特征工程可以解决什么样的问题和构建迭代过程说明

搜索与推荐Wiki 2020-06-12

> 原文链接：点击最下方【阅读原文】

特征工程可以解决什么样的问题？

特征工程是一个非常重要的课题，是机器学习中不可缺少的一部分，但是它几乎很少出现于机器学习书本里面的某一章。在机器学习方面的成功很大程度上在于如果使用特征工程。在机器学习中，经常是用一个预测模型（线性回归，逻辑回归，SVD等）和一堆原始数据来得到一些预测的结果，人们需要做的是从这堆原始数据中去提炼较优的结果，然后做到最优的预测。这个就包括两个方面，第一就是如何选择和使用各种模型，第二就是怎么样去使用这些原始的数据才能达到最优的效果。那么怎么样才能够获得最优的结果呢？

贴上一句经典的话就是：Actually the sucess of all Machine Learning algorithms depends on how you present the data. —— Mohammad Pezeshki

直接翻译过来便是：事实上所有机器学习算法上面的成功都在于你怎么样去展示这些数据。由此可见特征工程在实际的机器学习中的重要性，从数据里面提取出来的特征好坏与否就会直接影响模型的效果。从某些层面上来说，所使用的特征越好，得到的效果就会越好。所需要的特征就是可以借此来描述已知数据的内在关系。

总结一下就是：Better feature means flexibility. Better feature means simpler models. Better feature means better results.

有的时候，可以使用一些不是最优的模型来训练数据，如果特征选择得好的话，依然可以得到一个不错的结果。很多机器学习的模型都能够从数据中选择出不错的结构，从而进行良好的预测。一个优秀的特征具有极强的灵活性，可以使用不那么复杂的，运算速度快，容易理解和维护的模型来得到不错的结果。

什么才是特征工程？

Feature Engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. —— Jason Brownlee

Feature Engineering is manually designing what the input x's should be. —— Tomasz Malisiewicz

从这个概念可以看出，特征工程其实是一个如何展示和表现数据的问题，在实际工作中需要把数据以一种“良好”的方式展示出来，使得能够使用各种各样的机器学习模型来得到更好的效果。如何从原始数据中去除不佳的数据，展示合适的数据就成为了特征工程的关键问题。

特征有用性的预估

每次构造了一个特征，都需要从各个方面去证明该特征的有效性。一个特征是否重要主要在于该特征与要预测的东西是否是高度相关的，如果是高度相关，那么该特征就是十分重要的。比如常用的工具就是统计学里面的相关系数。

特征的构造过程

在实际工作中首先肯定要确定具体的问题，然后就是数据的选择和准备过程，再就是模型的准备和计算工作，最后才是展示数据的预测结果。构造特征的一般步骤：

[1] 任务的确定

根据具体的业务确定需要解决的问题。

[2] 数据的选择

整合数据，收集数据。这个时候需要对这些数据的可用性进行评估，可以分成几个方面来考虑。获取这批数据的难易程度，比方说有的数据非常隐私，这批数据获得的难度就很大。其次就是这批数据的覆盖率。比方说要构造某个年龄的特征，那么这些用户中具有年龄特征的比例是多少就是一个关键的指标。如果覆盖率低，那么最后做出的特征可以影响的用户数量就会有限制。如果覆盖率高，那么年龄特征做得好，对最后的模型训练结果都会有一个明显的提升。再就是这批数据的准确率，因为从网上或者其他地方获取的数据，会由于各种各样的因素（用户的因素，数据上报的因素）导致数据不能够完整的反映真实的情况。这个时候就需要事先对这批数据的准确性作出评估。

[3] 预处理数据

设计数据展现的格式，清洗数据，选择合适的样本使得机器学习模型能够使用它。比方说一些年龄特征是空值或者负数或者大于200等，或者说某个页面的播放数据大于曝光数据，这些就是数据的不合理，需要在使用之前把这一批数据排除掉。这个时候清洗异常样本就显得至关重要。

[4] 特征的构造

转化数据，使之成为有效的特征。常用的方法是标准化，归一化，特征的离散化等。

4.1 标准化（Standardization）

比方说有一些数字的单位是千克，有一些数字的单位是克，这个时候需要统一单位。如果没有标准化，两个变量混在一起搞，那么肯定就会不合适。

4.2 归一化（Normalization）

归一化是因为在特征会在不同的尺度下有不同的表现形式，归一化会使得各个特征能够同时以恰当的方式表现。比方说某个专辑的点击播放率一般不会超过0.2，但是专辑的播放次数可能会达到几千次，所以说为了能够在模型里面得到更合适结果，需要先把一些特征在尺度上进行归一化，然后进行模型训练。进行比例缩放或者归一化的时候，通常采取的方案有线性（linearization）最大最小归一化，对数（logarithm）归一化，或者 z-score 的归一化。其中 z-score 的归一化指的是 $(x-\mu)/\sigma$，这里 μ 是这个特征的均值，σ 是这个特征的方差。这里的归一化的关键之处在于数据的变化（Data Transforming）。对于处理一些大尺度数据（比方说某个视频被所有用户观看的次数之类的），一般会使用对数来处理数据，或者双曲线函数。例如：

$$f(x) = \log(1+x) : [0, +\infty) \to [0, +\infty),$$

或者

$$f(x) = x/(x+1) : [0, +\infty) \to [0, 1).$$

4.3 特征的离散化（Discretization）

离散化是指把特征进行必要的离散处理，比方说年龄特征是一个连续的特征，但是把年龄层分成5－18岁（中小学生），19－23岁（大学生），24－29岁（工作前几年），30－40岁（成家立业），40－60岁（中年人）从某些层面来说比连续的年龄数据（比如说某人年龄是20岁1月3日之类的）更容易理解不同年龄层人的特性。这里可以根据特征的分布图像做出不同的分割，比方说通过等频率分割（Equal-Frequency）得到的特征比等区间分割（Equal-Interval）得到的特征具有更好的区分性。典型的离散化步骤：对特征做排序－> 选择合适的分割点－> 作出对整体的分割 －> 查看分割是否合理，是否能够达到停止条件。

