第3章

噪声数据处理

噪声是影响机器学习算法有效性的重要因素之一，由于实际数据集存在采集误差、主观标注以及被恶意投毒等许多因素，使得所构造的数据集中难免存在噪声。本章分析噪声产生的原因，对噪声的类型进行归类，介绍了噪声处理的理论基础，着重介绍了标签噪声识别的三类算法。

3.1 噪声的分类、产生原因与影响

在机器学习训练数据中，存在两种噪声。

第一种噪声是属性噪声。可能由于设备、数据处理方法等因素影响，对于一个数据样本，在每个属性或特征的测量、处理的过程中会引入噪声。例如，血压、温度的测量由于突发因素的影响造成波动。

第二种噪声是标签噪声。带标签的数据是监督机器学习算法所必需的，由于人工标注时存在一定主观性、非专业性、经验不足等，导致数据集中的样本标签可能存在一定噪声。

属性噪声只发生在样本的属性值中，标签噪声仅出现在数据的标签中。在噪声数据处理中比较少考虑同时具有属性噪声和标签噪声的情况。在机器学习中，标签噪声对学习性能的影响比属性噪声大，因此也得到更多的关注。

深入分析噪声产生的原因，有利于寻找更合理的数据噪声处理方法。根据现有的研究，主要有以下4方面。

(1) 特定类别的影响，在给定的标注任务中，各类别样本之间的区分度不同，有的类别与其他类别都比较相似，就会导致这类样本标注错误率高。

(2) 标注人为的因素，包括知识领域背景、任务分配等，有的标注任务比较专业化，需

要更高的专业水平。如果分配给标注人的标注任务过多，容易导致在后期产生标注疲劳，从而引发更多的错误。

(3) 少数类的标注更容易错误，这是由于少数类样本往往比较少见，标注者不容易进行判断，因此会使得标签噪声和非平衡问题混杂在一起。

(4) 训练数据受到了恶意投毒，当在对抗环境下应用机器学习模型时，攻击者往往会通过一些途径向数据中注入恶意样本，扰乱分类器的性能。第 9 章将会详细介绍这种数据攻击方式。

一般认为，数据质量决定了分类效果的上限，而分类器算法只能决定多大程度上逼近这个上限。因此，标签噪声对于机器学习任务有很大的影响。具体可以从以下两方面来解释。

(1) 最直接的影响就是分类器准确性下降，噪声标签扰乱了标签和属性之间的关系，而基于这种关系的分类器显然就难以提升分类性能。

(2) 训练特征空间和模型复杂性增加。由于标签和属性之间的关系变得不确定，对分类有用的特征属性提取也就变得不可靠，从而导致特征空间和模型复杂度的增加。

当然，噪声对不同分类器的影响也是不同的，KNN、决策树和支持向量机等受标签噪声的影响较大。对于集成学习来说，Boosting 方法更容易受到标签噪声的负面影响，特别是 AdaBoost 算法，在迭代后期，算法会更多地关注错分类的样本，从而导致噪声样本的权值越来越大，降低了算法性能。而对于 Bagging 来说，训练数据集中随机加入少量的噪声样本，反而有利于增加 Bagging 中基分类器的差异性，最终可能提高整个学习模型的分类性能。

3.2 噪声处理的理论与方法

有时，训练数据中存在噪声是难以避免的，这样就提出了在噪声情况下机器学习的有效性问题。

L. G. Valian 较早开始了恶意错误样本学习的研究，允许学习算法的样本中存在一定的错误样本，研究中假设每个错误是以固定的概率独立发生在每个样本上。噪声样本学习问题可以参考 L. G. Valian 提出的概率近似正确(Probably Approximately Correct，PAC)理论[1]。

“近似”是指在取值上，只要计算结果和真实值的偏差小于一个足够小的值就认为“近似正确”，因此 PAC 理论不要求学习器输出零错误率，只要求错误率被限制在某常数 ε 范围内，ε 可为任意小。

根据 PAC 理论，有一个已经被证明的结论：对于任意的学习算法而言，假设训练数据的噪声率为 β，分类器的错误率为 ε，那么两者之间存在如下关系：

$$\beta \leqslant \frac{\varepsilon}{1+\varepsilon} \tag{3-1}$$

图 3-1 给出了噪声率和错误率(下界)之间的变化关系，从图中可以看出，机器学习系统允许训练数据存在一定噪声。但是当噪声率超过 50%时，分类器已经 100%错误了。基于这个出发点，在噪声数据处理中，一般假设噪声比例并不太高。

