对综述类文章
“A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches”
的阅读总结

陈宇飞
2020100181

该报告将分为三个部分，第一部分主要介绍综述类文章的内容，第二部分主要介绍综述类文章引文中 2-3 篇文献的内容，第三部分是对第二部分引文内容的评价和比较

文章简介

本篇文章主要研究了在不平衡数据集中，利用 ensemble 集成技术的优势，从而在分类任务中获得更好的效果。不平衡数据分类问题的背景，是由于数据收集的简便性和快速性，数据集的体量越来越大，容易收集到的数据极大地占据了整个数据集的配额，于此同时小概率事件、特殊事件在数据集中的占比越来越小。在这种情况下，小概率事件和特殊事件往往无法被传统分类器识别出，然而他们在现实世界中的意义却远远大于普通样本。因此，与一般的分类任务不同，不平衡数据集的分类问题，往往并不是将分类准确率 Accuracy 作为判断标准，而是关注混淆矩阵，比如以 AM 准则(arithmetic mean of True Positive Rate and True Negative Rate)来评价一个分类器的好坏。因为在不平衡数据中，如果按照准确率 Accuracy 来评判分类器的好坏，由于特殊样本的数量偏小，他们的对分类器的贡献也偏小。举例来说，假设在二分类中有 95 个大类样本和 5 个小类样本，即便使用最简单的“将所有样本判别问大类”的分类器也能获得高达 95%的准确性。在现实生活中，例如贷款性质的判别中，大部分样本都是偿还的贷款，而坏账的比例很小，但是坏账会带来极大的经济损失，也是我们所关注的目标，因此不平衡分类问题也越来越受到人们的重视。

这篇综述文章主要收集了以集成方法为基础的不平衡数据分类算法，包括了 Bagging 方法、Boosting 方法和 Hybrid-Based 方法三种集成路径。集成 Ensemble，顾名思义是将多个简单学习器组合成为一个总体学习器的技巧。对于 Bagging 来说，在训练每一个简单学习器时，我们一般通过某种随机采样的方法，从总体样本中抽出部分样本对模型进行训练，最后使用投票法或者平均法对所有简单学习器进行整合，得到最终的模型。对于 Boosting 来说，每一个简单学习器的训练是在以往所有学习器的结果上得到的，即在每一个 iteration 的训练中，学习器都希望弥补当前整合学习器和样本标签之间的差距，使得整合的学习器能够在迭代中提升分类等能力。将集成方法和传统学习器相结合，也是目前学术研究中比较热门的话题。

不平衡分类问题的介绍

不平衡数据集产生的影响

第一，当小类别样本在总体数据中占比过小时，有些情况下我们无法学习到小类别样本的分布，或者可学习的分布或其他结构类信息的波动性是非常大的，因此想要不平衡问题有一个较好的解答，首先需要要求小类别样本都是具有结构上的代表性的，即 representative。只有在这种情况下，学习器才能够通过样本中的小类别数据给出较好的分类预测。

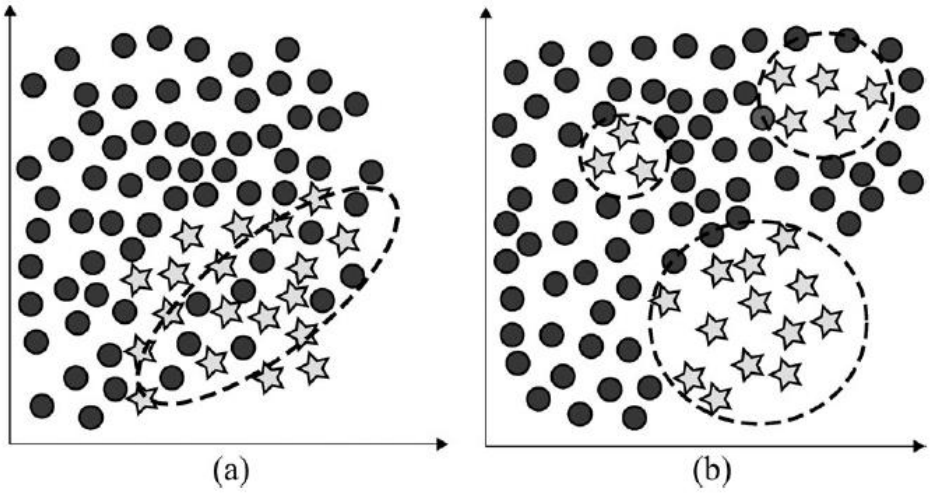

Fig. 1. Example of difficulties in imbalanced data-sets. (a) Class overlapping. (b) Small disjuncts.

