员将花费 1 美元成本。假设响应者平均消费了 10 美元，未响应者花费了 0 美元。则 Dtp=10-1=9 美元，Dfp=0-1=-1 美元。 因为我们考虑一个比例，取消单位，则

$$P^* = \frac{1}{1-\left(\frac{9}{-1}\right)} = \frac{1}{1-(-9)} = \frac{1}{1+9} = \frac{1}{10} = 0.1$$

同样的如果一个响应者平均消费了 4 美元，体验的成本考虑为 1 美元， Dtp=4-1=3 美元，Dfp=0-1=-1 美元，

$$P^* = \frac{1}{1-\left(\frac{3}{-1}\right)} = \frac{1}{1-(-3)} = \frac{1}{1+3} = \frac{1}{4} = 0.25$$

可以看出一个暗含的模式。如果假设一个固定的消费单位，并且分派-1 与 Dfp 相关，则选择 Dtp 的值，渴望得到的临界概率可以这样计算：

$$P^* = \frac{1}{1+D_{tp}}$$

假设 Dfp=-1，选择 Dtp 等于

- 1, implies $P^* = \frac{1}{1+1} = \frac{1}{2} = 0.5$
- 4, implies $P^* = \frac{1}{1+4} = \frac{1}{5} = 0.2$
- 9, implies $P^* = \frac{1}{1+9} = \frac{1}{10} = 0.1$

观察到这个方法是以特殊的情况将成本合并到利润当中。如果对目标轮廓设置一个固定的成本为 1。SAS EM 借此成本从设置在决策矩阵中的值减去它。

结果是，为了得到截距 0.1，应该指明：

		Predicted	
		1	0
Actual	1	9	0
	0	–1	0

如果，没有指定成本，则

		Predicted	
		1	0
Actual	1	10	0
	0	0	0

如果您已经指定一个成本为 1.0 可用于预测目标事件（在这个例子中，Y=1 ）。在某些情况下，目标是使用先验概率仍可能产生零（只有截距项）模型，即使是非常大的比重放进收益矩阵，除非变量选择的设定被修改。为了处理这种情况，采用过抽样和调整基于过抽样的收益矩阵而不是调整原始的先验概率的方法是较好的。这样做，你必须调整后验概率，除非你只对事件的排序感兴趣。在这种情况下调整的概率可能会影响到若干意见如何被分类，但事件的排序不会被改变。

2.5 错误分类代价的差异

出现此问题时，你正在使用决策规则与实际的误分类代价不一致的样本。模型选择旨在寻求使模型最优化的某一决策规则。当决策规则没有反映真实的误分类代价，选择的模型可能无法达到最优。为了克服这个问题，创建目标轮廓应尽可能的贴近实际的决策规则。这样，变量选择和模型评估能得到最佳的可能模型。

3.数据划分

确定如何准备数据和取样之后，下一步则是考虑如何划分数据以进行分析。今天的数据挖掘

方法允许非常灵活的建模策略在很短的时间内被部署就绪。这样的建模策略灵活性使得数据可以被更好地拟合。但另一方面灵活的方法也能导致即使是大量数据也出现过度拟合。通常挖掘时数据总是足够的，所以合适的保留取样就很有必要。这里的错误多由错误理解所划分数据的作用，或二类备用样本中某类的数量分配不当造成。

3.1 误解：数据集划分的定位

在 SAS Enterprise Miner 中，数据划分节点共有 3 种关键数据集可用：训练集、校验集、测试集。训练集用于建立候选模型；校验集用于在某节点或跨节点间比较不同模型；测试集则用于提供最终模型实际性能的无偏估计。

这里最常见的错误是对 SAS Enterprise Miner 关于这些数据集的默认设定理解有误。当在建模节点中使用神经网络，决策树或逐步回归时，如果校验集存在，SAS Enterprise Miner 默认使用校验集来对建模节点选择不同模型。测试集不被建模节点用作模型选择，但也为这些观测结果产生预测值和符合程度的统计结果。可是，SAS Enterprise Miner 在模型比较节点中，比较不同模型对建模节点的符合程度时 ，默认将使用在测试集中表现最好的模型。在这里，测试集的表现没有提供模型性能的无偏估计，因为它现在被用作模型选择，所以已经等同于又一个校验集。当然默认值也可以修改为用校验集或者训练集来测试在模型选择节点的性能，但在缺省情况下且测试集存在时，将使用测试集来确定最佳模型。如果没有测试集，则基于校验集选择最佳模型。如果两者都没有，则基于训练集选择最佳模型。

将测试集当作另一个校验集还可能有一个好处。校验集有助于减少用训练集建立候选模型带入的偏差，测试集则可以最小化用校验集在给定建模节点中选择最佳模型带来的偏差。实际建模中训练、校验和测试集的性能应该是相差不大的。如果这些数据集的结果相差很大，意味着可能有模型过拟合，或取样策略不具有代表性，或样本数不足。无论你怎样使用测试集，都很有必要了解结果可能产生偏差的情况。总的说来，测试集只有在选定最终模型时才可以提供对模型性能的无偏估计。

3.2 误区：未能考虑对默认划分的更改

默认情况下，数据分割节点将原始数据分为训练集（40%），校验集（30%）和测试集（30%）。当目标变量是类别变量时该节点缺省将样本按目标变量分层[4]。可是，没有特定的方法可以保证将观测值正确划分为训练，校验和测试集。所以，用户必须小心确保能根据手头的问题进行相应的合理划分。在前面关于样本大小的讨论中，关键是调整数据集以确保在训练和校验集中有足够的观测值。如果没有足够的数目来单独划分一个测试集，最好还是使用现有训练和校验集里的数据。