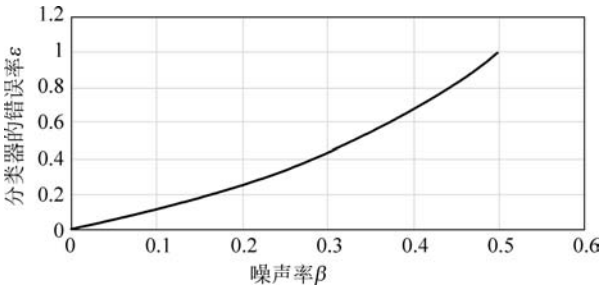

图 3-1　噪声率和错误率(下界)之间的变化关系

要提高学习系统的准确率,其中的途径之一便是减少训练数据的噪声水平。从技术方法的角度看,噪声学习的处理方法可以分为数据层面、算法层面和模型层面。在数据层面,使用各种方式识别噪声,经过清洗之后再训练模型;在算法层面,运用合适的算法进行噪声过滤;在模型层面,主要是构造并训练对噪声鲁棒的模型。

去除训练集中的噪声不但有助于改善分类器训练性能,而且对于机器学习模型在对抗环境下的防御策略也有显著作用。分类器的恶意攻击者可以在训练数据中添加带毒样本,可能表现为一种噪声,而进一步结合攻击者的目的、手段,可以更加有效地进行噪声过滤。因此,噪声过滤也是一种重要的机器学习攻击的防御方法,更多的防御方法将在第12章中介绍。

实际问题中,可以根据噪声程度、噪声类型、噪声产生原因等因素来选择噪声处理方法。由于在模型构建时,无法判断所使用的数据是否包含标签噪声,所以直接对标签噪声进行建模或在模型中考虑标签噪声的做法,并不能使模型的性能得到保障。因此,对噪声进行清洗再训练的方法比直接构建噪声鲁棒模型更常用。

3.3　基于数据清洗的噪声过滤

在这类方法中,一般假设噪声标签样本是分类错误的样本,因此就把噪声样本的过滤问题转换为普通的分类问题。这种方法的基本思路是消除或纠正训练数据中的错误标签,这个步骤可以在训练之前完成,也可以与模型训练同步进行。噪声去除方法具体包括直接删除法、基于最近邻的去噪方法和集成去噪法等。

1. 直接删除法

直接删除是一种最简单的噪声清洗方法,根据若干规则,直接在数据集中进行样本匹配,如果符合规则则认为是噪声并删除样本。

用于噪声清洗的规则基于两种情况:把看起来比较可疑的实例删除或者把分类错误的训练实例删除。在具体实现方法上,判断样本的可疑程度,可以使用边界点发现之类的数据挖掘方法。

该方法的主要问题在于容易造成数据样本数量减少,特别是对于训练样本本来就少的应用或是非平衡分类问题,简单地基于规则进行样本删除并不可取。

2. 基于最近邻的去噪方法

在第 2 章中提到了可以使用 KNN 及其改进方法来改善非平衡数据分布，其思路是利用样本分布中可能存在的异常来做相应的调整。噪声样本一般也会表现出异常的特征值，因此可以用最近邻方法来进行噪声过滤。

从 KNN 本身原理来看，k 越小，所包含的近邻数量越少，一旦有噪声样本，那么会导致附近的样本分类错误。如图 3-2 所示，黑圆和白圆表示两类已知标签的样本，a 是噪声样本。显然当 k 较小，如 $k=1$ 时，b 和 c 这两个样本都会被分为白圆样本，因为它们离噪声点 a 最近。当 k 大时，噪声点对 b、c 判断结果的影响就减小了。因此，KNN 当 k 较小时，噪声会导致其近邻样本分类错误。可以利用这种噪声敏感性进行噪声过滤，当发现若干样本分类错误都是由同一个邻居而引起时，那么这个邻居样本就可能是噪声。

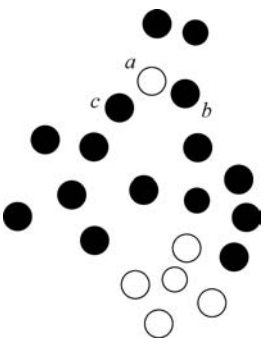

图 3-2　KNN(k 较小时)噪声的影响

KNN(k 较小时)是一种典型的噪声敏感模型，除此之外，在非平衡分类中提到了若干基于 KNN 的方法，包括浓缩最近邻 CNN、缩减最近邻 RNN、基于实例选择的 ENN 等，也都可以用于噪声过滤。除了 KNN 外，噪声敏感的分类器还有 SVM、AdaBoost，选择噪声敏感的模型有利于从训练数据中识别过滤噪声。

