第 33 卷 第 9 期 计 算 机 工 程 2007 年 5 月
Vol.33 No.9 Computer Engineering May 2007
·人工智能及识别技术· 文章编号：1000—3428(2007)09—0191—03 文献标识码：A 中图分类号：TP311

用改进的 1-DNF 算法获取最强反例集合的方法

赫枫龄，左万利，于海龙
（吉林大学计算机科学与技术工程学院，符号计算与知识工程教育部重点实验室，长春 130012）

摘 要：利用正样例集合和未标识样例集合获取初始的最强反例集合是使用两步框架方法构造一个面向PU问题文本分类器的基础。该文指出了使用 1-DNF算法抽取初始的最强反例集合的局限性，提出了对算法 1-DNF的改进方法。实验结果表明，与原算法相比，它大大增加了获取的最强反例数目，加快了算法的收敛速度，提高了分类器的精度。

关键词：文本分类；面向PU问题的文本分类；文本分类器

Method for Extracting Strongly Negative Data Set by Improved 1-DNF Algorithm

HE Fengling, ZUO Wanli, YU Hailong
(College of Computer Science and Technology Engineering, Jilin University, Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Changchun 130012)

【Abstract】 Extracting initial strongly negative data set from positive data and unlabeled data is a base for constructing a PU-oriented text classifier by two stage frame method. The limitations in the 1-DNF algorithm for getting initial strongly negative data set are described. An improved 1-DNF algorithm is proposed. The experiment result demonstrates the number of initial strongly negative examples got from positive data and unlabeled data is increased greatly, compared with original 1-DNF algorithm. The convergence speed of algorithm is accelerated, and the precision of the classifier is raised.

【Key words】 Text classification; PU-oriented text classification; Text classifier

1 概述

构造传统的文本分类器使用的训练集合通常由正例集合(P 集合: Positive)和反例集合(N 集合:Negative)组成，利用 P 集合和 N 集合来构造文本分类器，然后用这个分类器对新的文档进行分类处理就可以了。在现实文本分类器的构造过程中，类别的标识是一个非常枯燥、费时费力的工作，尤其是反例集合 N 的收集更是非常棘手，为了使分类器的精度较高，反例集合 N 通常应该是无偏的，即反例集合 N 应包含非正例类别的所有类别。由于反例集合的收集比较困难，人们转而研究基于 PU 问题的文本分类，所谓基于 PU 问题的文本分类，主要是从训练集合特点的角度进行称呼的，基于 PU 问题的分类就是指用于构造分类器的训练样本集合是由已标识的 P 集合和一个全局的未标识 U(Unlabeled)集合组成，而不存在标识的 N 集合，它的特点在于不需要手工进行反例集合 N 的收集，只使用标识的正例集合 P 和一个全局的未标识集合 U 来构造分类器，其中 U 集合中的每一个样例都是随机选取的，而且通常数量远远大于标识的 P 集合中样本的数量。

PU问题文本分类存在着容易收集训练样本集合的优越性，现对它的研究也日益增多。Denis.F [1]进行了使用正例标识集合P和全局未标识集合U进行训练学习的理论研究，并通过试验给出了K-DNF和决策树都可以用于解决PU问题,并阐述了未标识集合U在分类中起到的作用。Bing Liu提出了S-EM算法[2,3](将贝叶斯和EM算法结合起来)构造分类器，取得了很好的效果，并给出了基于PU问题的两步分类的理论框架，采用两步分类的理论框架来构造PU问题的文本分类器由如下 2 步组成：(1)利用P集合中的一些信息从未标识集合U中获取最强反例集合N，构造一个初步的分类器；(2)利用这个分类器，对U集合剩下的元素进行分类，反复迭代，扩大N集合，从而得到最终的分类器。