第二，不平衡问题中可能有大类别样本和小类别样本的定义域出现重叠的现象，在这种情况下我们很难再去给出一个 discriminative rules。

第三，当我们的小类别样本数据过小，很有可能发生的一个问题是小类别样本由于样本过少，原本的拓扑在样本形式下被分割成多个小部分，从而使分类器无法逼近真实的分布。

不平衡分类问题的评价标准

TABLE I
CONFUSION MATRIX FOR A TWO-CLASS PROBLEM

	Positive prediction	Negative prediction
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

在上表中我们给出了二分类问题中混淆矩阵四个元素的名称，可以看出准确率有:

$$\mathrm{Acc} = \frac{TP + TN}{TP + FN + FP + TN}.$$

另外我们还可以使用 AUC 准则，如

$$\mathrm{AUC} = \frac{1 + TP_{\mathrm{rate}} - FP_{\mathrm{rate}}}{2}.$$

其中

$$TP_{\mathrm{rate}} = \frac{TP}{TP+FN}$$

并且

$$FP_{\text{rate}} = \frac{FP}{FP+TN}$$

当然我们也可以定义其他类型的 measurement 衡量标准，这些衡量标准其实都是为现实意义下判别错误的损失所决定的。

不平衡问题的处理

不平衡问题的处理有一下几个途径：

第一，通过修改现有算法的损失函数，倾向于给予小类别标签更高的误判损失。这类方法要求算法设计者对实际问题有深入的了解，知晓小类别样本与平凡样本之间重要性的比例或差距。

第二，使用数据处理方法，将不平衡数据集进行填补和删除，来获得平衡的数据集，在平衡的数据集中使用传统的分类方法进行模型训练和预测。

1. 随机降采样(Random Undersampling)：通过随机删除大类别样本，来达到数据平衡的目的。
2. 随机升采样(Random Oversampling)：通过学习小类别样本的分布结构，随机构造新的小类别样本，来达到数据平衡的目的。

文章阅读（一）

标题：AdaCost Misclassification Cost-sensitive Boosting

摘要

Adacost 是一个以 adaboost 为模板，在损失函数部分针对不同标签的误判设置了与数据有关的代价敏感损失 Cost-sensitive loss 的一种 boosting 算法。在分类问题中，将 0-1 损失替代为类别相关的误判损失用于更新 boosting 算法的相关参数。除算法外，这篇文章从理论上证明了在 cost-sensitive 的范畴下，AdaCost 算法的累计误判损失上界要比 Adaboost 要好；从实验上验证了算法在降低累计误判代价的优秀效果。

算法

算法可以被如下的流程图简要表示。其中x_i是特征向量，c_i是损失因子，y_i是标签。在每个循环中，我们训练一个新的弱学习器，更新每个变量的损失因子，最后构成最终的分类器。

- Given: $\mathcal{S} = \{(x_1, c_1, y_1), \ldots, (x_m, c_m, y_m)\}$; $x_i \in \mathcal{X}, c_i \in \mathbb{R}^+, y_i \in \{-1, +1\}$.
- Initialize $D_1(i)$ (such as $D_1(i) = c_i / \sum_j^m c_j$).
- For $t = 1, \ldots, T$:
  1. Train weak learner using distribution D_t.
  2. Compute weak hypothesis $h_t : \mathcal{X} \to \mathbb{R}$.
  3. Choose $\alpha_t \in \mathbb{R}$ and $\beta(i) \in \mathbb{R}^+$.
  4. Update

$$D_{t+1}(i) = \frac{D_t(i)\exp\Big(- \alpha_t y_i h_t(x_i) \boxed{\beta(i)} \Big)}{Z_t}$$

  where $\beta(i) = \beta\Big(\text{sign}(y_i h_t(x_i)), c_i\Big)$ is a cost-adjustment function. Z_t is a normalization factor chosen so that D_{t+1} will be a distribution.
- Output the final hypothesis:

$$H(x) = \text{sign}(f(x)) \text{ where } f(x) = \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

