基于最大化F1值学习的不平衡数据

集分类算法研究

摘要：

随着移动互联网技术的兴起，越来越多的原始数据被收集起来用于分析与挖掘，其中很多领域内的数据是严重不平衡的，既属于不同类别的样本数量相差极为悬殊。传统的机器学习方法通常采用全局分类准确率作为训练目标，在不平衡数据集上表现欠佳，所以不平衡数据分类算法也逐渐成为一个受人关注的课题。

目前主要的不平衡数据集分类方法分为两种，分别是数据重采样法以及集成学习方法，他们的主要思路是通过改变原始数据集中样本的权重或者比例来使原始数据达到一种“平衡”状态，从而适用于传统的分类算法，这类算法的结果与数据分布情况相关密切，往往需要一个复杂的调节过程才能得到一个较好的结果而且泛化能力不强。所以本文提出一种直接以F1值为训练目标的算法来解决不平衡数据集分类问题，并取得了不错的效果。

引言（introduction）

随着大数据时代的到来以及各种网络化的系统变得规模越来越大，结构越来越复杂，各种各样的监视、安全、金融等系统每天都获取海量的原始数据，所以原始数据的分析与潜在信息的挖掘对于分类与决策过程起着至关重要的作用。尽管现有的方法在数据挖掘领域已经获得了巨大的成功，但是在许多的实际问题中还存在着各种各样的问题，不平衡数据便是其中一个相对比较新而且比较艰巨的挑战。

不平衡数据分类问题是指不同类别样本数目相差悬殊情况下的分类学习问题。以二分类为例，若其中有一类（正类、多数类）的样本数量比另一类（负类、少数类）的样本数量多很多，那么就称这样的分类问题为不平衡数据集分类问题。

不平衡样本分类问题从被研究学者们关注至今约有十余年，期间有很多的相关研究，在这里大体可以将所有常见的解决方法其分为两大类：

**（1）经典不平衡分类处理方法** 这一类算法的主要思想是将不平衡数据集经过一系列处理或者分类过程中采取区别对待数据集中的实例，从而将原始数据转变为一个相对的“平衡态”，进而解决不平衡问题；

**（2）传统分类器优化法** 对于不平衡数据集分类问题，这一类算法的比例远少于经典不平衡处理方法，并且他们没有一个普遍的处理规律。他们通常的思路是将传统分类算法的训练结果进行进一步的针对不平衡数据的调整，从而使得到的模型可以处理不平衡数据集。

### 经典不平衡处理方法【与b并列，同缩进】

根据现有的研究成果，经典的解决不平衡数据集的主要方法可以总结为以下几种：

**（1）原始训练数据重构** 通过重采样技术改变多数类或少数类样本的数目，从而使原始数据变得相对平衡；

**（2）集成学习方法** 训练多个弱分类器，最后通过投票或权值相加的方法得出分类结果，通常在训练过程中通过权值改变或原始数据集分割来解决不平衡问题；

**（3）敏感代价学习方法** 改变原始数据在评判标准中的权值，通常使用人为设定的敏感代价矩阵来协助计算分类损失，以解决不平衡问题。

### 传统分类器优化法

传统分类器优化算法有一个共同的特点，就是针对不平衡样本分类问题，他们不再是像经典不平衡处理方法那样针对不平衡的数据集进行一些平衡化的调整，他们的主要思想是通过修改分类器的训练过程或者分类过程来适应不平衡的数据集，既通过优化算法的训练过程来减轻不平衡分布对训练过程的影响，或者采用正常的训练思路训练模型后，通过一系列其他过程进行对模型的调整，又或是得到普通的模型之后在分类阶段采用与经典分类阶段不同的方法来解决不平衡问题。

【以下与a，b不并列】

对于现有的不平衡数据集分类方法，他们大多是将原始数据集采用某种策略转化为平衡的状态，进而采用经典的机器学习方法解决问题，而经典机器学习算法的主要思想是独立同分布假设以及归纳偏置的总结，如果改变了原始数据集的分布，通常会使经典机器学习方法无法对原始数据集进行合理的拟合，从而导致结果不稳定或结果较差等问题。而本文中的算法基于最小化损失学习，直接以不平衡数据集分类评价标准F1值作为训练目标，从而解决了上述问题。

最小化损失学习：（参见文章Joachims T. A Support Vector Method for Multivariate Performance Measures[C]//Proceedings of the 22nd international conference on Machine learning. ACM, 2005: 377-384）

对于现有的大多数的机器学习算法，我们通常认为训练数据集与测试数据集满足独立同分布，这个分布也被认为是真实的样本空间分布，也就是我们通过训练数据与测试数据总结出归纳偏置来对真实空间中的未知样本进行分类。而机器学习算法的训练目标便是在所有的假设空间中找到一个特殊的假设来使大小为的测试数据集达到设定的损失最小值，既公式2-22达到最小值。对于普通传统的损失函数，由于其通常认为空间中各点是相互独立的，所以可以将式损失函数转化为2-23的形式，也就是把整体损失转化为了所有样本单独损失的加和。