3. 集成去噪法

集成分类方法对若干弱分类器进行组合，根据结果的一致性来判断是否为噪声，也是一种较好的标签去噪方法。

对于给定的标签数据集，如何排除其中的噪声样本？主要的问题是集成分类器的选择和训练。根据所使用的训练数据，可以分为以下两类处理方法：①使用具有相同分布的其他数据集，当然该数据集必须是一个干净、没有噪声的数据；②不使用外部数据集，而是直接使用给定的标签数据集进行 K 折交叉分析。

图 3-3　集成去噪

如图 3-3 所示，训练样本集即是标签数据集 T，包含标签噪声，但不需要人工标注是否有噪声，而是希望通过集成学习来过滤这些噪声样本。基于 K 折交叉分析的集成投票法首先对数据集进行划分，并分别划分出 k 组训练集和测试集，如图 3-3 是集成去噪的流程图。对于每一组划分得到数据集，分别训练基分类器，然后对于相应的测试集进行噪声检测。其中，检测的方法是基于每个基分类器的判定结果的一致性或多数情况，其基本出发点仍是错分即为噪声。对于一致投票而言，当所有的学习者都同意删除某个实例时，它就会被删除。对于多数投票，被超过半数的分类器错分的样本则可以视为标签噪声样本。

这种方法在标签数据集上就可以进行，但要求基分类器有一定的噪声鲁棒性，即少量的噪声不影响分类性能。从这个角度看，基于噪声数据清洗和鲁棒性模型的噪声处理方法在研究和应用上也不是完全独立的。在噪声清洗中使用噪声鲁棒性模型作为基分类器也许是一种不错的方法。

噪声鲁棒的分类器有神经网络、贝叶斯等。噪声鲁棒与否，除了与分类器类型有关，还与损失函数有关。对于均匀分布的标签噪声，0-1 损失和最小平方损失是抗噪声标签的，而指数损失、对数损失和合页损失等则不具备抗噪能力。

3.4　主动式噪声迭代过滤

基于数据清洗的噪声过滤方法的隐含假设是噪声为错分样本，把噪声和错分样本等同起来。这对于离群点噪声是合适的，但是数据中通常有些噪声和错分样本并没有明显差异，特别是位于分类边界的样本。在这种情况下，引入人类专家的交互来协助机器处理这类样本就显得非常有必要了[2,3]。

主动学习框架和理论为人类专家与机器学习的协作提供了一种有效的途径，它通过迭代抽样的方式将某种特定的样本挑选出来，交由专家对标签进行人工判断和标注，从而构造有效训练集。

如图 3-4 所示是一个主动学习框架，包含两部分：训练和查询。在训练环节，利用学习算法对已标注的样本进行学习获得分类器模型。由于通过人工标注获得了一定量的确定性标注样本，利用这些样本来提升未知噪声的检测效率是标注的目的，因此，主动学习的噪声过滤中的模型训练环节可以采用监督学习、半监督学习等方法。

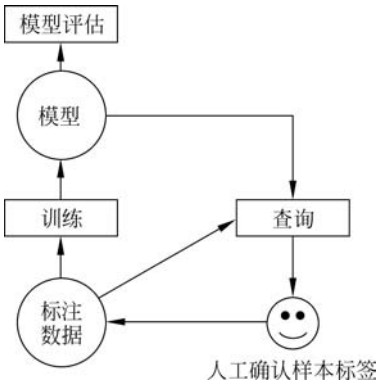

图 3-4　噪声过滤的主动学习框架

在查询环节，基于模型通过运用样例选择算法从训练数据中选择一些需要人工确认的样本，由领域的专家进行标注确认。同时，将确认后的样本加入分类器的训练数据集中。如此重复迭代训练和查询，直到模型的泛化能力不再增强为止，这个终止条件的判断是在模型评估的基础上完成的。最终可以得到一个泛化能力比较强的分类器，同时完成噪声的处理。

在这个框架中，关键问题是如何选择需要人工确认的样本。一方面要尽可能过滤噪声；另一方面也要考虑人工标注的工作量。按照目前文献，查询策略主要有两类，即基于池的样本选择算法和基于流的样本选择算法。这两类策略分别对应流式的主动学习(sequential active learning)和基于池的主动学习(pool-based active learning)。