有时候，含目标特征的观测值数量少到根本不该被分割的地步，此时要非常小心，因为建模节点将无从判断可能的过拟合问题。回归模型和决策树模型可以由熟悉数据内部关系的专家来判断修建。但是作为极度灵活的建模方法，神经网络没有类似的保护措施，所以当校验数据不足时最好避免使用神经网络。实践中，常有含目标特征的观测值数量少于期望的情况。此时分析员要灵活决定是否分割和如何去分。

[4]译注：将样本按目标变量分层，让每类变量在每个数据集中按其比例均匀分布

4 变量选择

在数据抽样之后，尤其是数据分割完成之后，要进行变量选择。变量选择就是从大量的候选变量中识别出对模型比较重要的变量子集。因为回归模型（线性和非线性回归模型，如神经网络）是基于完整记录的，无论是依赖变量还是独立变量，都必须是完整的，替换记录中的缺失值十分重要。而决策树可以自动处理含有缺失的记录，缺失对于这类模型就不是必须考虑的问题。

在很多情况下，数据补缺既要在变量选择前做，又要在变量后做。通过比较数据补缺后的变量和原始数据的变量，可以得到一些启发。缺失的出现可以归咎于两个原因，一是标准的设定（即，只有不符合特定标准的值或标志才被认为是缺失的5）；二是数据的不完整性。前者是数据准备问题而不是数据补缺问题，它涉及到猜测变量的未知值。后者中的数据补缺可能会影响到选择的变量。如果在执行补缺时选择不同变量，那么补缺过程就可能出现一些问题。不管变量选择是在补缺前还是在补缺后，数据准备过程中是否考虑了先前讨论的连续变量和类别变量中可能存在的问题才是重要的。变量选择中可能出错的情况包括选择前未进行变量评估、只运用一种变量选择方法，以及误解或者忽略了变量选择选项的设置。

4.1 变量选择前缺乏变量评估

数据挖掘分析通常要考虑大量的变量。此外，依赖于时间的变量、总统计变量、转换变量以及缺失指示变量6都会生成大量的预测变量。像先前描述的那样，在数据准备前通过变量选择方法筛选这些潜在的输入变量，可能会选出一些看似重要的变量，但这些变量却不能很好地概括总体。在很多情况下，由于一些变量的缺失值比例、峰值比例（如果存在一个峰值），和过多类别的原因，排除这些变量可能要比包含它们更合适。一些模型方法可以合并其中的一些类别，也可以解决偏度较大的预测变量（如，分箱转换）转换或者剔除这些存在问题的变量通常远比把它们不作处理就包含在模型中要好。

由于包含大量变量，预先评估这些变量，听起来可能令人痛苦不堪。尽管一些自动的处理方法可以避免大量不愿看到的建模问题，但它们也不是完美的。数据看起来不错有时可能会由于其他的一些原因而不合适。例如，某个变量可能会经常不按它名字或标签所指的含义使用。如果分析员只关注模型的拟合度，而不分析研究输入变量的值，那么就会造成模型在实际应用中的性能表现不佳。这种情况下，多想想“垃圾进，垃圾出”这句格言是比较有用的。

研究分析整个变量集合以及识别出使用这些变量所蕴含信息的方式，不仅对现在的模型有价值，对将来的模型也同样有价值。在某些情况下，这种探索分析可以识别出未来模型会使用到的变量子集，这样就可以缩短其他模型的建模时间。此外，在不需考虑目标变量时，大部分变量转换操作可以改善记录的实用性。尽管在进行数据分箱处理时，需要优化该变量与目标变量的关系，而执行这一步时却无需将来的进一步探索分析，这也可以额外缩短建模时间。需要标准转换的一些变量转换操作，一般可以嵌入到 ETL 过程中去，这样可以进一步减少建模时间。

通常，所采取的措施要根据具体的情形而定，在自动选择和转换变量前探索分析所有可能的变量不一定是可行的或切实际的。在这些情况下，检查已选择的变量以确保结果模型选择的变量能很好地概括总体，是非常重要的。

---

[5]译注：这跟具体的应用相关。例如，如果年龄小于 18 岁，在某些应用中就可以被认为是缺失的。

[6]译注：缺失指示变量，Indicators，在一些文章中把翻译为指示器。Sas中可以为含有缺失值的变量生成一个指示变量，用来说明该变量在记录中是否缺失，如果缺失就标记为 1，否则为 0

4.2 仅使用一种选择方法

SAS EM 提供了若干种不同的变量选择方法。一些模型节点也有变量选择功能，这些模型节点包括变量选择节点、决策树以及回归节点。如果变量选择过程局限于这些节点中的一个或这些方法中的一种，就可能遗漏一些对改进模型整体性能比较重要的预测变量，限定的选择方法还可能会使一些不应该进入最终模型的预测变量而进入了模型。在有限变量的情况下，运用多种变量选择方式的优势或许会受到限制。然而，随着变量的数量和变量之间的关系复杂性增加，多种选择方法的优势就会增加。因为一定的情形下很难事先知道哪种方法是最好的，这时考虑各种变量选择方式将非常有用。

对于目标变量是分类型的模型，变量选择可以提供三种方式，其中包括 R2，卡方和两种方法的结合。目标变量是连续型的模型只要 R2 方法可以选择。在两种方法的结合中，变量只有同时被 R2 和卡方选择的时候才能作为输入进入该方法。实际上，两种方法结合的方法可以避免过度拟合情况，但是考虑到变量选择方法只在训练数据上进行，这种方法就显得有点不完善。结果会出现选出的变量可能不适合其他预留的数据。

