**一 可解释性概述**

**1 什么是可解释性**

在机器学习领域，可解释性是指对一个算法的原理或者对得出结果的依据做出解释。就像我们想要知道模型对未见样本 x 给出预测 y = f ( x ) 的原因。有些模型本身就是可解释的，比如逻辑回归、决策树、朴素贝叶斯等；有些模型则很困难解释，比如神经网络。

**2 为什么需要可解释性**

算法工程师：在算法构建的过程中，工程师需要根据实际的应用场景选择不同的框架和网络结构来获得更高的精度和泛化能力，那么他们需要进入网络的内部结构，研究不同的成分的功能并且对整体效用的影响。这时候的可解释性针对于网络结构，用于提高网络性能。（BERT低层 捕捉的是短语结构方面的信息，单词和字的特征表现在3-4层，句法信息的特征表现在6-9层，句子语义信息的特征表现在10-12层，主谓一致的特征信息表现在8-9层）

产品使用者或受用者：算法构建出来后，需要被接纳被使用，那么黑盒模型中的不可见性便是可能存在的风险，针对一些决策失误风险较高的领域如医疗、金融，构建一个诊断和投资决策的推荐系统，可解释的决策机理是使用者更加关注的点，他们期望知道得出决策是依据什么信息经过怎样的逻辑推导，以保障他们采用模型的信心。

领域专家：传统的领域专家需要了解某个领域的知识后，针对不同的应用需要去构建有效决策依赖的特征和相应的规则，当模型具有可解释性后，他们可以通过研究高效模型获取其依赖的特征和规则，这往往是启发性的知识，并可以启发该领域后期的研究。

**3 可解释性分类**

可解释性总体可以分为 “事前可解释性” 和 “事后可解释性”

**3.1 事前(ante-hoc)可解释性**

指通过训练结构简单、可解释性好的模型或将可解释性结合到具体的模型结构中的**自解释模型**使模型本身具备可解释能力。

**3.2 事后(post-hoc)可解释性**

指通过开发可解释性技术解释已训练好的机器学习模型。根据解释目标和解释对象的不同，post-hoc可解释性又可分为全局可解释性(global interpretability)和局部可解释性(local interpretability)。全局可解释性旨在帮助人们理解复杂模型背后的整体逻辑以及内部的工作机制，局部可解释性旨在帮助人们理解机器学习模型针对每一个输入样本的决策过程和决策依据。

**3.2.1 全局可解释性**

一般指模型告诉我们它眼中各类标签对应的样本应该具有什么样的特征。具体来说，可以利用可理解的规则集合对模型生成可解释的符号描述；或从难解释模型中提取一个可解释模型；又或者我们把模型结果定住，反向找哪些输入能使得结果最显著，从而获取机器眼中某类对象的总体特征（对象甚至可以是抽象的概念）。下图是神经网络模型中代表 “happy” 和 “serious” 的图像特征

|  |  |
| --- | --- |
| 截图.png | 截图.png |

**3.2.2 局部可解释性**

一般以输入样本为导向，试图分析某个输入样本的每一维特征对模型最终决策结果的贡献。经典的局部解释方法包括敏感性分析解释、局部近似解释、梯度反向传播解释、特征反演解释以及类激活映射解释等



参考链接：

[1] <https://blog.csdn.net/wxc971231/article/details/121184091>

[2] <https://zhuanlan.zhihu.com/p/428131762>

[3] <https://christophm.github.io/book/>

**二 可解释性算法**

**1 基于模型梯度（IG）**

简单来说就是对于给定的输入x以及模型F(x)，我们想办法指出x的哪些分量对模型的决策有重要影响，或者说对x各个分量的重要性做个排序，用专业的话术来说那就是“归因”。一个朴素的思路是直接使用梯度∇xF(x)来作为x各个分量的重要性指标，而积分梯度是对它的改进。这里应用了事后可解释性-局部解释-敏感性解释的思想，可以对某个特征进行删除、添加、或者增加扰动来观察对结果的影响程度。