基于池的样本选择算法通过选择当前基准分类器最不能确定其分类类别的样例进行人工标注。基于流的样本选择算法按照数据流的处理方式，在某个时间窗内可以参照基于池的样本选择算法。具有代表性的基于池的样本选择算法有基于不确定性采样的查询方法、基于委员会的查询方法、基于密度权重的查询方法等，下面进行介绍。

1. 基于不确定性采样的查询方法

基于不确定性采样的查询方法的策略是将分类模型难以区分的样本提取出来，在衡量不确定性时可以采用的方法有最小置信度、边缘采样和熵。

1）最小置信度

所谓最小置信度就是选择最大分类概率最小的样本，即

$$x_{\mathrm{LC}}^{*}=\arg\max_{x}(1-P_{\theta}(\hat{y}\mid x))=\arg\min_{x}P_{\theta}(\hat{y}\mid x) \tag{3-2}$$

其中

$$\hat{y}=\arg\max_{y}(P_{\theta}(y\mid x)) \tag{3-3}$$

θ 是模型参数；x 表示样本；y 表示类别。

例如，两个样本 a、b 的类别概率分别为(0.71,0.19,0.10)、(0.17,0.53,0.30)，那么根据最小置信度准则，应当选择样本 b，因为从最大概率来看，b 的不确定性大于 a。

2）边缘采样

边缘采样是选择那些类别概率相差不大的样本。

$$x_{\mathrm{M}}^{*}=\arg\min_{x}(P_{\theta}(\hat{y}_1\mid x)-P_{\theta}(\hat{y}_2\mid x)) \tag{3-4}$$

其中，$\hat{y}_1$、$\hat{y}_2$ 是样本 x 归属概率最大的两个类别。

对于上述 a、b 两个样本，应当选择 b，因为 0.53−0.30<0.71−0.19。对于二分类问题，边缘采样和最小置信度是等价的。

3）熵

熵衡量了在每个类别归属概率上的不确定。选择熵最大的样本作为需要人工判定的样本。

$$x_{\mathrm{H}}^{*}=\arg\max_{x}\left(-\sum_{i}P_{\theta}(y_i\mid x)\cdot\ln P_{\theta}(y_i\mid x)\right) \tag{3-5}$$

对于上述 a、b 两个样本，它们的熵分别是 0.789、0.999，可见 b 的不确定大于 a，因此也应当选择 b。

2. 基于委员会的查询方法

当主动学习中采用集成学习作为分类模型时，这种选择策略考虑到每个基分类器的投票情况。如图 3-5 所示，m 个基分类器给 n 个样本分类，要从中选择可能是噪声的样本。其基本依据是每个基分类器对每个样本的投票或分类结果 y_{ij}。相应地，通过基于投票熵和平均 K-L 散度来选择样本。

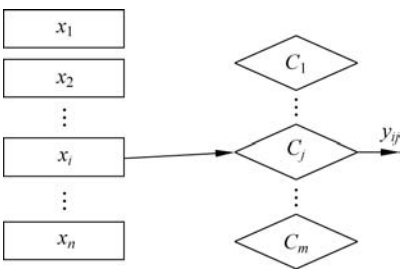

图 3-5　m 个基分类器给 n 个样本分类

对样本 x 计算投票熵时，把 x 的每个类别的投票数当作随机变量，衡量该随机变量的不确定性。

$$x_{\mathrm{VE}}^{*}=\arg\max_{x}\left(-\sum_{i}\frac{V(y_i)}{C}\cdot\ln\frac{V(y_i)}{C}\right) \tag{3-6}$$

其中，$V(y_i)$表示把 x 标注为 y_i 的分类器的个数；C 表示分类器总数。投票熵越大，就表示对分类结果越不确定，因此就越有可能被选择出来。

另外一种选择方式是基于平均 K-L 散度。

当每个基分类器为每个样本输出分类概率时，可以使用平均 K-L 散度来计算各个分类器的分类概率分布与平均分布的平均偏差。对于某样本，如果其平均偏差越大，则各分类器输出的概率分布的一致性越差，就越有可能被选择出来让人工进一步确认。

平均 K-L 散度的选择依据如下：

$$x_{\mathrm{K\text{-}L}}^{*}=\arg\max_{x}\left(\frac{1}{C}\sum_{c=1}^{C}D(P_{\theta^{(c)}}(y_i\mid x),P_C(y_i\mid x))\right) \tag{3-7}$$