决策树和逐步回归模型默认情况下使用验证数据获得最终的结果，这样可以避免过度拟合。但是，使用唯一的方法可能无法解决模型的其他问题。考虑模型的交互性和非线性关系时决策树模型是非常好的，但是，它不能很好地模拟简单的线性关系。逐步回归模型不能识别出非线性的关系，除非把这种关系添加到模型中去。例如，线性模型不能评估二次方关系，但是可以通过把二次项加入模型来实现。此外，逐步回归模型可以用来优化数据中可能导致错误结果的随机相关（chance associations）。最后，逐步回归要在变量选择的同时进行模型拟合，这样速度一般会比较慢。

为避免这些问题，尤其是涉及到大量变量的时候，可以考虑使用各种不同的变量选择方法，利用这些方法选出的变量来建一个预测变量池。这可能需要一些手工操作，但是这种方法是最安全的，可以确保每个变量被公平地考虑。经过最初的变量选择后，你可以执行第二次的变量选择，确保使用的方法可以避免模型的过度拟合，例如决策树或逐步回归。然而，先前讨论的方法的危险之处是，我们已经运用多种变量选择方法选出了最初的候选变量，这些变量限制先前提到的问题对模型的影响，而不是决定建模的最终变量集，以及通过评估模型在不同候选输入变量上的性能决定最终模型。

4.3 误解或忽略变量选择选项

变量选择节点有两种执行模式，即卡方和 R2 模式。该节点也可以使用两种模式结合的方式。结合的方式只考虑同时满足卡方和 R2 两种模式的变量。不管选择哪种模式，理解默认设置以及如何根据期望的结果来改变设置是非常关键的。

4.3.1 选择卡方设置选项

在卡方模式下，用户可以指定三种额外选项：其中一个是用来决定把连续型变量分成多少个相同的箱；另一个是控制在运行二项分割时的迭代次数；最后一个是指定一个用来决定预测变量是否保留的阀值。默认设置下，阀值为 3.84，它对应于自由度为 1 的卡方分布中的显著水平 α=0.05。阀值不是用来判断变量最终是否重要，而是用作判断的标尺。任何没有满足最小显著水平要求的变量都会被排除。

不幸的是，从数据挖掘定义上看，大多数数据挖掘应用要有大量的数据。但是随着数据量的

增加，特定的检验统计指标就会越显著。换句话说，就是在给定样本量差别不显著，而在更大的样本量上就会变得显著。很多情况下，数据挖掘时通常取 α=0.05 可能会使过多的变量进入模型。尝试把 α 设置为不同的值可以使分析员精确的控制保留变量的个数。在变量还需进一步选择的情况下，可以考虑保留一些不太重要的变量直到进入下一个变量选择阶段。然而，如果这个节点用于选择进入最终模型的变量，那么就需要慎重考虑这些不太重要的变量的实际重要性了，评估它们是否要包含在最终的模型中。

至于卡方的其他设置，增加迭代次数可能获得较好的拟合，但是会消耗大量的处理时间，而减少迭代次数可能在一定程度上影响性能，但可以节约大量时间。在数据挖掘中，记录数以及变量数可能会极其大，因此减小循环数和/或者增大阀值，可以加快处理时间。同样，减少把分箱数控制在默认的 50 箱下也可能提高运算速度。

4.3.2 选择 R2 设置选项

在 R2 模式中，每个变量的相关系数平方（简单 R2）都会被计算，并与默认的最小值进行比较。R2 小于最小值的变量将会被拒绝。在这个选择后，再进行逐步选择。而那些在逐步选择时 R2 改善小于阀值的变量也会被剔除。其他设置也能用于分组的连续变量（AOV16 变量）和类别变量时请求交互。AVO16 选项基于方差分析为每个连续型输入生成了 16 组数据。

如先前讨论的卡方模式，用户可以为 R2 值设置最小阀值来决定变量是否能进入模型。对于给定的数据集，该值设置的过小就会包含太多变量，而设置过大就会排除很多变量。变量数的最大值选项可以控制包含的最多变量数，因此，模型包含的变量数可能会出现少于满足最小 R2 的变量数。此外，如果要进行其他的变量选择操作，允许额外变量进入模型的变量候选集中是非常合适的。如果现有的变量数远少于期望的变量数，那么就不需要通过改变这些设置的取值来选择变量。AOV16 变量允许预测变量和目标变量之间存在非线性关系，但是这样会增加参数估计的时间。同样地，生成交互变量也会增加要估计的参数。因此，使用这些方法时要保守点。除非变量开始时有很多类别，否则不需要把类别变量分组。

5 替换缺失值

SAS EM 的替换节点在默认情况下，使用变量的平均值填充连续变量的缺失值，使用抽样模式填充类别变量的缺失值。不幸的是，这些方法在一些情形下可能不是很理想。这个问题的严重程度跟缺失值占的比例以及模型目标变量有关系。缺失值对模型的影响可能会很惊人，因为完整记录所占的比例通常比较小。随着变量的增加，记录含有缺失的概率就会增加。在不考虑选择的补缺方法的情况下，评估和目标变量有一定关系且有缺失的记录是非常重要的。补缺时经常会因为没有评估补缺方法或忽略了缺失指示变量而出现错误。

5.1 未评估补缺方法

对于连续变量，缺失值一般暗示着变量的值为特定值（一般是 0）。在事先知道的情况下，使用实际值来补缺要远比使用补缺方法来猜测值更有意义。使用平均值补缺会使记录看起来很特别，当缺失比例比相对比较高时，会在分布中出现一个峰值，这可能会导致先前提及的数据分布问题。此外，有时候变量的值可能会对精确预测结果非常关键，此时，最好使用决策树预测该值，即运用现有的信息来获得缺失值的最佳预测值。由于决策树方法会为每个缺失变量建立一个独立的模型，这种方法不应当运用于所有变量，尤其是当一个缺失蕴含一个特定的补缺后的值时候，或者当要补缺的变量对模型不是非常重要的时候。