**公理**：当归因结果与人类经验相异的时候我们很难去判断出到底是确实是算法模型的问题还是归因技术本身的问题，因此需要设定可解释性算法必要满足的公理，由此用满足公理的技术进行归因得到的结果必定是模型所应有的。

**基线值：**当我们把一件事情的发生归结于某个原因的时候，我们就可以把没有该原因的情况当做基线值，对于一个图像识别系统来说，基线值可以是一张全黑的图片；而对于一个NLP系统而言，基线值可以是全部值为0的词向量。

**1.1 两个公理**

**敏感度（Sensitivity）**

如果针对一个特征，当输入非基线值的时候应当获得有状态的预测（基线值对应无状态）；当选取两个不同的输入值（相对于基线值而言）时，模型应该输出不同的预测，这样子该特征将被赋予非零归因，简而言之就是梯度处处不为零

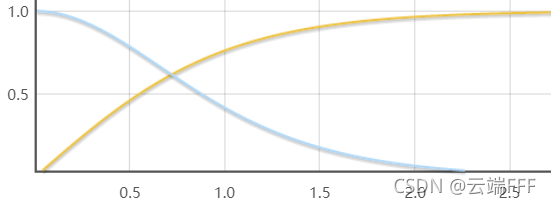
**实现不变性（Implementation Invariance）**

尽管模型内部实现不一致，但对于所有的输入，输出结果都是一致的两个模型被称为功能等效。这样对两个功能等效的模型进行归因分析时，获取到相同的归因 满足实现不变性，就可以将得到的输出责任归结于输入，而和模型内部结构无关。如果不能保持实现不变性的话，对于两个相同的网络，归因方法却会给出两个不同的值，这显然是不对的。

**1.2 传统梯度和反向传播的归因**

**梯度归因（Gradients）**

使用梯度表示输入特征的重要性，但对于一些非线性函数不可避免的存在梯度饱和阶段，梯度趋于0，违背敏感性原则，此时梯度趋于0的特征被认为是归因比重不大的因素，但可能并不是这样的，举个例子，假设我们要判读某张图片是否为大象的图片，显然鼻子长度是一个重要的指标。但是对于已经有长鼻子的大象来说，对其鼻子长度增长或缩短一点对于预测结果而言已经意义不大了。



纵轴是模型将图像分类为大象的概率f(x)，横轴是大象鼻子长度xi，黄色线代表随着鼻子长度增大，分类为大象的概率不断上升并收敛到1，蓝色线为黄色线的梯度，可见，当黄线处于饱和区，其对应的梯度接近于 0，如果只看梯度，这种情况下会得出鼻子长度无关紧要的错误结论。

对于实现不变性上面的例子，当我们假定的非线性函数处处可微时，就可以使用链式法则，而链式法则本质上是满足实现不变性的。

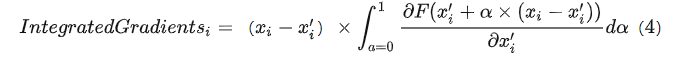
**基于反向传播的归因**

基于反向传播的归因技术中，有DeepLift和Layer-wise relevance propagation (LRP)这两种技术通过设置跨度较大的离散值来保障梯度处处存在，所以满足了敏感性原则，但是由于离散函数不满足链式法则，故函数内部的实现细节不可忽略，所以不满足实现不变性。

**1.3 积分梯度算法（Integrated Gradients）**

积分梯度算法的结合了直接梯度和基于反向传播的归因技术DeepLift和LRP的分而治之的设计思想，满足敏感性和实现不变性的公理。

设输入为x，基线值为x‘ ，函数映射表示为F，对输入的第i个维度求积分梯度可以表示如下：



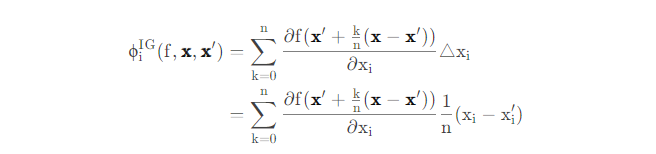
从上式的表达可以看出积分梯度算法仅仅考虑了模型的输入输出且函数处处可微，不需要模型内部细节的参与，因此积分梯度算法是满足实现不变性的.而对于敏感型的公理，文章直接提出了一个新的公理为**完整度（Completeness）**，并且**直白地**说完整度是敏感型的更高实现，其本身蕴含着敏感性。