其中，$D(,)$代表两个分布的 K-L 散度；$P_C(y_i|x)$是平均分布，即

$$P_C(y_i\mid x)=\frac{1}{C}\sum_{c=1}^{C}P_{\theta^{(c)}}(y_i\mid x) \tag{3-8}$$

3. 基于密度权重的查询方法

位于类别边界的噪声样本一般来说都是难以区分的，更需要由领域专家进行判断和标注，因此如何提升这部分样本被选择的可能性是设计主动学习选择策略的重要问题之一。

难以区分的类别边界样本一般都具有密度比较大的特点，因此基于密度权重的策略就是在基于不确定性采样的查询、基于委员会的查询的基础上进一步考虑样本密度的影响，优先选择所在区域密度高的样本。

用公式表示为

$$x_{\mathrm{ID}}^{*}=\arg\max_{x}\left(\phi_A(x)\cdot\left(\frac{1}{U}\sum_{u=1}^{U}\mathrm{sim}(x,x^{(u)})\right)^{\beta}\right) \tag{3-9}$$

其中，ϕ_A 表示使用不确定方法或委员会查询得到的样本判决函数；$x^{(u)}$是第 u 类的代表元，类似类中心点；U 表示类别个数；β 是一个参数；sim 是相似性计算函数。

4. 其他策略

其他经典的策略还有梯度长度期望(Expected Gradient Length，EGL)方法，根据未标注样本对当前模型的影响程度，优先筛选出对模型影响最大的样本；方差约简(Variance Reduction，VR)策略，通过减少输出方差能够降低模型的泛化误差。

3.5　噪声鲁棒模型

噪声鲁棒模型通过在分类模型中嵌入噪声处理的学习机制，使得学习到的模型能抵抗噪声。在机制设计上，可以从错误样本权重调整、损失函数设计等角度提升模型的噪声

鲁棒性。

3.5.1 错误样本权重调整

机器学习模型对训练样本进行拟合，以 SVM 为例，目标是学习一个函数 $f(\boldsymbol{x})=\boldsymbol{w}\boldsymbol{x}+b$，其中 $\boldsymbol{w}$ 可以看作适合所有训练样本的权重向量。当训练集中包含噪声样本时，必然会影响 $\boldsymbol{w}$ 的优化。如果在优化过程中，自动调整噪声样本的权重，那么就可以构建噪声鲁棒模型。然而这种优化方法，在不同模型中的设计方法也有所差别。这里以 AdaBoost 为例来介绍这种分析方法、设计思路和实现方法。

AdaBoost 是 Boosting 的自适应算法，它集成了若干基分类器，采用级联的方式进行模型训练。某个基分类器训练完成后，根据其测试结果提升错误分类的样本的权重，从而使得下一个基分类器在训练时更关注这些被错误分类的样本。

为了对它进行噪声鲁棒性改造，首先介绍该算法的具体过程，并分析该算法的噪声鲁棒性。下面给出 AdaBoost 算法的形式化描述，总体上看包含了初始化、迭代和集成三个主要步骤。