所以根据式2-22与2-23，并且根据训练数据与真实数据间的独立同分布假设，或者说训练数据是真实空间的采样假设，最终可以将2-22式转化为2-24，进而转化为3-10的形式，这也就是为什么通过训练数据集可以对真实空间进行训练或者拟合。









对于大部分传统机器学习方法，他们的训练思想通常是构建单个样本的输出与目标输出间的误差函数，并将所有样本的误差相加来作为训练集的总体损失。最理想的损失函数为0-1损失函数，既如果输出与目标输出属于同类别那么损失为0，不同则为1，如式3-11所示。



而0-1损失的平均值不一定适用于所有的问题，所以最小化损失学习这一思想被提出，通过采用自定义的损失函数代替0-1损失进行训练，以达到适应不同问题的目的，这一思想最早应用于结构化支持向量机。

其具体思想是将单一的样本与单一的输出间如式3-12的这种假设形式转化为式3-13的形式，在真实训练问题中我们使用训练样本的特征集合和目标输出集合来求解问题，所以式3-13中的训练空间为，目标空间为，假设*h*变为了整体假设，对应了所有样本输入以及所有的分类器输出。而整体损失由每个样本损失加和的式3-10的形式转化为了式3-14的形式，求假设3-13便等价为求解式3-15的结果。至此便将传统的机器学习分类算法利用独立性求解损失的思路转换为最小化全局损失学习的思路，以此来通过不平衡数据集分类评价标准建立损失，适应不平衡样本分类问题。









基于最大化F1值的不平衡数据集分类算法：

1、不平衡学习与最小化损失学习（imbalance learning and minimize lose learning）

假设现有一个一维不平衡数据集，包含多数类与少数类两种样本，其概率密度曲线如下图3-1所示，并假设多数类与少数类的样本比例为，其中。显然以传统分类器的基本思想既以最大化全局准确率为最终训练目标，对于两类的边界部分，即使其概率密度大小相似，但是因为多数类与少数类样本基数不同，所以边界部分的多数类样本个数会远多于少数类的样本个数，最终分类界线极有可能在图中线b的位置附近，处于偏向少数类的一侧。

而经典的不平衡数据集分类算法的思想是直接采用某种方法强行的缩小多数类与少数类之间的样本比值，通常会使两类的样本点数目达到相同或十分接近，然后应用传统分类算法时，如果原始数据集的概率密度曲线还是如图3-1所示，由于已经不存在两类样本基数不同这一问题，所以使全局分类准确率最高的分类界限应该是图中的线a。这条线是以两类概率密度曲线交点的横坐标作为分界阈值，分界线左边的少数类与右边的多数类为错分样本，由面积法易证，此时分类错误样本数最小。

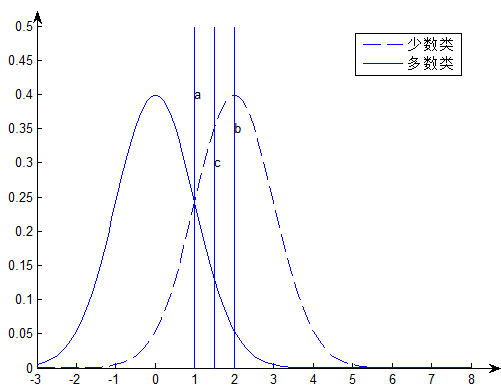


图3-1 数据集概率密度曲线示意图

然而由于样本空间的改变，我们只能认为该解（线a）是当前改变后的样本空间中最佳分类线，并不能确定该分界点是否也是原数据集上的最佳点，对于不平衡样本分类问题，通常有专用的评价标准来评价最终分类效果的好坏，而经典不平衡样本分类方法通常只能使原始空间的不平衡分类评价指标有所上升，但通常无法证明该解一定是该原始样本空间中的最优解。

所以本算法基于最小化损失学习，采用了常用的不平衡数据集评价指标F1值构建了一个特殊的损失函数，应用于神经网络模型，以解决不平衡数据集分类的问题。

2、损失函数的构建（适当精简）

本算法中选取了F1这一最经典的复合评价标准作为优化目标，所以可以将损失函数设置为(1-F1)值，对于神经网络的训练，损失值最小和F1值最大其实是一样的，只要将梯度前的符号进行改变即可，所以这里将最小化损失的概念扩充到了最大化目标函数，既将3-15式转化为3-16式的形式。