**完整度（Completeness）**

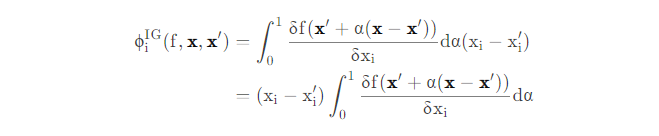
归因技术以积分的方式加和了从基线值到输入间的输出值称为满足完整度。直接梯度是选取当前输入一个点进行输入输出的归因，当输入的特征值刚刚好处于梯度饱和阶段时，获得的归因结果在该特征上的归因比重往往微乎其微，但该输入特征并非不重要，这其实才是敏感性真正要解决的问题。而积分梯度通过在基线值和输入值之间选取了无限多个积分点进行积分加和即满足完整度，这时并不拘泥于梯度饱和阶段而更像是对x-x'整体进行均衡加权归因，这样便巧妙地解决了梯度饱和问题，因此说完整度是蕴含了敏感性，是其更高的实现。

**公式理解**

在x'和x间进行插值，插值数为n，特征xi的重要性如下，而插值路径可以有多种选择，这里直接用线性插值



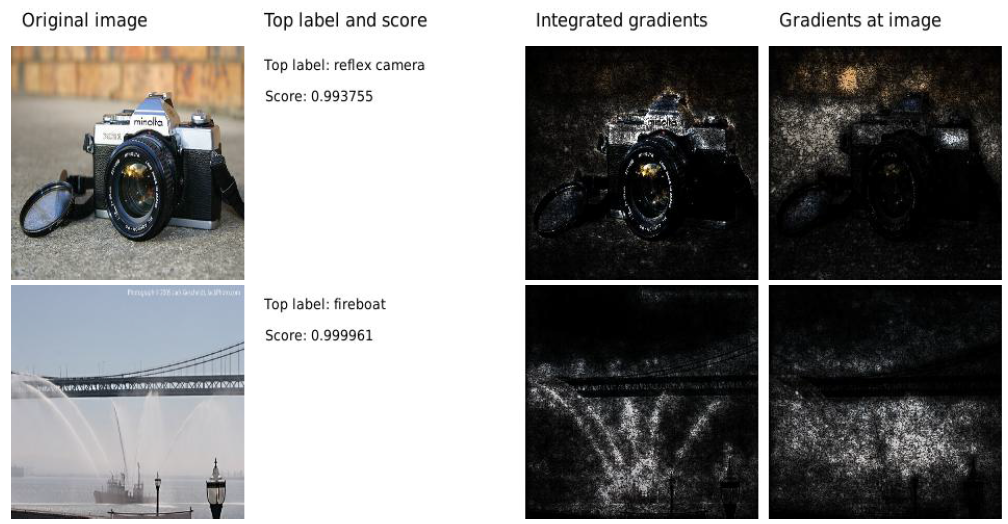
当n趋于∞时，

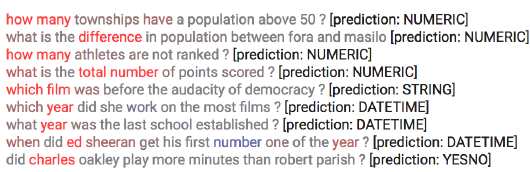


插值路径

截图.png

**1.4 结果**





参考链接：

[1] <https://blog.csdn.net/wxc971231/article/details/121184091>

[2] <https://zhuanlan.zhihu.com/p/428131762>

[3] <https://zhuanlan.zhihu.com/p/365815861>

[4] <https://arxiv.org/abs/1703.01365>

[5] <https://www.spaces.ac.cn/archives/7533>

**2 基于线性模型（LIME）**

LIME全称Local Interpretable Model-Agnostic Explanations，在2016年《“Why Should I Trust You?” Explaining the Predictions of Any Classififier》这一文中提出来的，是一种事后解释方法，即在建立模型之后所做的解释，其名称也很好的反应了它的特点：