算法：AdaBoost

输入：训练数据集 $T=\{(x_i,y_i),i=1,2,\cdots,N\}$，基分类器的个数 M

输出：组合分类器 $G(x)$

1. 初始化

设置训练样本的初始权重为均等值，$w_{1i}=\dfrac{1}{N}$，并记相应的训练样本权重为

$$D_1=(w_{11},w_{12},\cdots,w_{1N})$$

2. 迭代

for $m=1$ to M　　# 进行 M 轮迭代

(1) 使用 T 和 D_m 训练得到一个基分类器，记为 $G_m(x)$。

(2) 计算分类器 G_m 的分类错误率：

$$e_m=P(G_m(x_i)\neq y_i)=\sum_{i=1}^{N}w_{mi}I(G_m(x_i)\neq y_i) \tag{3-10}$$

如果 $e_m>0.5$，结束迭代。

(3) 计算分类器 G_m 在组合分类器中的重要性，即其权重系数：

$$\beta_m=\frac{1}{2}\ln\frac{1-e_m}{e_m} \tag{3-11}$$

(4) 计算训练集中每个样本的权重：

$$w_{m+1,i}=\frac{\mathrm{e}^{-\beta_m y_i G_m(x_i)}}{Z_m}w_{mi} \tag{3-12}$$

其中，Z_m 是归一化因子，表示如下：

$$Z_m=\sum_{i=1}^{N}w_{mi}\mathrm{e}^{-\beta_m y_i G_m(x_i)} \tag{3-13}$$

更新每个样本的权重，并记为 $D_{m+1}=(w_{m+1,1},w_{m+1,2},\cdots,w_{m+1,N})$。

3. 集成

构建组合分类器，表示如下：

$$f(x)=\sum_{m=1}^{M}\beta_m G_m(x) \tag{3-14}$$

最终的分类器记为

$$G(x)=\text{sign}(f(x))$$

AdaBoost 在学习基分类器时，按照指数损失调整分类器的权重系数：

$$L=\sum_{i=1}^{N}\mathrm{e}^{-y_i(f_{m-1}(x_i)+\beta_m G_m(x_i))}$$

其中，$f_{m-1}(x)=\sum_{i=1}^{m-1}\beta_i G_i(x)$。

最小化损失后可以得到式(3-11)的分类器权重。由式(3-12)可以看到 AdaBoost 的样本权值调整方法。如果样本分类错误，$G_m(x_i)$ 与 y_i 异号，否则两者的计算结果同号。因此，当样本 x 分类错误时，其权值以 e^{β_m} 变化；而对于正确分类的样本以 $\mathrm{e}^{-\beta_m}$ 变化。并且，从上述算法流程及式(3-10)可以看出，$0\leqslant e_m\leqslant 0.5$，相应地，$\beta_m\geqslant 0$。因此，对于错误的样本，其权重 $w_{mi}>\mathrm{e}^0=1$，而分类正确的样本，其权值 $w_{mi}\leqslant\mathrm{e}^0=1$。

下面分析噪声数据对 AdaBoost 分类器的影响。

对于训练集中的噪声样本而言，其在每轮迭代过程中都很可能无法被正确分类，因此，每轮的权值会以 e^{β_m} 因子进行调整。假如每轮都被错误分类，则经过 t 轮后得到的权重为 $\mathrm{e}^{\beta_{m1}},\mathrm{e}^{\beta_{m2}},\cdots,\mathrm{e}^{\beta_{mt}}$，可见噪声样本的权重得到了快速增加而变得很大。最终，在算法流程中，当某次的基分类器错误率超过 0.5 时就停止迭代。

此外，当基分类器的错误率越高，其权重 β_m 越小，错误样本的权值就越小，正确分类的样本的权值就越大，两类样本的权值差异越小；反之，β_m 越大，两类样本的权值差异越大。β_m 对错误样本和正确分类样本的权值调整影响如图 3-6 所示，图中横坐标为 β_m。

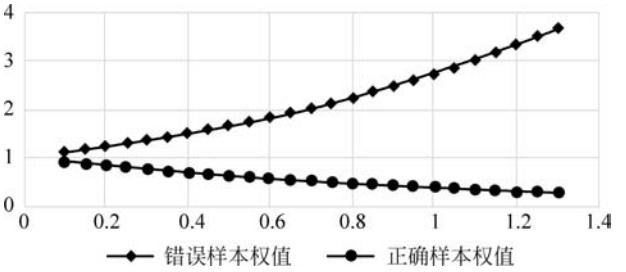

图 3-6　基分类器的权重对样本权重的影响

综上所述，根据 AdaBoost 的工作原理，可以发现 AdaBoost 串接的基分类器中，越往后面，错误标签的样本越会得到基分类器的关注。因此，当训练数据存在噪声样本时，噪

声样本容易导致串接在后面的基分类器产生过拟合。从这个角度看，AdaBoost 是一个对标签噪声敏感的分类方法。

在提升 AdaBoost 的噪声鲁棒性时，可以利用这个特点，删除权重过高的样本或调整异常样本的权重来降低标签噪声的影响。基于这个思想，Domingo 和 Watanade 于 2000 年提出了 MadaBoost 算法[4]，解决了 AdaBoost 的标签噪声敏感问题。针对噪声样本在后期的训练权重过大的问题，算法重新调整了 AdaBoost 中的权值更新公式，设置了一个权重的最大上限 1，限制标签噪声造成的样本权值的过度增加。

3.5.2 损失函数设计

在迭代学习中改变样本权值消除噪声数据影响的方法，虽然有一定依据，但是也容易造成样本权值的误调整。为此，可以从损失函数的设计角度进行改进，细化不同样本的权重调整方式。