对于本节中的神经网络模型，我们依然使用如式3-17中所示传统神经网络的sgn函数作为最终分类标准，对于整个训练集上上的输出我们用来表示，而目标输出依然用来表示。为了用与表示最终的F1值，我们首先需要求出召回率（Recall）与精确率（precision），根据观察混淆矩阵与、之间的关系，可以看出混淆矩阵中的参数*TP*满足公式3-18，并且可以将公式2-3转化为公式3-19的形式，而式2-6便可以变为式3-20的形式。所以最终的F1值可以表示为式3-21的形式。











然而观察公式3-21我们可以发现因为与都是0与1组成的序列，虽然可以通过与表示出整个训练集的训练目标，但是由于求解的过程中包含了一步的操作，所以该F1值仍然是离散的，并无法与我们的神经网络输出以及各个节点间的连接层建立直接的数值联系。在结构化支持向量机中求解任意目标函数时采用的是空间遍历以及双重优化的思路，也因此需要耗费巨量的时间，而在本文的算法中，我们放弃了这一思路，而是采用神经网络输出层sigmod函数特有的式3-5的性质来将与建立起联系。

本算法利用神经网络训练过程是采取先将使用当前状态的网络进行分类，然后求解损失并对此损失进行优化已达到下一更优状态这一思想，将训练中的评价过程进行变换，不再使用当前状态下神经网络的具体分类结果，而是采用当前输出的形如式3-5的概率求解损失的期望值，并对期望值进行优化，这样既可以建立输出和参数与目标间的直接联系，也可以通过优化期望值来增加目标获取更高值的概率，这样不失训练的意义。

然而建立期望联系我们并不能采用2.2.3节中的精确求解过程，首先是因为精确求解需要耗费的时间，这有悖于本节设计快速算法的初衷，其次可以看到2.2.3节中介绍的算法最后求解出的状态—概率空间已经无法存储每一样本对其的贡献，也就是无法直接建立期望与样本之间的关系，所以也无法直接建立期望与神经网络权值之间的关系，为了解决这两个主要问题我们采用下式3-22中的近似关系。



由于有期望与协方差的关系3-23，所以对于式3-22中的分子分母分别取期望，分子的期望便如式中右侧的分子的形式，而分母的期望由于后续反向传播算法中希望可以对有所样本都可以带入，而不是对于一部分样本会舍去分类器输出，所以继续采用的平方的形式而不是直接期望的形式，但是即使取平方由于最终将会接近于0或1，所以其整体值是接近的。而对于式3-23中的后一项两个变量的协方差，我们可以发现*X*与*Y*均为，他们分别在分子与分母，所以*X*与*Y*的变化趋势相反，根绝协方差的性质，可以认为协方差将一直0，所以有关系3-24，当算法不断优化右方的近似值的同时，由于期望值为近似值的上界并且相对接近，所以也可以不断使期望扩大，收敛于一个全局最优解或者局部极优解，已达到训练目的。





综上，我们便将原始的神经网络训练目标3-12完全转变为了式3-13的形式，并根据不平衡样本分类评价标准中常用的F1值进行了目标函数的设计，然后将离散函数近似为一个连续函数，使该函数可以完美的与每个训练集样本输出建立联系，进而与神经网络参数建立联系，并且证明了神经网络的训练过程中可以在优化该近似函数的同时优化整个训练集上的最终F1值，满足最初的设计思想。

3、训练过程（建议采用给出结论并在附录中证明以节约篇幅）

对于神经网络模型，其最常用的有效的权值更新策略便是反向传播算法，由于本算法的最终结果将目标函数训练到最大值所以，其更新过程如式3-24与3-25所示。其中代表了学习率，其大小影响了神经网络的收敛速度以及收敛精度，偶尔也可能会影响到最终收敛于哪一个极优解附近。其设置方案通常有人为设定法也有退火法既设定一个相对较大的初值并随着训练过程不断的缩小。





所以为了求解出每一个权值的更新量，我们需要求解出整个目标函数F1对于每一个节点中每一个参数的偏微分，其求法如下式3-26。其中wij代表节点j的第i个权值，netj代表节点j的内积结果既通过sigmod函数（式3-2）前的结果，而xji代表节点j对应第i个权值的输入，这一项在训练过程中永远是已知第，只要通过当前状态的节点权值和输入便可以求出。接下来要求的便是式3-26结果中的前半部分既最终输出对于每个节点输出的偏微分。



为了求解各个节点的偏微分，我们需要对所有节点分为两类，一类为输出节点，另一类为隐藏层节点，由于前文中介绍过的最终目标函数是直接与输出建立了关系，所以输出节点可以直接通过当前状态下的目标函数值求解出偏微分，而对于隐藏层节点，则需要通过该节点的下游节点来求解偏微分，这也是方向传播算法链式法则的主要思想。