Local: 基于想要解释的预测值及其附近的样本，构建局部的线性模型或其它可解释模型；

Interpretable: LIME做出的解释易被人类理解。利用局部可解释的模型对黑盒模型的预测结果进行解释，构造局部样本特征和预测结果之间的关系；

Model-Agnostic: LIME解释的算法与模型无关，无论是用Random Forest、SVM还是XGBoost等各种复杂的模型，得到的预测结果都能使用LIME方法来解释；

Explanations: 人类可解释的特征。LIME是一种事后解释方法。

LIME可处理不同类型的输入数据，如表格数据(Tabular Data)、图像数据（Image Data）或文本数据(Text Data)。对于表格数据，如用银行客户行为数据预测理财产品销售，训练完复杂模型后可以用LIME得到哪些特征影响理财产品销售；图像数据，如识别图片中的动物是否为猫，训练完复杂模型后可以用LIME得到图片中的动物被识别为猫是因为哪一个或几个像素块；文本数据，如识别短信是否为垃圾短信，训练完复杂模型后可以用LIME得到一条信息被判断为垃圾短信是因为哪一个或几个关键词。

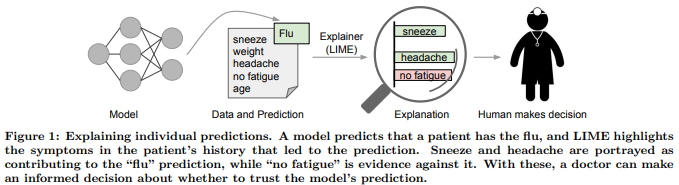
**2.1 引言**

LIME:部署在真实场景时，评价指标不一定真的科学，需要可解释性模型；

SP-LIME还可以从海量数据集中挑选异常样本，有代表性的样本来反应全局可解释性。

实验： 从不同的模型挑选更优秀的模型，通过特征工程改进模型

**2.2 LIME基本假设**



对于文本或是图像，可解释特征是什么，词和图像块（可解释性元素）

单个样本的可解释性：提升人类对算法模型的接受程度

解释整个模型：选取有代表性的样本反应全局的可解释性

数据泄露：一种是完全无关的特征加入到了数据中，一种是高度相关的类似标签加入到了数据中

比较模型：准确率不能说明一切

**可解释性**：

要求解释器的模型与特征都必须是可解释的，像决策树、线性模型都是很适合拿来解释的模型；而可解释的模型必须搭配可解释的特征，才是真正的可解释性，让不了解机器学习的人也能通过解释器理解模型。

**局部保真度**：

灰色直线和原模型曲线局部尽量重合

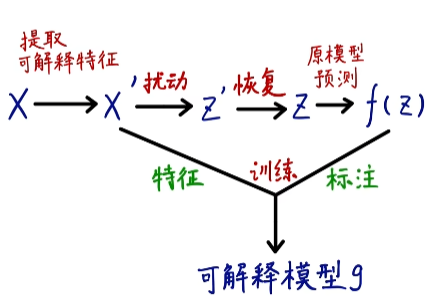
既然我们已经使用了可解释的模型与特征，就不可能期望简单的可解释模型在效果上等同于复杂模型（比如原始CNN分类器）。所以解释器不需要在全局上达到复杂模型的效果，但至少在局部上效果要很接近，而此处的局部代表我们想观察的那个样本的周围。