变量是分类型的时候，缺失可以作为单独的类别来处理。在含缺失的记录所占的比例比较小

的情况时，这种方法可能会出错，因为它为了少量的记录而为模型增加了一个参数，这时比较合适的做法就是默认的模式，如果整个变量对预测响应比较重要，还可以使用决策树。对于连续型变量，决策树方法除非是真正需要，否则不应用于所有变量。

5.3 忽略缺失指示变量

与回归和神经网络模型相比，决策树有个很独特的优势，就是它可以直接处理缺失值，而不用作任何补缺处理。在缺失与目标变量相关的情形下，决策树可以合理的响应，因为它知道哪条记录有缺失值。然而，在回归或神经网络模型中，这些缺失值已经替换为平均值或其他值，那么这条记录就可以用来分析。如果不使用其他措施，回归和神经网络模型现在是无法知道哪条记录先前含有缺失值。

为了解决这个问题，你可以为这些变量生成缺失指示变量。但不是为每个含有缺失的变量都生成指示变量，而是为那些在缺失和完整的情况下对结果有很大差别的变量生成指示变量，即考虑在缺失对响应有帮助的时候，把变量数增加一倍。同时也要考虑缺失在记录中占的比例，这样就能避免为那些其实所有记录都在相同类别的变量，重新生成一个新的类别变量[7]。

6 拟合线性回归模型

线性回归模型由于简单和易于接受而非常受人们欢迎。神经网络和决策树多少有些受怀疑，回归模型却因其有效和便于解释而被广泛接受。不幸的是，一个方法的普及使用有时会导致对它潜在缺点的忽视。在拟合线性回归模型时出错，经常是源于对逐步回归方法的过度使用，或者是对结果的错误解释。

6.1 对逐步回归方法的过度使用

逐步回归技术因为它的易于使用，以及它不需要人力过多的参与就能生成模型的能力而大受欢迎。利用逐步回归的两个大毛病，一是它有优化数据中偶然因素的趋势，以及它相对缓慢的运行速度。不幸的是，这些毛病会随着观测值和变量的增多而变得更严重。

逐步回归相对缓慢的运行速度，可以很容易地通过计算任意一个包含大量输入变量的大型数据而表现出来。通常，对建模而言，时间都是有限的。把时间花在等待回归模型的处理运行，不如花在使用各种变量选择技术来评估那些输入变量，这样就可以在运行回归之前减少要考虑的变量数量。那些变量选择技术经常能帮你洞察模型输入数据，以生成一个更好的最终模型。

关于这个问题，更多的考虑见 Derksen and Keselman (1992)。他们考察了各种逐步回归的变量选择技术，声称，使用逐步回归选择算法的子模型都包含了可信变量以及干扰变量。他们进一步宣称，在那些最终的子模型里面，留下来的可信变量的平均数目不到实际可信变量的一半。这对关心模型效果的建模人员和试图解释结果的商业分析者来说都是很糟糕的消息，因为他们面临这样一个风险，即他们的商业决策要基于这么一个随机发生的模式[8]。

---

[7]译注：如果缺失的比例大，所有记录都有缺失，那么就没有必要再生成新变量来指示是否缺失，可以考虑直接拒掉。或者当缺失的比例很小时（也许可以忽略不计），也没有必要去生成新变量，可以考虑直接补缺。

[8] 译注：前面说过，逐步回归有优化数据里偶然因素的趋势，使用这个方法会把一些不相关的变量纳入模型。这些偶然因素进入模型，所展示的模式就是随机的了。

对上述发现的反应，即一个模型是不是需要包括每一个变量，是要听取专家意见的。Derksen and Keselman (1992)总结道，“对最初的自变量集合的选取要非常小心，包括研究那些仅仅是基于理论或以往的研究而知晓或估计的对反应变量有关的变量。”这个处理方法的困难是双重的：

1. 需要考察的变量数目众多，使得仔细考察它们太花时间；
2. 数据挖掘的目标是发掘数据中新的模式，这些模式可能与通常的思维方式和/或者历史发现一致，也可能不一致。

另外，在一个典型的数据挖掘问题中大量的可选择变量也可能包括大量的干扰变量。

一篇更新的文献指出 Derksen and Keselman 的担忧可能被夸大了。Ambler, Brady, and Royston (2002)考察了逐步回归的性能并发现，即使在数据中有大量的不相关的变量，这个标准的变量选择技术运行也良好。这些发现似乎与 Derksen and Keselman 不一致。面对这些对立的观点，在使用逐步回归技术建立模型时就更要小心谨慎。

为了克服这个问题，分析人员必须选择这么一种策略，在规定的时间期限之内得出模型，这个模型又要尽量避免选中那些干扰变量。当变量的数目急剧增加时，任何建议的解决方案都需要重新修订。另外，犯第一类错误或第二类错误的严重性也同样要通过评估模型包含一个特定变量来衡量。所有选定的策略都须经分析人员的细心评判，以便他们针对特定的情形做出恰当的选择。

6.2 不精确地解释模型结果

对那些基于数据分析而做出决策的人来说，回归是一个熟识的概念，相对地，其他如决策树和神经网络模型可能就不那么熟悉了。这种熟悉程度的差异导致对回归模型的使用偏倚，因为人们都认为它更具有可解释性，以及由于它悠长的使用历史而产生的相对安全性。这些结论可能是有误导性的，可能导致对模型结果的不正确解释和/或不恰当应用。