Figure 1: AdaCost

实验

数据集:
文章使用的数据集来源于 UCI Machine Learning Database 中的分类数据集，人工制造的分类数据集和一个现实世界中的信用卡诈骗数据集（注意到都是不平衡的数据集）。在表格中我们也给出了小类别样本所占数据集总量的比例。

损失因子的设定:
对于一般数据集的损失因子, 文章针对正样本和负样本分别将损失比率设定为 2 到 9 的值; 对于真实数据的损失因子，文章引用了前文的处理方法。在投入训练之前，损失因子被标准化到[0,1]区间上。

Table 1: Data Set Summary

$\mathcal{S}$	Data	Data Size	Testing Size	Positive%
1	hypothyroid	3163	CV	4.77
2	boolean	32768	CV	13.34
3	dis	2800	972	4.63
4	crx	690	CV	44.5
5	breast cancer	699	CV	34.5
6	wpbc	198	CV	23.74
7	chase	40K*10	40K*10	≈ 20

实验结果:

Table 2: Percentage Cumulative Loss by cRIPPER, AdaBoost and AdaCost for Six Data Sets

$\mathcal{S}$	R	cRpr	Bst	Cst	(C-B)(%)	(C-P)(%)
1	2	1.4	1.6	1.2	-0.4(-25)	-0.2(-16)
	3	1.8	2.0	1.6	-0.3(-17)	-0.2(-10)
	4	2.1	2.2	1.8	-0.4(-16)	-0.3(-12)
	5	2.5	2.7	2.2	-0.4(-16)	-0.3(-11)
	6	3.2	3.0	2.5	-0.6(-19)	-0.7(-23)
	7	3.1	2.8	2.7	-0.1(-3)	-0.4(-13)
	8	3.0	3.1	2.7	-0.4(-12)	-0.3(-10)
	9	3.0	3.2	2.5	-0.7(-23)	-0.5(-17)
	μ	2.5	2.6	2.2	-0.4 (-16.0)	-0.4(-14.2)
2	2	13.8	10.5	3.3	-7.2(-69)	-10.6(-76)
	3	14.2	11.6	5.0	-6.6(-57)	-9.2(-65)
	4	15.4	10.9	6.9	-4.0(-37)	-8.5(-55)
	5	14.7	11.4	7.3	-4.1(-36)	-7.4(-50)
	6	13.9	9.3	8.1	-1.3(-13)	-5.8(-42)
	7	19.5	9.6	8.5	-1.1(-11)	-11.0(-57)
	8	18.0	9.6	8.3	-1.3(-14)	-9.6(-54)
	9	18.3	11.0	8.1	-3.0(-27)	-10.2(-56)
	μ	16.0	10.5	6.9	-3.6 (-34.1)	-9.1(-56.7)
3	2	2.3	2.6	2.0	-0.6(-24)	-0.3(-14)
	3	4.1	3.5	3.1	-0.4(-12)	-1.0(-25)
	4	5.0	4.3	4.3	0.0(0)	-0.7(-14)
	5	6.2	4.9	4.4	-0.5(-10)	-1.8(-29)
	6	6.5	7.0	5.5	-1.5(-21)	-1.0(-15)
	7	7.6	8.0	6.7	-1.2(-15)	-0.9(-11)
	8	6.7	7.6	6.1	-1.5(-20)	-0.6(-9)
	9	7.8	10.1	7.1	-3.0(-30)	-0.7(-9)
	μ	5.8	6.0	4.9	-1.1 (-18.2)	-0.9(-14.9)