对于输出节点，我们采用链式法则，用函数将所求分解为式3-27的形式。其中oj为netj经过函数处理后的结果，既式3-22中的h(x)的一个分量，所以式3-27中的最后一项可以通过3-28求解，而前一项可以直接求导，其求解过程如式3-29，为了简洁该式中省略了上一节目标函数前面的常数系数，这对优化过程没有影响。







对于隐藏节点更新量，这里并没有办法直接使用目标函数求解，所以要根据每个隐藏节点的下游节点来更新其连接参数，其更新方案如式3-30所示。由于本算法为二分类问题，所以输出节点也就是只有一个。并且将求解输出节点的方法以及函数的导数带入等式3-30中之后，便可以得到式3-31的形式，其中为输出节点的偏微分，为节点的第个分量，也就是输出节点与第个隐藏节点之间的连接权值。





以上便是针对不平衡数据集的最小化损失学习算法的原始思想推演过程以及最终的训练时更新权值的方案，其算法过程如下算法3-2所示。

算法3-2 最小化损失神经网络

|  |
| --- |
| 输入：学习率，最大迭代次数，样本集合，隐藏节点数目，目标F1值  输出：输入—隐藏层连接系数矩阵，隐藏—输出层连接系数向量 |
| 1. 对与进行初始化，每一个分量的范围为(-0.1,0.1) 2. ,, 3. For i = 1 To 4. 对样本集合求解，并求解当前F1值 5. If(>) 6. Return 当前, 7. Else if (>) 8. , 9. End if 10. 根据式3-28与3-29更新，根据式3-30与3-31更新 11. End For 12. Return , |

实验结果与分析：

本章中的实验数据集均来自于UCI机器学习数据集，对于数据集的选取过程，主要是选取那些在其他不平衡数据集分类算法研究中出现过的数据集，共以下8个数据集，其参数如下表4-1所示。

表4-1 数据集参数表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 数据集 | 样本个数 | 少数类比例 | 属性个数 |
| 1 | YEAST | 1484 | 12.60% | 8 |
| 2 | Abalone | 4177 | 8.02% | 8 |
| 3 | Glass | 214 | 23.83% | 10 |
| 4 | Breast Canser | 699 | 34.50% | 9 |
| 5 | Vehicle Silhouettes | 946 | 23.43% | 18 |
| 6 | Haberman | 305 | 26.47% | 3 |
| 7 | Ecoli | 335 | 2.69% | 7 |
| 8 | Credit | 30000 | 22.12% | 24 |

本节采用了SMOTE算法，adaboost算法，结构化支持向量机算法，经典神经网络算法，敏感代价学习算法以及本文的算法进行了对比，其结果如下表所示。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | SMOTE | Adaboost | SSVM | ANN | SCL | ML-ANN |
| 1 | 0.747 | 0.717 | 0.645 | 0.667 | 0.717 | 0.846 |
| 2 | 0.406 | 0.399 | 0.486 | 0.352 | 0.331 | 0.581 |
| 3 | 0.980 | 0.922 | 0.980 | 0.837 | 0.866 | 1.000 |
| 4 | 0.984 | 0.943 | 0.961 | 0.984 | 0.956 | 0.994 |
| 5 | 0.995 | 0.997 | 0.962 | 0.945 | 0.979 | 0.995 |
| 6 | 0.459 | 0.423 | 0.632 | 0.413 | 0.660 | 0.647 |
| 7 | 0.762 | 0.696 | 0.727 | 0.941 | 0.941 | 0.941 |
| 8 | 0.534 | 0.435 | -- | 0.503 | 0.516 | 0.542 |

由上表可见，本文中的算法在不平衡数据集分类算法上取得了一定的成功，其结果通常优于以往的算法。

结论：

本文算法主要从不平衡数据集分类评价标准入手，并不再以传统分类方法的全局准确率为训练目标，改为以训练数据分类F1值为训练目标，直接从不平衡数据集分类效果差的根本入手，切中肯綮。本文的算法直接从分类器的损失函数入手，不再使用传统损失函数，而是直接构建了与F1值相关的损失函数，通过当前分类器输出求解F1值的近似期望来作为损失，通过证明该近似期望为F1值期望的下界，来确认对其优化的可行性，并且由于其不再具有原F1函数的离散性，而是直接与分类器输出建立了联系，所以可以通过反向传播算法进行迭代，以此完整的解决不平衡数据集训练问题。

对于本文中算法，还有以下问题可以进行研究或优化：

（1）从严格意义上来讲，本文的算法没有将F1值的精确期望与神经网络模型的输出或者模型参数建立直接的数学联系，如果可以攻克这一难点，可能将会使算法分类准确度有进一步的提升。

（2）本文中的算法无法并行运算，并且无法使用传统人工神经网络学习方法中的batch等加速求解过程，算法运行时间远高于经典神经网络的。所以如果可以解决此问题，应该可以在算法时间性能上有质的飞跃。