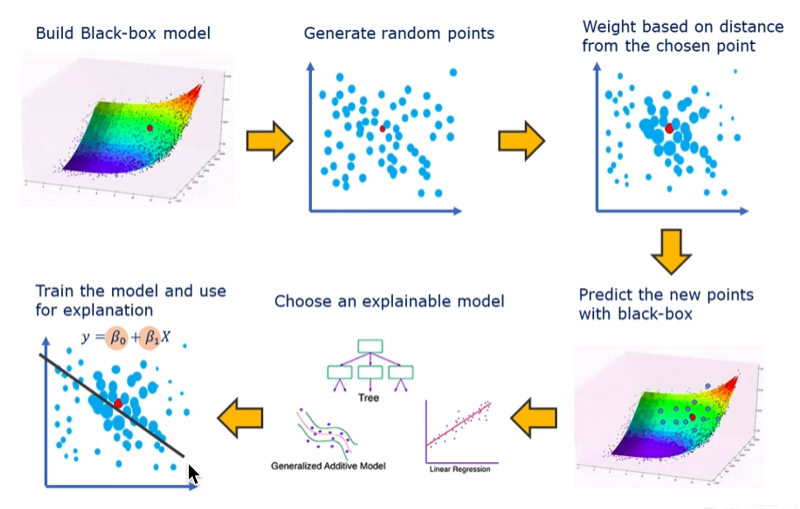
**与模型无关**：

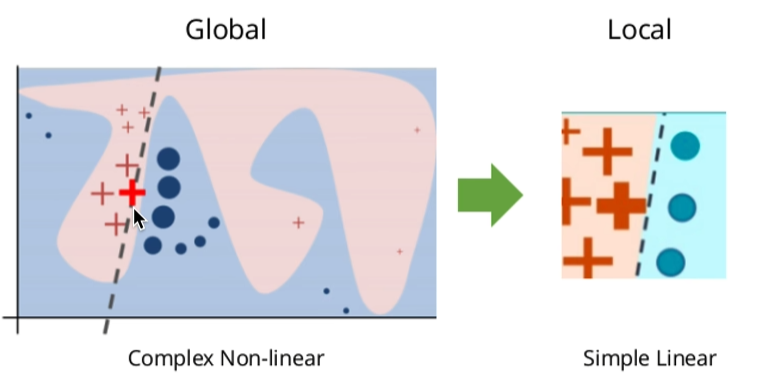
这里所指的是与复杂模型无关，换句话说无论多复杂的模型，像是SVM或神经网络，该解释器都可以工作。

**2.3 原理**

LIME是一种通用的可解释性的范式，而不是某一种算法。总结一句话:使用“可解释特征”训练“可解释模型” 在“特定样本的局部线性邻域” 拟合“原模型”