假设你的回归曲线由下式给出

$$\hat{Y}_i = 2 * X_i$$

其中：

$\hat{Y}_i$ 是第 i 个观测对应的因变量的预测值

X_i 是第 i 个观测的自变量的值

上式表明，X_i 每增加一个单位，相应的因变量 $\hat{Y}_i$ 就增加两个单位。不幸的是，随着更多的自变量的引入，这种解释就变得更复杂。又假设你的回归曲线由下式给出，

$$\hat{Y}_i = 2 * X_{1i} + 7 * X_{2i}$$

其中，

$\hat{Y}_i$ 是第 i 个观测对应的因变量的预测值

X_{1i} 是第 i 个观测的第一个自变量的值

X_{2i} 是第 i 个观测的第二个自变量的值

可知上面的等式可以改写成

$$\hat{Y}_i - 2 * X_{1i} = 7 * X_{2i}$$

这表明，X_2 和 Y 中不能由 X_1 解释的部分存在一定的关系。如果两个变量高度相关，那它们就不可能模型中并存。在这种情形下，该模型就不能说一个变量是重要的，而另一个就是不重要的。相反，这个模型只不过表明，这个被拒绝的变量没能显著地解释那个选中的变量所解释不了的变异性。最终模型中变量的特定组合可能包含被拒绝变量的大部分信息。

在一些案例中，建模者需要评估每个变量对最终模型的价值。只要意识到任何做出的决策都是相对于一个既定的数据的给定的自变量集合而言，这样的评估就是有理由的。随着数据不断地进行着变量选择的交替，这个任务和随之而来的解释工作都将变得更复杂，因为大多数人都想解释其中的主要方面，但以上交替过程使这些解释依赖于有问题的变量的数目。另外，与能更有效地处理非线性问题的决策树和神经网络模型相比，回归模型实在是适应性不足。回归模型需要一个自变量与因变量的关系定义得比较好的模型。假如这个模型不能充分地反映其中的关系，回归模型表面上的易于解释性就变得毫无意义了,因为它不能如实地反映输入变量与目标变量的关系。

为了克服这个问题，就需要考虑所有能影响一个特定的变量在一个特定模型中的重要性的因素。任何变量对模型的重要性都需要在真正对模型有用的变量集里来解释。在对似乎是从解释中得出的结论投入大量资源之前,需要花些时间在备用数据如测试数据上核查得出的关系是否有效。还有，考察其他更具适应性的模型如决策树、神经网络模型的效果，以评估回归模型是否会丢失有效信息也是有用的。从回归模型中获得的结论，只有当模型提供了对真实关系的充分反映才是有用的。细心对待那些看似没有用处的发现。这些发现可能会是一些额外洞察的源泉，比如对建模的真实关系的洞察，或者对建模过程中的问题的洞察。

7 拟合决策树模型

起初，决策树模型似乎在几乎所有的地方都运作良好。它们能对交替关系自动建模；能在事先不知道模型结构的情况下拟合复杂的模型；它们处理缺失值也广受好评；而且它们也给自己留下了易于解释的好名声。决策树这些显而易见的优点有时会隐蔽它们的一些不足，比如它们内在的不稳定性，以及它们不能够对简单的线性关系建模。

7.1 忽略决策树内在的不稳定性

回归模型的解释困难问题，是决策树模型同样要面对的，因为它非常依赖于训练数据。当输入数据哪怕只有一丁点的改动，决策树就可能从根节点生成一个不同的分叉，或者甚至选择一个完全不一样的变量来生成初始的分叉。对任何一个节点的任何变化，都会影响到后续节点，所以即使样本在本质上是同质的，也可能生成不一样的决策树。然而，决策树的总体性能还是稳定的

(Breimanet al. ,1984)。

为了克服这个问题，对从单个决策树得出的结论就不应该太看重。评估训练数据的不同样本以查看决策树模型的内在变异，这会是一个有用的方法。为了避免训练数据可能出现的过度拟合问题，使用验证数据是非常必要的，当部署一个模型时，过度拟合会导致糟糕的结果。在一些情形下，模型的关系非常模糊，就需要考虑建立几种不同的决策树，以及通过合成初始决策树的预测值来生成新的预测值。合成预测值会造成对模型的解释困难，但却可以减少决策树模型内在的不稳定。

7.2 忽略决策树的局限

决策树通过对数据集的递归分割来实现，这使得它们能够探测出复杂的非线性或者非连续的关系。不幸的是，决策树在拟合简单的线性关系或者平滑一些变动的关系时却一筹莫展。进一步说，决策树对这种关系就没能提供多少洞察。考虑一个决策树如何对这么一个从总体中抽出的样本建模，其中对取值为 0 到 1 的 x 来说，y=x。这个决策树可能在 x=0.5 处进行初始的分叉，然后得出预测说，当 x>0.5,y=0.75,而当 x<0.25 时，y=0.25。连续的递归分割将导致一个过度复杂的模型，而这个模型所阐述的关系，用一个简单的线性回归模型就可以表达。

为了克服以上问题，就需要经常考虑另外的建模策略，以提供对数据更好的拟合效果，或者更简单的表达方式。在非线性关系不明显，或者一个关系能用连续的表达式表示时，单纯依赖决策树模型会导致一个糟糕的模型。

8 拟合神经网络模型

神经网络拥有在事先不知道模型的精确结构的情况下，拟合平滑非线性模型的能力。理论上说，一个拥有一个隐层和足够隐节点的多层感知器就是一个万能拟合器(Ripley 1996)。这表明，在给定的误差区间里，一个多层感知器能够近似模拟任意一种结构。这样优秀的适应性和能力使神经网络模型大受欢迎，不过对它的误用仍然会带来一些问题，这些问题来自神经网络模型缺少对变量的选择，和/或者对神经网络的错误认识。需要提醒的是，拥有分成训练数据和验证数据的足够数据，对防止过度拟合是非常重要的。