$\mathcal{S}$	R	cRpr	Bst	Cst	(C-B)(%)	(C-P)(%)
4	2	14.0	10.5	5.4	-5.1(-48)	-8.6(-61)
	3	11.7	12.4	12.1	-0.3(-3)	0.4(4)
	4	11.1	11.2	11.3	0.1(1)	0.2(2)
	5	9.7	9.9	10.0	0.1(1)	0.3(3)
	6	9.8	7.4	7.3	-0.1(-2)	-2.5(-25)
	7	8.5	8.1	4.9	-3.2(-39)	-3.6(-42)
	8	8.1	10.6	8.7	-2.0(-19)	0.6(7)
	9	7.7	11.1	8.9	-2.2(-20)	1.2(15)
	μ	10.1	10.2	8.6	-1.6 (-15.5)	-1.5(-14.8)
5	2	4.4	3.2	1.7	-1.4(-45)	-2.7(-61)
	3	3.7	3.4	1.8	-1.6(-46)	-1.9(-51)
	4	3.8	4.8	3.1	-1.8(-37)	-0.7(-19)
	5	4.1	4.6	4.2	-0.4(-9)	0.1(2)
	6	3.5	3.6	2.2	-1.4(-39)	-1.3(-38)
	7	3.5	3.2	3.2	-0.1(-2)	-0.3(-9)
	8	3.3	3.5	2.2	-1.4(-38)	-1.1(-34)
	9	3.2	3.1	3.0	-0.1(-3)	-0.2(-7)
	μ	3.7	3.7	2.7	-1.0 (-27.6)	-1.0(-27.7)
6	2	35.8	43.7	34.0	-9.7(-22)	-1.8(-5)
	3	38.9	35.1	20.5	-14.6(-42)	-18.4(-47)
	4	36.7	33.5	35.7	2.1(6)	-1.0(-3)
	5	35.0	34.5	22.6	-12.0(-35)	-12.4(-35)
	6	31.2	18.7	11.1	-7.6(-41)	-20.1(-64)
	7	28.6	28.6	28.8	0.1(1)	0.2(1)
	8	24.6	24.8	25.1	0.4(2)	0.5(2)
	9	25.5	27.1	25.7	-1.4(-5)	0.2(1)
	μ	32.0	30.8	25.4	-5.3 (-17.3)	-6.6(-20.6)

文章阅读（二）

标题：SMOTEBoost: Improving Prediction of the Minority Class in Boosting

摘要

针对不平衡分类问题，文章提出了一种基于 sampling 的 boosting 方法来提升现有算法实验效果。文章将 over-sampling 作为解决 imbalance 问题的方法，

算法

该算法通过给样本赋予不同的权重（也可以理解为抽样的概率）来达到构建平衡数据集的功能。在每一次迭代中，在样本权重的基础上使用 SMOTE 算法对小类别样本进行样本模拟构建（即增加小类别样本的数量），在新的数据集中训练弱分类器，计算分类问题的残差（或者说误判损失），利用残差对样本权重进行更新。在反复迭代的基础上，SMOTEBoost 算法就可以得到一个可以对抗不平衡问题的分类器了。

- Given: Set S $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ $x_i \in X$, with labels $y_i \in Y = \{1, \ldots, C\}$, where C_m, ($C_m < C$) corresponds to a minority class.
- Let $B = \{(i, y): i = 1,\ldots,m, y \neq y_i\}$
- Initialize the distribution D_1 over the examples, such that $D_1(i) = 1/m$.
- For $t = 1, 2, 3, 4, \ldots T$
  1. Modify distribution D_t by creating N synthetic examples from minority class C_m using the SMOTE algorithm
  2. Train a weak learner using distribution D_t
  3. Compute weak hypothesis h_t: $X \times Y \rightarrow [0, 1]$
  4. Compute the pseudo-loss of hypothesis h_t:
     $$\varepsilon_t = \sum_{(i,y)\in B} D_t(i,y)(1 - h_t(x_i, y_i) + h_t(x_i, y))$$
  5. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$ and $w_t = (1/2)\cdot(1-h_t(x_i,y)+h_t(x_i,y_i))$
  6. Update D_t : $D_{t+1}(i, y) = (D_t(i, y) / Z_t) \cdot \beta_t^{w_t}$

     where Z_t is a normalization constant chosen such that D_{t+1} is a distribution.
- Output the final hypothesis: $h_{fn} = arg\ \max_{y\in Y} \sum_{t=1}^{T} (\log \frac{1}{\beta_t}) \cdot h_t(x, y)$