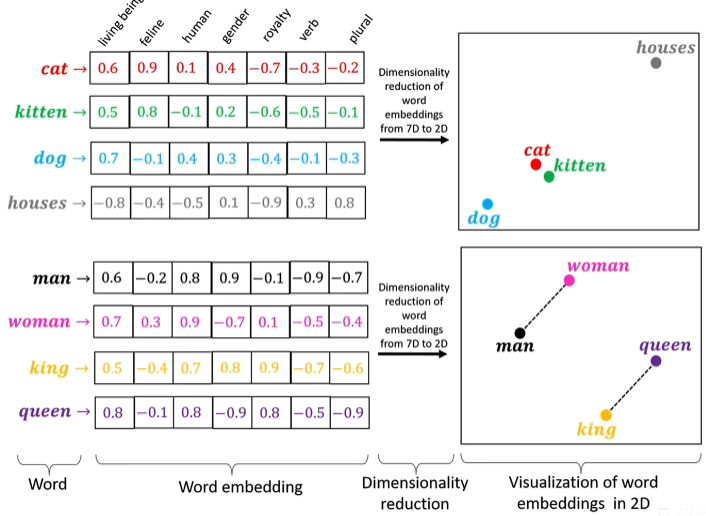




全局非线性但是可以使用局部线性去拟合

**2.3.1 文本数据可解释性分析**

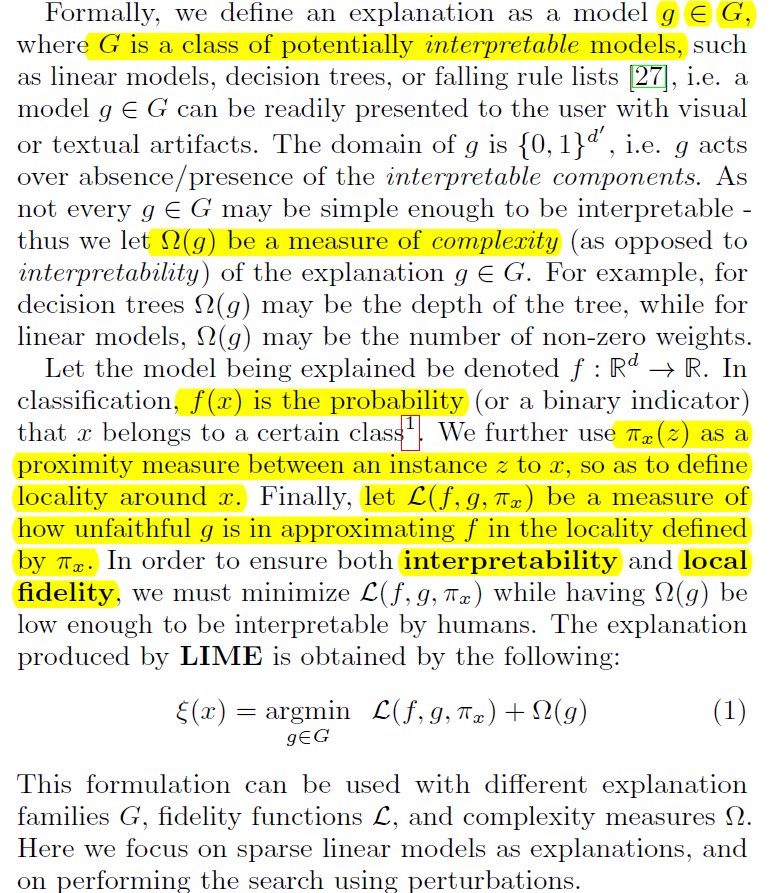
词向量是人类不可解释的，词向量可能几十维，上百维，没人知道每个维度具体指什么意思，这就无法和人类的常识对应，可以使用词袋模型，每句话就可以变为一个向量，这个向量就可以作为可解释特征





**2.3.2 保真度-可解释性权衡**

权衡可解释性模型复杂度和拟合能力



G：不同复杂度的可解释性模型

g：某一个可解释性模型

f(x)：原模型预测概率

πx(z): 新生成的样本z距待测样本x的距离

L(f,g,πx): 局部不可信度，越小，可解释性模型越贴近原始模型(local fidelity)

Ω(g)：可解释性模型的复杂度，是否被人类理解（interpretability）

最后的目标函数：既要可信又要通俗理解

**2.3.3 稀疏线性可解释模型**

截图.png

L(f,g,πx)：训练线性模型均方误差损失函数

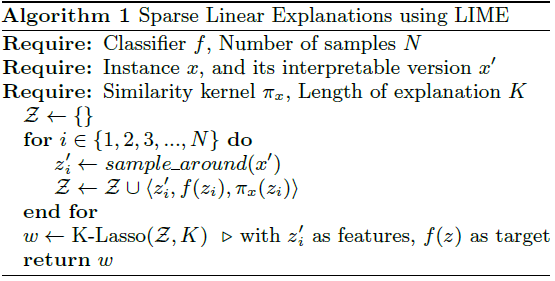
z：新生成的样本

z'：可解析特征

πx(z): 样本z距待测样本x的距离

f(z)：原模型预测结果

g(z')：线性模型预测结果



采集领域样本，构建可解释性特征

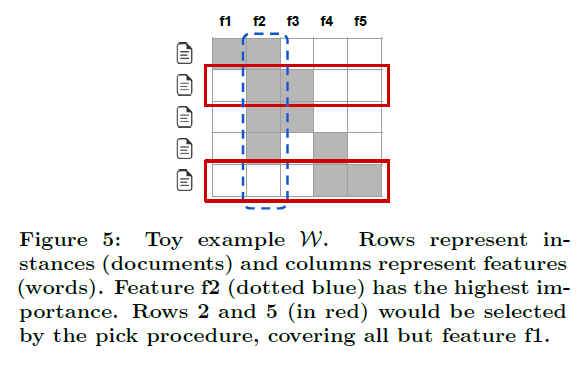
LIME最难的步骤，如何构造可解释性特征，仍然需要人工先验知识，而且也有个缺点，原模型在局部邻域仍然非线性用可解释模型无法很好拟合。