8.1 不能进行变量选择

在拟合一系列模型方面，神经网络和决策树以及逐步回归模型没什么两样。跟它们不同的是，在系列中的每一个模型里，神经网络都使用相同的变量。不是在一个特定的模型中修正变量或者组合变量，一个神经网络模型是在中止条件满足之前，在不断的迭代过程中更新模型系数而得到。因为神经网路没有执行变量选择，所有的初始变量，即使它们对拟合模型只有一点或者就根本没有任何作用，都会出现在评分数据里。结果，评分数据就比实际所需要的大多了，这会造成额外的内存和时间浪费。

另外，神经网络模型生成了比相应的线性回归模型多的多的参数。一个拥有两个连续自变量的简单的线性回归模型只需要两个斜率参数，对应于每一个连续的自变量。但是，一个拥有一个隐层和三个隐节点的简单多层感知器实际需要 13 个估计参数。当变量的数量增加，需要的参数的数目也急剧增加，这也就增加了计算时间。

解决这些问题，重要的是在拟合神经网络模型之前进行变量选择。变量选择能剔除不必要的变量，因此能大量减少需要估计的参数的数目以及计算时间。评分数据也只需要更少的变量，它

的运行也会更有效率。因为一些选择变量的方法在本质上是线性的，这使得为模型正确地选择变量是一件有挑战的事情。有些时候，有必要在另一些模型中保留一些变量，已决定它们是否对拟合效果有所提升。如前所述，通过各种方法进行变量选择能有助于重要的变量得以进入模型。

8.2 错误地理解神经网络模型

由于其在很大程度上不能够被解释，而且在总体上也不为分析人员和业务人员所知，神经网络通常不被认为是一种建模工具。神经网络可以对其他模型所潜在的缺陷提供诊断方面的洞见，而且比较不同模型的结果有助于确定什么对提高模型的效果真正有用。

举个例子，有这么一种情形，最好的决策树模型的拟合效果非常差劲，但最好的神经网络模型和最好的回归模型在验证数据上效果相差不大。假如分析人员不考虑神经网络模型，只检验回归模型，效果也不会差多少。又考虑这么一个类似的情况，最好的决策树模型拟合地很糟糕，而最好的回归模型拟合效果稍好，但是最好的神经网络模型相比回归模型效果有很大的增进。决策树的糟糕拟合效果可能表明自变量和因变量的关系有平滑的变动。神经网络的效果比回归模型好，表明回归模型没能捕捉住自变量和因变量之间的复杂关系。没有神经网络模型的结果比较，人们就会选择回归模型，更多的解释就会由这个模型得出，而这个模型却不足以表达正确的关系。即使神经网络模型并不是呈现给最终客户或管理层的备选模型，它也能够对其他模型进行很好的诊断。

在另一种情形，最好的决策树模型和最好的神经网络模型可能运作良好，但回归模型的效果在某种程度上差些。在这种情况下，决策树模型因为其相对易于解释而可能被选中，但是神经网络的拟合效果表明决策树模型能足够总结这些关系。然而想象另一种情况，决策树模型比神经网络和回归模型效果都好。这种情形可能表明，在有缺失值的情况下，一些变量的行为对因变量是异常的。因为决策树可以直接处理缺失值，它们可以区别缺失值和对回归模型和神经网络而言是异常被剔除的值。在这种情形下，就需要更细致地考察缺失值指标，而不是只看回归模型带来的增加的模型适应性，因为神经网络模型表明，这种提高的适应能力并没有带来拟合效果的提升。

解决以上问题，就需要明智地选择变量，拟合一个神经网络模型，同时保证有足够的数据留给验证数据。如前所述，通过各种方法进行变量选择能有助于重要的变量得以进入模型。通过决策树、回归和神经网络模型来评估模型的拟合效果，能够更好地理解数据的关系，另外，利用以上信息去确认提高总体拟合效果的方法。

9 模型比较

要想正确地评估各备选模型，既需要理解如何应用分割数据集，也需要相关业务知识，以便知道模型结果是如何被应用于实际的。一些评估标准重点考察模型的整体效果，也有些只着重考虑模型在数据某一个方面上的效果。连续目标变量的模型评估相对容易理解，但是分类目标变量的评估有时会比较困难。在前面的讨论中我们已经讨论了分类目标，这一部分重点讨论二元目标。

9.1 曲解 LIFT

对于二元目标变量的预测来说经常会参考 LIFT 值，但是如果没有考虑环境变量的话，这个值有时候会大（或者小）得出奇。Lift 是组内比率与整体比率的比值。LIFT=组内发生的百分比/整体发生的百分比。

举个例子，目标整体发生概率是 50%，通过模型筛选可以找到概率为 75%的群体，那么 LIFT

值可以这样来计算，用 75%比上整体概率 50%，得到 LIFT 值为 1.5；然而我们在考虑另外一个场景：目标整体发生概率为 1%，模型提升后为 8%，这时 lift 值为 8。LIFT 值为 8 远比 LIFT 值为 1.5 要好得多。然而 LIFT 值为 1.5 对应了预测可能的 25%的提高，LIFT 8 仅仅对应了 7%的提高。而且结果的可信度是其 9 倍之多（75% VS 8%）。

当整体概率为 50%时，LIFT 值最大（提升至 100%时）是 2，然而当整体概率为 2%时，LIFT 值最大（提升至 100%时）可以达到 50。这个例子说明没有考虑整体环境，单纯的 LIFT 值是没有意义的。为了避免这个问题，要通盘考虑实际提升值和模型最终的概率（命中率）。