Fig. 1. The SMOTEBoost algorithm

实验

Table 2. Summary of data sets used in experiments

Data set	Number of majority class instances			Number of minority class instances		Number of classes
KDDCup-99 Intrusion	**DoS**	**Probe**	**Normal**	**U2R**	**R2L**	5
	13027	2445	17400	136	1982	
Mammography	10923			260		2
Satimage	5809			626		2
Phoneme	3818			1586		2

数据集：这篇文章中一共使用了 4 个数据集，其中 3 个数据集为二分类数据集，一个数据为多分类数据集。

实验结果：

Method			*Recall*	*Precision*	*F-value*	***Method***			*Recall*	*Precision*	*F-value*
Standard RIPPER			57.35	84.78	68.42	Standard Boosting			80.15	90.083	84.83
	N_{u2r}	N_{r2l}	*Recall*	*Precision*	*F-value*		N_{u2r}	N_{r2l}	*Recall*	*Precision*	*F-value*
SMOTE	100	100	80.15	88.62	84.17	SMOTE -Boost	100	100	84.2	93.9	**88.8**
	300	100	74.26	92.66	82.58		300	100	87.5	88.8	88.15
	500	100	68.38	86.11	71.32		500	100	84.6	92.0	88.1
First SMOTE then Boost	N_{u2r}	N_{r2l}	*Recall*	*Precision*	*F-value*	Ada-Cost	*Cost factor*		*Recall*	*Precision*	*F-value*
	100	100	81.6	90.92	**86.01**		c = 2		83.1	96.6	**89.3**
	300	100	82.5	89.30	85.77		c = 5		83.45	95.29	88.98
	500	100	82.9	89.12	85.90						

案例分析报告

在本次综述类文章阅读过程中，我们首先了解到当前阶段使用 ensemble 系列方法（一般 ensemble 的基础方法包括：bagging, boosting 和 hybrid）解决不平衡 classification 分类问题（即，样本集合中某一种（或多种）类别的样本数量只占很小的一部分，在平均权重准确率的衡量标准下, 分类器倾向于忽略小类别样本, 只关注大类别样本的成功判断这类问题）的大致情况。一般地，要解决数据不平衡问题，要么从算法入手，根据问题本身设计不同的损失函数（通常是基于类别不同的重要性设计误判损失，使误判小类别样本会产生更大的损失），驱使算法更多关注小类别样本的成功判别概率，使分类器更贴合现实问题的要求；另一种方法即从数据层面入手，既然数据不平衡，一种简单的思路就就是通过删减过多的无用大类别样本，或者生成数据量过少的小类别样本，实现样本数据的平衡，在这种情况下可以不需要对损失函数进行调整，也能有较好的判别效果。于是，从上述两种角度出发，在基础的不平衡分类算法中加入集成运算的思路, 期望得到更快、更好的算法效果。在这种情况下，我选择了基于 boosting 的两篇文章进行精读，期望在选题。数据、方法、写作、学术规范的角度下进行比较和分析。

一、选题。

从选题上，两篇文章都是基于现有方法进行提升，但是基础方法不同。

Adacost 在 Adaboost 分类算法的基础上，加入了 cost-sensitive 代价敏感的元素，使得 adaboost 算法结构中损失函数部分不再是简单的 0-1 损失，而是样本标签相关的损失矩阵, 在新损失函数的监督下, adaboost 在每个迭代周期（iteration）所更新的样本权重会更加注重小类别样本，从而使得最终的分类器能够适用于类别不均衡的显示问题中。这种算法的设计，来源于一个优秀的算法，和一个新颖的研究领域之间恰当的融合。但从 critical thinking 的角度出发，这种 cost-sensitive 的思路和 adaboost 算法的结合的来源是什么，是什么驱动着两者的结合，从理论需求上其实是不明显的，虽然作者给出了 Adacost 算法相对于 adaboost 算法更好的算法上界，但是从实验的优越性上似乎更吸引 reviewer 的赞同。