4.4 特征的二值化（Binarization）

二值化指的是把某个特征映射成 0 或者 1 两个值，比方说判断某个用户是否收听某个节目，用 1 表示该用户收听这个节目，否则用 0 表示该用户没有收听这个节目。

4.5 特征的交叉

比方说，某个男性用户在一级分类新闻资讯的取值是1，那么他在这个新闻咨询下的所有专辑和节目的取值都是1。对娱乐新闻的取值也是1，对八卦头条的取值也是1，对NBA战况直播的取值还是1。也就是说（男性，娱乐新闻），（男性，八卦头条），（男性，NBA战况直播）这三个分类的取值都是1。这样看起来就不符合常理，照理说男性用户对NBA的兴趣应该是远大于娱乐新闻的。比较合理的做法是男性在某个专辑的取值是所有男性在这个专辑的点击率，也就是说（男性，娱乐新闻）的取值是男性对娱乐新闻的点击率，（男性，NBA战况直播）的取值是男性对NBA战况的点击率。此时的值不再是 0 或者 1，而是 [0,1] 之间的某个实数，可以对这个实数进行加减乘除等运算操作。除了性别和点击率的交叉特征，还可以进行年龄，地域，收入等特征和点击率的交叉。

[5] 选择特征的常用方法

5.1 过滤（Filter）

过滤这种方法是选定一个指标来评估这个特征的可行性，根据指标值来对已经构造的特征进行排序工作，去掉无法达到要求的特征。这个时候，选择一个合适的指标来判断特征是否有效就是关键所在。从统计学的角度来看，相关系数（Correlation Coefficient）是一个评价两个随机变量 X 和 Y 是否线性相关的一个重要指标。

$$\rho_{XY} = cov(X,Y)/(\sigma_X\sigma_Y) \in [-1,1]$$

就是相关系数的计算方法。如果 $\rho_{XY} < 0$，那么说明两个变量是线性反相关的；如果 $\rho_{XY} > 0$，那么说明两个变量是线性相关的。不过需要主要的是，即使 $\rho_{XY} = 0$，也只是说明两个变量是线性无关的，并不能推出它们之间是独立的。此时知道的就是一个线性分类器并不能把这个特征的正负样本分开，需要把该特征和其他特征交叉或者做其余的特征运算，形成一个或者多个新的特征，让这些新的特征发挥新的价值，做好进一步的分类工作。

5.2 包装（Wrapper）

包装这种方法和前面的过滤方法不一样。包装方法需要首先选定一种评估模型效果的方法，比方说 AUC（Area Under the Curve），MAE（Mean Absolute Error），Mean Squared Error（MSE）等。此时有两个不同的方案，分别是前向特征选择（Forward Feature Selection）和后向特征选择（Backward Feature Selection）。前向特征选择是从空集（Empty Set）开始，使用一种贪心算法，每次在现有特征的基础上逐渐添加一个使得模型效果更好的特征。反之，后向特征选择是从（Full Set）开始，每次去掉一个让模型效果提升最多的特征。或者可以使用增 L 去 R 算法（Plus-L Minus-R Selection），也就是说从空集开始，每次增加 L 个，同时减去 R 个，选择最优的特征，其中 L>R。或者从全集开始，每次减去 R 个，同时增加 L 个，选择最优的特征，其中 L<R。

5.3 嵌入（Embedding）

嵌入特征选择方法和算法本身紧密结合，在模型训练的过程中完成特征的选择。例如：决策树（Decision Tree）算法每次都是优先选择分类能力最强的特征；逻辑回归（Logistic Regression）算法加上 L1 和 L2 等惩罚项之后，也会使得某些信号比较弱的特征权重很小甚至为 0。至于 Logistic Regression 的一些算法，可以参考 TG（Truncated Gradient Algorithm），FOBOS（Forward and Backward Splitting），RDA（Regularized Dual Averaging Algorithm）， FTRL（Follow-the-Regularized-Leader Algorithm） 算法，参考文献：Ad Click Prediction: a View from the Trenches。

[6] 模型的使用

创造模型，选择合适的模型（LR，SVM，SVD等），用合适的模型来进行预测，用各种统计指标来判断该特征是否合适。

[7] 上线的效果

通过在线测试来看效果。数据的转换（Transforming Data）就是把数据从原始的数据状态转换成适合模型计算的状态，从某些层面上来说，“数据转换“和”特征构造“的过程几乎是一致的。

特征工程的迭代过程

特征工程的迭代步骤

[1] 选择特征

需要进行头脑风暴（brainstorm）。通过分析具体的问题，查看大量的数据，从数据中观察出数据的关键之处；

[2] 设计特征

这个需要具体问题具体分析，可以自动进行特征提取工作，也可以进行手工进行特征的构造工作，甚至混合两种方法；

[3] 判断特征

使用不同的特征构造方法，来从多个层面来判断这个特征的选择是否合适；

[4] 计算模型

通过模型计算得到模型在该特征上所提升的准确率。

[5] 上线测试

通过在线测试的效果来判断特征是否有效。监控重要特征，防止特征的质量下滑，影响模型的效果。