9.2 选择错误的评估变量

模型建立完成后，通常会对小部分高可能性客户给出模型评估统计值。从利于业务实施的角度考虑，企业通常把业务目标定位于只对前 3%的高可能性客户，这可能会比定位于更多部分客户更有效果。如果将评估关注在模型整体拟合效果或者某一部分会被忽略的人群，模型实施效果可能会很差。此外，这种策略还可以产生一个有用的模型，对少量高可能性客户处理的很好。否则，对大量客户评估，评估效果比较低，显得模型可用性不强。本质上讲，模型的效果和预测能力密切相关，而预测能力又和前面讨论的目标特征紧密相关。要想得到实施效果好的模型，必须确保在目标特征下暗含的决策规则和客户的商业决策利益一致。

10 给新数据打分

评分是大多数预测模型的最终目标。在很多情况下，用来评分的数据的数量远大于用来建模的数据的数量。当大量的数据必须被评分时，低效率的评分编码就会产生很多错误。

10.1 低效的评分编码的形成

在数据挖掘的过程中碰到几百甚至上千个变量是很普通的事情。如果在不同时间点都有值，即便输入变量数量较少也会膨胀到一个很大的数量。举个例子来说，观察一个 6 个月的序列，每月有 400 个值，那么评分会把这 400 个值变成 400*6=2400 个分值。在大多数情形下，保持较少变量数量在模型中是非常重要的。然而，评分编码可能包含不必要的编码，因为一些不必要的变量也通过了数据转换和清理。当使用决策树时这个问题会特别的显著。如果一个变量在模型中不重要但是被用于决策树中，那么这个变量可能也需要参与评分。因为每一个决策树对应一个模型，那么就会有很多模型需要被打分，而不仅仅只是需要的那个预测模型。甚至即使决策树没有被使用，那些非重要变量产生的编码也会降低处理速度，特别是这些变量数量过多时。

为了克服这个问题，生成决策树时尽量只是用重要的变量，那些计算变量尽量不去用。例如，当某一指标发生缺失值，那么这个缺失值生成计算变量大多数情况下是零，而不是其他值。因此，大多数情况下，这样的变量参与的决策树会产生大量的多余代码进而只能提供一个较次的结果。为了避免这个问题，首先要进行数据处理和变量选择，然后使用只包含重要变量的数据集创建模型，这样才可能得到有效率的代码。

10.2 忽视模型的效果

为避免模型使用训练集数据过度拟合，校验集数据是非常有用。然而，校验集的效果往往对模型的选择具有直接作用，导致校验集效果总是优于总体数据的结果。当训练集和测试集被过度抽样时，这种影响更加明显。为了调整过度抽样数据得到的预测概率，可以规定谁优先的方法。

当然，衡量模型结果最好的方法是使用实际数据检验。

在许多情况下，引入一个代理的目标变量很有用，这样被模型化的目标就不会和实际关注的结果完全相关。

我们来看这样一个例子，一家公司要对一个产品进行促销，为了使促销更有效率，这家公司需要了解最有可能对这项产品促销做出反应的目标客户群。因此，需要建立一个“偏好模型”预测当前的消费者是否会购买这个产品，或者需要建立一个“反应模型”预测消费者是否会对促销做出反应。无论是哪个模型，目标变量和实际关注的结果都是有差异的。在“偏好模型”中，差异非常明显，因为当前购买该产品的消费者不一定会对该产品的促销做出反应。在“反应模型”中，这种差异更加微妙，对过去的促销有反应的消费者也不一定对现在的促销感兴趣，因为很可能是和以前完全不同的促销。即使是一样的，随着时间的变化，消费群体也发生了变化。因为一般都缺乏反应数据，“偏好模型”看起来是最佳方式。但对打分数据的效果最终还是可能比训练集或校验集的差，因为模型目标不是实际关注的目标。

这是大多数建模时会遇到的问题，因而对实际模型的效果进行监控显得非常重要。用总体数据建模的实际效果提供了一条基准线，模型不断优化直到个真实的反应模型产生。因为产品存在一个生命周期的问题，对一个产品产生兴趣的新进消费群可能和之前的消费群有很大差别，即使这些消费群体都对这项产品感兴趣，但不能保证所有的消费群体都会对这个产品的促销感兴趣。此外，清楚每个消费群体中有多少人可能会做出反应也是非常重要的。解决这些问题的唯一方法是保留一定数量的候选，能够使你既可以监控目标群体的结果，也可以监控控制群体的结果。两个群体的不同反应可以让你确定哪些消费者会对你的促销兴趣最大。

11 数据聚类

聚类通常是分析新数据的早期步骤之一，希望籍此获取数据的群组信息，而通过这些群组可以帮助识别特定的子群以优化想要的结果。聚类最初用于处理多元（数值的）数据，而通过一些技术手段，类别数据也可以进行聚类合并。应用聚类分析常犯的错误是试图去拟合一个简单的聚类模型，或者在聚类中包含太多的类别变量。

11.1 构建一个聚类方案

聚类没有什么正确或错误。使用不同的变量会生成不同的群。最好的聚类方法是由商业目标所决定的。预测模型提供了划分群组更为适当的方法，这些群组最大程度地划分了数据，同时兼顾了目标输出。聚类只是简单的试图找到按照输入量和定义的距离来进行的自然分组。

在很多案例中，分析师尝试用所有变量来聚类，期望通过算法来识别变量重要与否。这种方法的问题在于其实只有少量变量对生成聚类结果有重大影响，很多附加变量对生成聚类结果作用不大，反而影响解释那些对聚类结果有重大影响的变量是如何生成聚类结果的。此外，聚类采用“组间距离最大化”来划分群组，因而生成聚类结果的那些变量间通常具有较大的差异。这听起来似乎不错，然而挑战在于不同的聚类结果表达了不同类型的信息。生成的聚类结果难于解释，难以通过必要的商业标准来评估聚类结果的价值。