损失函数也称为代价函数、误差函数，是学习理论中的重要概念，是一种衡量预测函数拟合真实值程度的一种函数。一般地，损失函数越小，模型拟合效果越好。

对于噪声鲁棒模型设计，最优的分类器应当能够对错误标签（噪声样本）进行误分，即具备纠正标签错误的能力。然而对于不完美的基分类器，噪声样本会被正确分类，从而导致其权重逐步下降而减少了被误分的机会。因此，可以从损失函数的角度对此问题进行纠正。

1. 在损失函数中处理噪声

为了设计合理的损失函数，从以下四种情况入手进行分析。

（1）噪声样本被正确分类；

（2）非噪声样本被正确分类；

（3）噪声样本被错误分类；

（4）非噪声样本被错误分类。

对于这四种情况，(1)(4)应增大损失，(2)(3)应减少损失，改进现有损失函数使之满足这些情况。全体样本根据其分类结果归属噪声或正常样本的可能性，调整其损失量化的增减方向，如图 3-7 所示。

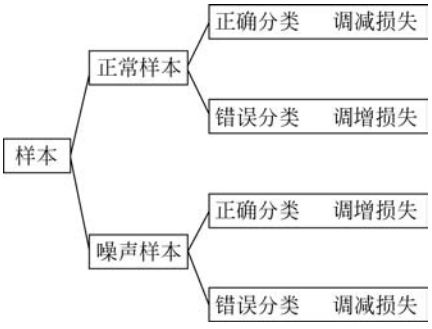

图 3-7 损失函数调整

在监督学习场景下，模型训练过程中要区分正确分类和错误分类两种情况是没有问题的。但是当没有正常与噪声的标注数据时，要进一步区分正常样本和噪声样本，就难以实现了。在这种情况下，只能依赖于先验知识、无监督或半监督信号。

1）选择噪声敏感算法为每个样本打分

噪声敏感模型能够在一定程度上推测噪声的可能性，典型的方法有 KNN（k 较小时）、SVM、AdaBoost 等，有利于提高从训练数据中识别噪声的准确性。

2）根据先验知识

把噪声置信度引入损失函数中。一般而言，分类算法所给出的信号可以用来定义噪声置信度。例如，在 KNN 中，通过样本 x 的最近 k 个邻居的标签分布来判断 x 为噪声的置信度，显然与 x 标签不同的邻居越多，噪声置信度就越大。

一些概率模型也可以用来量化噪声置信度。例如，EM 算法估算样本 z_i 属于类别 c 的概率为 $p(c|z_i)$，如果某个样本对每个类归属概率相差不大，那么其也具有一定的噪声置信度。

那么，如何把这些信息融合到分类模型的损失函数中呢？这与损失函数的形式有关。由于 AdaBoost 对噪声敏感，有研究人员提出了基于噪声检测的 AdaBoost（ND-AdaBoost）[5]。在 AdaBoost 中集成了一个基于噪声检测的损失函数，以便在每个迭代步骤中更准确地调整权重分布。

ND-AdaBoost 在 AdaBoost 损失函数的基础上，增加了如下一个反映噪声置信度的因子。

$$\phi(x)=\mathrm{sgn}(\bar{\mu}-\mu(x)) \tag{3-15}$$

其中，$\mu(x)$是样本 x 为噪声的置信度，可以用上述各种方法来衡量；$\bar{\mu}$ 是所有样本的噪声置信度的平均值；$\phi(x)$的取值范围为$[-1,1]$，其中 -1 表示 x 为噪声，1 表示 x 为正常样本。在这个计算方法下，噪声置信度大于平均值的样本被视为噪声。

把置信度因子加入损失函数中，添加的方法是把 $\phi(x)$与分类器的结果相乘。

$$L=\sum_{i=1}^{N}\mathrm{e}^{-y_i(f_{m-1}(x_i)+\beta_m G_m(x_i)\phi_m(x_i))} \tag{3-16}$$

其中，$f_{m-1}(x)=\sum_{i=1}^{m-1}\beta_i G_i(x)\phi_i(x)$。

重新整理式(3-16)，可得

$$L=\sum_{y_i G_m(x_i)\phi_m(x_i)=-1}\mathrm{e}^{-y_i f_{m-1}(x_i)}\mathrm{e}^{\beta_m}+\sum_{y_i G_m(x_i)\phi_m(x_i)=1}\mathrm{e}^{-y_i f_{m-1}(x_i)}\mathrm{e}^{-\beta_m} \tag{3-17}$$