SMOTEBoost 是基于 SMOTE（Synthetic Minority Oversampling Technique）这个在数据不平衡分类领域熟知的算法，合并上 boosting 算法的框架的一个组合算法。SMOTE 站在数据的角度，通过学习小类别样本的分布性质，在样本集中 synthetically 地生成新的小类别样本，从而使得不平衡的问题减轻或消弭。这篇文章的思路和 Adacost 类似，将两个新兴的 topic 结合在一起，通过做实验得到超过 benchmark 效果的算法。

这两篇文章相比于传统统计，可能更偏向于机器学习或者计算机；相比于算法的理论性质，更关注算法在实际运用中的效果和效果；相比于解决现阶段不平衡分类问题中存在的问题，更关注新算法的设计和实现。

以上就是我对上述两篇文章选题的思考。

二、数据

与案例分析型论文不同，算法型论文一般没有提出自己的数据集。相反，算法型论文往往会约定俗成地使用相同的数据集，以便于和社区中其他学者提出的算法在效果上进行比较和分析。Adacost 中引用了 5 个 UCI 机器学习数据库中的不平衡分类数据，还有一个由其他论文提供的金融类真实数据；SMOTEBoost 类似地也引用了以往论文中出现并公开的不平衡数据集。

一般来说，不同数据集的选择可以帮助 reviewer 理解算法优劣势，即算法更

适合处理什么样的数据。一般地，我们对数据集的观察有：数据集样本数，样本特征维度，数据集稀疏性，数据特征类型（int/float/categorical 等等）。例如 svm 在处理高维度大样本时，就会出现模型训练时间成本急剧上升等问题。从数据集的选择上我们可以判断出，相比于 cost-sensitive 方法，under-sampling 方法似乎对于数据集形式的要求更小，不仅可以处理二分类问题，并且也可以轻松地移植到多分类问题中。

最后，数据也是影响我们算法效果的重要因素。在实验过程中，算法型论文常常会将自己提出的方法和社区中其他处理相同问题的算法相比较，在相等的实验条件下（超参数的选择，数据清晰和降维等等）得到准确度或其他衡量标准的比较，从而更有力地说明文章提出算法的优秀性质。

三、方法

事实上，较为宏观、整体上的算法解释我们在之前的论文阅读中就已经提到过，一般来说，算法型文章会使用算法流程图对自己提出的算法进行简要的总结，然后通过附加的文字对算法流程图进行详细的说明；在实验部分，还会更细致的介绍实验过程中超参数的选择、参数分析和设定，“train-test-validation”数据分割等细节，只有在这些内容完备的情况下，读者才可以复现新算法，验证可靠性。

Adacost:

- Given: $\mathcal{S} = \{(x_1, c_1, y_1), \ldots, (x_m, c_m, y_m)\}$; $x_i \in \mathcal{X}, c_i \in \mathbb{R}^+, y_i \in \{-1, +1\}$.
- Initialize $D_1(i)$ (such as $D_1(i) = c_i / \sum_j^m c_j$).
- For $t = 1, \ldots, T$:
  1. Train weak learner using distribution D_t.
  2. Compute weak hypothesis $h_t : \mathcal{X} \to \mathbb{R}$.
  3. Choose $\alpha_t \in \mathbb{R}$ and $\beta(i) \in \mathbb{R}^+$.
  4. Update

$$D_{t+1}(i) = \frac{D_t(i)\exp\Big(- \alpha_t y_i h_t(x_i) \boxed{\beta(i)} \Big)}{Z_t}$$

  where $\beta(i) = \beta\Big(\text{sign}(y_i h_t(x_i)), c_i\Big)$ is a cost-adjustment function. Z_t is a normalization factor chosen so that D_{t+1} will be a distribution.
- Output the final hypothesis:

$$H(x) = \text{sign}(f(x)) \text{ where } f(x) = \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

Figure 1: AdaCost

输入： x 是自变量，y 是二分类相应变量，c 是正实数的损失因子。

初始化：我们首先使用损失因子对每个样本的权重进行初始化。

训练：不妨设我们 Boosting 的最大迭代次数是 T，在每个迭代周期中，1.我们首先使用给定权重的数据集训练一个弱分类器（注意到，这里的弱分类器是未知的，一般由 boosting 算法设计时确定弱分类器的函数空间），2.给出判定条件，3.并计算相应的权重值，4.更新所有样本的权重。