为了克服这一问题，考虑通过一些有业务意义的、相关的数据子集来建立一系列的聚类方案。因为不同的变量集合生成不同的聚类结果，利用相关变量的子集提供了一系列有意义的、可解释的聚类。这些不同的变量集合生成的聚类反映了个体的不同侧面。例如，一个聚类模型的重点可能在于不同的购买模式，而其他聚类模型集中在人口统计有关的数据或者是地理数据。另外一些聚类模型可能关注顾客忠诚度标准或者购买类型。这种聚类结果更易于解释，而且不同的聚类方

案可用于不同商业目的。

11.2 包含（很多）类别变量

如前所述，聚类方法最初用于分析多元连续数据。通过定义不同类别层次间的距离也可以将聚类用于类别数据。一方面，这种定义生成了可以最大化划分数据的变量（如用 0 和 1 表达性别这种二元变量，所以除了这两个值之外没有任何介于 0 和 1 之间的值），这些变量是聚类所需要的。另一方面，这些变量已经包含了类别信息，用数值数据替换类别数据以便生成类别意义不大。在很多案例中，类别变量促成了大量聚类的形成，但是仅着眼于将特定类别变量与存在的连续变量结合，会使聚类结果解释不清。

为了克服这个问题，考虑只用数值变量聚类。如果你想探索基于类别数据的特定数据子集，利用感兴趣的变量分组来建立剖面，然后采用有重要成员的分组变量进行聚类分析。该方法使聚类结果更易解释，也更易评估基于不同商业目的的聚类方案。

12 进行关联和序列分析

关联和序列分析都是寻找、识别在同一交易（关联）或订单的各项之间关联性的描述性挖掘技术。因为每项与任意其他项之间可能的组合都应该被考虑，这种相对简单的描述性技术可能很耗时。应用这项技术的失误主要是由于数据准备不充分引起的。

12.1 数据集排序错误

为了应用关联或者序列分析，交易型数据集是必须的。如果数据表不是通过 ID 排序，那么需要先排序。交易型数据集可能比较大，所以排序过程需要较大的运算空间会导致过多的处理时间。为了克服这个问题，如果不希望额外的处理时间，那么进行关联或序列分析之前要先对数据集排序。另外，确保系统选项中允许利用所有可获取的内存进行运算。

12.2 输出数量管理的错误

在 SAS Enterprise Miner 5 之前，管理感兴趣的关联项目的数量是很重要的，这是因为处理时间增长远远快于可能的兴趣条目的增长。在当前的版本中，用户可以用关联或者序列分析处理高达 100000 条数据。经验表明，当处理大量数据的时候，那些非平凡项出现的次数相对较少。当分析大量数据的时候，与采用“不正确的数据准备或者准备分类预测失败”一节中提及的数据准备方法相比，需要花费更多的时间进行数据准备。以相对频繁项为中心能极大地提升处理速度。

结论

本文包括一组策略，用于辅助分析员识别并纠正常见的数据挖掘错误。正如早先提到的，这里建议的某种方法不一定适合特定的分析场景。不过，这些方法已经在一些广泛的业务场景中得到有效应用。正确应用这些技巧，将会缩短处理时间并提高结果模型的效用。

参考文献

Ambler, G., A. R. Brady, and P. Royston. 2002. “Simplifying a Prognostic Model: A Simulation Study Based on Clinical Data.” Statistics in Medicine. 21:3803 – 3822.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. Classification and Regression Trees. London: Chapman and Hall.

Derksen, S. and H. J. Keselman. 1992. “Backward, Forward, and Stepwise Automated Subset Selection Algorithms: Frequency of Obtaining Authentic and Noise Variables.” British Journal of Mathematics and Statistical Psychology 45:265 – 282.
Muller, K. E. and V. A. Benignus. 1992. “Increasing Scientific Power with Statistical Power.” Neurotoxicology and Teratology. 14:211 – 219.
Ripley, B. D. 1996. Pattern Recognition and Neural Networks. Cambridge: Cambridge University Press.

感谢

我要感谢 Ross Bettinger，Jay King，Bob Lucas，Craig DeVault，Jim Georges 和 Annette Sanders，感谢他们的审稿并提供了可贵的意见。我还要特别感谢 Ross Bettinger 的贡献，感谢她本文几次草稿的复审。接着，我还要感谢 Will J.E.Potts，感谢它为目标轮廓章节撰写了很多决策原理。当然，我还要感谢领导、组织、党和国家的大力支持，最后我还要感谢 ttnn 的小伙子们，包括 Jiangtang Hu，Delin He，Richard Zhang，Hunter Dong， Xining Ding，兰德里尼，Qing，Jeff Yang，感谢他们将此文翻译成中文。

联系信息

您的评论和质疑都是可贵并让人鼓舞的，请联系作者：

Doug Wielenga
100 SAS Campus Drive
Cary, NC 27513
E-mail: Doug.Wielenga@sas.com

译者信息

Jiangtang Hu（jiangtanghu@gmail.com）
Delin He（beijinhe@gmail.com）
Richard Zhang（richard_zzh@hotmail.com）
Hunter Dong（hunterdong@gmail.com）
Xining Ding（happyjava.dennis@gmail.com）
兰德里尼（zw8048@hotmail.com）
Jeff Yang（gnayyoung@gmail.com）
Qing（happyscry@gmail.com）

《ttnn BI 观点》 http://groups.google.com/group/ttnn