可以看出，在第 m 次迭代时，有以下两种情况。

（1）如果 x_i 被错分，即 $y_i G_m(x_i)=-1$，那么损失函数的第一项是计算正常样本的损失，第二项计算噪声样本的损失。对于正常样本而言，e^{β_m} 因子使得损失调增。对于噪声样本而言，$\mathrm{e}^{-\beta_m}$ 得到损失调减。

（2）如果 x_i 被正确分类，即 $y_i G_m(x_i)=1$，那么损失函数的第一项是计算噪声样本的损失，第二项计算正常样本的损失。对于噪声样本而言，e^{β_m} 因子使得损失调增。对于

正常样本而言，$e^{-\beta_m}$ 得到损失调减。

2. 损失函数的其他形式

在二分类问题中，基于损失函数的处理会使得模型对标签噪声的鲁棒性更好，0-1 损失对于对称或均匀标签噪声体现出良好的鲁棒性。使用不同损失函数训练模型，所得到的噪声敏感性会有一定差异，下面进行介绍。令 y 表示目标值，$f(x)$表示预测值。

假设有 N 个样本 $D=\{(x_i,y_i),i=1,2,\cdots,N\}$，学习到的模型是 $f(\cdot)$，那么 $y=f(x)$表示模型的预测值，则该模型的损失函数为 $L(Y,f(x))$，以下是不同的损失函数。

1）0-1 损失函数

$$L(Y,f(x))=\sum_{i=1}^{N}l(y_i,f(x_i))$$

其中

$$l(y_i,f(x_i))=\begin{cases}1, & y_i\neq f(x_i)\\ 0, & y_i=f(x_i)\end{cases} \tag{3-18}$$

2）绝对值损失

$$L(Y,f(x))=\sum_{i=1}^{N}l(y_i,f(x_i))$$

其中

$$l(y_i,f(x_i))=|y_i-f(x_i)| \tag{3-19}$$

3）平均绝对误差(MAE)

$$L(Y,f(x))=\frac{1}{N}\sum_{i=1}^{N}|y_i-f(x_i)| \tag{3-20}$$

4）平方损失函数

$$L(Y,f(x))=\sum_{i=1}^{N}(y_i-f(x_i))^2 \tag{3-21}$$

5）均方误差

$$L(Y,f(x))=\frac{1}{N}\sum_{i=1}^{N}(y_i-f(x_i))^2 \tag{3-22}$$

6）均方根误差

$$L(Y,f(x))=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-f(x_i))^2} \tag{3-23}$$

7）交叉熵损失

对于二分类，可表示为

$$L(Y,f(x))=-\frac{1}{N}\sum_{i=1}^{N}y_i\ln f(x_i)+(1-y_i)\ln(1-f(x_i)) \tag{3-24}$$

对于多分类，可表示为

$$L(Y,f(x))=-\frac{1}{N}\sum_{i=1}^{N}y_i\ln f(x_i) \tag{3-25}$$

8）指数损失

$$L(Y,f(x))=\frac{1}{N}\sum_{i=1}^{N}\mathrm{e}^{-y_i\ln f(x_i)} \tag{3-26}$$

9）Hinge 损失函数

$$L(Y,f(x))=\sum_{i=1}^{N}l(y_i,f(x_i))$$

其中

$$l(y_i,f(x_i))=\max(0,1-y_if(x_i)) \tag{3-27}$$

由于损失函数是模型优化的依据，为了防止过拟合，往往都还需要在函数定义的基础上，添加正则化处理，提升模型参数的泛化能力，常用的有 L1、L2 范数。同时，正则化项之前一般会添加一个系数，由用户指定，用于平衡正则化的重要性。

参考文献

[1] Kearns M J. The computational complexity of machine learning[M]. Cambridge: MIT Press,1990.

[2] 袁龙. 基于主动学习的标签噪声处理技术研究[D]. 重庆：重庆邮电大学,2019.

[3] 孟晓超. 基于主动学习的标签噪声清洗方法研究[D]. 太原：山西大学,2020.

[4] Domingo C, Watanabe O. MadaBoost: A modification of adaBoost[C]. Proceedings 13th Conference on Computational Learning Theory. 2000: 180-189.

[5] Cao J J, Kwong S, Wan R. A noise-detection based AdaBoost algorithm for mislabeled data[J]. Pattern Recognition, 2012, 45: 4451-4465.