输出：最终的分类器是所有迭代周期中分类器的加权和。

AdaCost 的思路是，在每一次迭代过程中，赋予容易判断错误的点更大的权重，从而使下一周期迭代的过程中，算法更加关注这些容易判断错误的点。与此同时，由于小类别样本 minority class 的存在，我们还需要使小类别样本在被误判时，有相对更大的权重，因此文章给出了β参数的定义，期望能够使算法更关注小类别样本。

SMOTEBoost:

- Given: Set S $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ $x_i \in X$, with labels $y_i \in Y = \{1, \ldots, C\}$, where C_m, ($C_m < C$) corresponds to a minority class.
- Let $B = \{(i, y): i = 1,\ldots,m, y \neq y_i\}$
- Initialize the distribution D_1 over the examples, such that $D_1(i) = 1/m$.
- For $t = 1, 2, 3, 4, \ldots T$
  1. Modify distribution D_t by creating N synthetic examples from minority class C_m using the SMOTE algorithm
  2. Train a weak learner using distribution D_t
  3. Compute weak hypothesis h_t: $X \times Y \rightarrow [0, 1]$
  4. Compute the pseudo-loss of hypothesis h_t:
     $$\varepsilon_t = \sum_{(i,y)\in B} D_t(i,y)(1 - h_t(x_i, y_i) + h_t(x_i, y))$$
  5. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$ and $w_t = (1/2)\cdot(1-h_t(x_i,y)+h_t(x_i,y_i))$
  6. Update D_t : $D_{t+1}(i, y) = (D_t(i, y) / Z_t) \cdot \beta_t^{w_t}$

     where Z_t is a normalization constant chosen such that D_{t+1} is a distribution.
- Output the final hypothesis: $h_{fn} = \arg\max_{y\in Y} \sum_{t=1}^{T} (\log \frac{1}{\beta_t}) \cdot h_t(x, y)$

Fig. 1. The SMOTEBoost algorithm

输入：x 是自变量，y 是相应变量（分类），并限制小类别标签。

初始化：将每个样本的权重初始化为均匀的。

训练：在 T 个迭代周期中，1.使用 SMOTE 算法，根据当前权重分布生成新的小类别样本，2.在新生成的数据集中训练弱学习器，3.给出判定准则，4.计算误判损失，5.计算权重参数 6.更新样本权重。

输出：最终的分类器是所有迭代周期中分类器的加权和。

可以看出，两种算法其实都是以 adaboost 分类器为基础，通过调整样本分布或者误判损失来达到处理 imbalance 问题的。

四、写作

写作上，两篇文章都是以会议论文的格式进行写作的。写作格式上，“摘要-简介-算法-（理论）-实验-总结”的框架都是完整的，所有出现的数学符号都有相应的说明和阐释，在基础的写作上没有任何问题。

唯一可能存在的不足点是，文章似乎并没有强调出将 boosting 引入 imbalance 问题的必要性和重要性，更像是两个热门 topics 结合的产物，这是需要我们注意的。因为热门在意味着关注度的同时，也意味着有许多学者也同样在研究该问题，我们如何通过写作来展示自己的特点和必要性，在这种情况下变得尤其重要。

五、学术规范

两篇文章在引用文献及格式上都没有任何问题。在论文写作的过程中，如果使用 Latex 程序进行写作，我们可以使用（对应会议论文或者期刊论文）格式文件对排版等细节进行限制。此外，文献的使用和应用也可以直接由 bibtex 功能实现。一般地，在算法型论文中，我们会在“简介”和“相关研究”部分大量地引用文献，来告诉 reviewer 近几年来我们研究的 topic 是非常热门且对社区有极大贡献的，同时这也是帮助没有接触过该领域的人梳理历史研究脉络的机会，在这一部分我们应该通过引文来说明，我们提出的算法、理论或者思想是重要的，是能够解决社区中存在问题的。