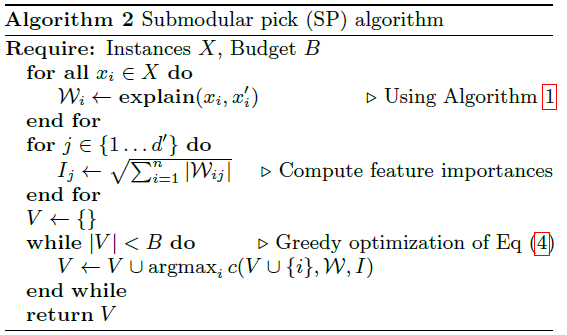
**2.4 选取有代表性样本**

怎样解释模型，就是选取有代表性的样本，不能只看原始数据和预测结果

1 计算每个特征在所有样本中的代表性（直观理解哪个特征出现的多）

2 从包含代表性特征的样本中选择样本，尽量覆盖所有特征，每个样本特征尽量不要冗余



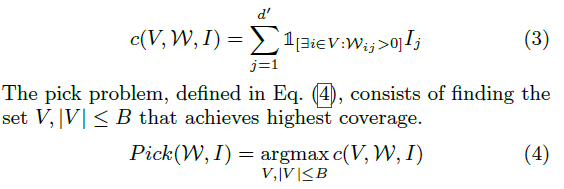


算法2:

获取数据集每个样本每个可解释特征权重

计算每个可解释特征的“代表性”

找到使得可解释特征代表性求和最大的样本



c(V,W,I)：特征覆盖度

Pick(W,I): 找到能使c最大化的样本集V

**2.5 总结**

优点：

1 通用性，任何复杂模型 都可使用LIME进行解释

2 给出特征重要性，可以改进特征工程

3 对扰动数据使用可解释性模型预测原模型的预测结果

4 可解释单个样本预测结果，选取代表性样本

5 人工设计，构造可解释性特征

缺点：

1 人工设计，构造可解释性特征不一定科学

2 局部线性可能无法拟合原模型

3 不同样本，如何计算邻域样本权重（只知道大的方向，远的小，近的大，但界线不好确定，参数？）

4 每个待测样本都需训练对应的可解释模型，耗时长

参考链接：

[1] <https://mp.weixin.qq.com/s/qkWGKrAuDQqmNl-vxTekXg>

[2] <https://blog.csdn.net/weixin_42347070/article/details/106455763>

[3] <https://blog.csdn.net/weixin_42347070/article/details/106076360>

[4] <https://arxiv.org/abs/1602.04938>

[5] <https://www.bilibili.com/video/BV12V4y1G7KF/?spm_id_from=333.788&vd_source=a9f595e4b26bcb66fef9c3acae4fc4fd>

**3 基于注意力机制（ATT）**

参考链接：

[1] <https://blog.csdn.net/Western_europe/article/details/109611695>

[2] <https://arxiv.org/abs/2005.00928>

**三 开源工具**

**1 captum**

[1] <https://github.com/pytorch/captum>

**2 TrustAI**

[1] <https://github.com/PaddlePaddle/TrustAI>

[2] <https://aistudio.baidu.com/aistudio/projectdetail/4622139?contributionType=1>

[3] <https://developer.baidu.com/article/detail.html?id=295219>

现状：现在使用的模型越来越大，数据形式各种各样，业务涉及越来越深

优势：对业务输入，badcase分析，关键词挖掘，数据增强提供理论依据