Bing Liu[2,3]通过实验给出了使用正例集合P和未标识集合U构造分类器的可能性正确的条件：“ 最大化未标识集合U中反例的个数，同时保证正例被正确分类 ”。在如上所述的第(1)步中尽可能多地从未标识集合U中获取最强反例是用 2 步方法构造PU问题的文本分类器的前提，本文采用了改进的1-DNF算法[4]用于获取最强反例集合N，实验结果表明，与原算法相比，它大大增加了获取的最强反例数目，减少了第(2)步构造分类器的迭代次数，加快了算法的收敛速度。

2 算法 1-DNF 的局限性

为了从未标识集合U中获取反例的信息，首先要利用文本特征在正例样本集合和未标识样本集合中出现频率的不同来抽出反例集合。例如：如果一个文本特征在正例集合中出现频率大于 90%，而其在未标识集合出现的频率仅有 10%，这样的特征就可以把它当成正例的特征。通过这样的特征在正例集合和未标识集合中出现的频率不同，首先建立一个正例特征集合。而如果未标识集合U中的样例文档未包含任何这样的特征的，就可以把它从未标识集合U中抽取出来，标识成反例，加入训练集合。根据这一思想，Hwanjo Yu提出了

基金项目：国家自然科学基金资助项目“ 具有增量特性的移动式主题爬行技术 ”(60373099)
作者简介：赫枫龄(1962－)，男，教授，主研方向：Web 挖掘与网络搜索引擎；左万利，博士、教授；于海龙，硕士
收稿日期：2006-06-19 **E-mail：**wanli@mail.jlu.edu.cn

1-DNF[4] 算法，用于从未标识集合U中获取最强反例集合N。

1-DNF方法首先根据特征在标识的正例集合出现的频率大于在未标识集合U中出现的频率构造正例特征集合*PF*。1-DNF基于这样的假设：一篇未标识集合的文档如果没有任何*PF*中正例特征出现，则可认为此文档为可信的反例，1-DNF划分的反例较为准确，而且Bing Liu和Hwanjo Yu[4]也分别做过相关的实验，寻找可信反例的准确率较高。

但是算法 1-DNF 对反例的选取要求过于严格，且存在着不合理性。例如如果某个特征在正例集合样本文档中出现的概率为 1%，在未标识集合的样例文档中出现的概率为 0.5%，使用 1-DNF 算法，该特征也会被加入正例特征集合，这样就会导致正例的集合过于庞大。所以，使用 1-DNF 算法得到的反例非常少，在某些情况下甚至为 0，非常不利于第(2)步的迭代算法。本文对 1-DNF 算法进行了改进，应用于用 2 步方法构造的分类器 WVC，并通过实验证明，获得的可信反例的数量比原算法大大增加。

3 从 PU 文本集合中获取最强反例集合

算法 1-DNF所定义的正例特征集合为：在P中出现的频率大于其在U中出现频率的特征。这个算法中明显存在不足之处，就是它仅仅考虑一个特征在P和U中出现的频率的差异来定义所谓正例特征，而没有考虑其本身在P中出现的绝对频率。例如，一个特征在P中出现的频率为 1%，在U中出现的频率为 0.7%，这样的特征明显不是一个正例的特征，但是1-DNF则把它加入了正例特征集合。这样导致的结果就是正例特征集合非常庞大，在进行最强反例集合NEG_0的提取过程中，NEG_0集合内的文本样例过少，甚至没有，对于此后的迭代训练也是非常不利的(初期偏差得太多，很容易导致迭代的偏差越来越大)。基于此点，本文对 1-DNF进行了改进，不仅仅利用特征在P和U中出现的频率的不同，而且考虑了其在P集合内出现的频率。设定阈值为 λ%，规定必须同时满足以下两种情况的特征才纳入正例特征集合：(1)在P集合中出现的频率大于其在U集合中出现的频率；(2)在P集合中出现的频率大于 λ% (此 λ% 通过实验获得)。称此算法为改进的1-DNF，下面对此算法作以详细的介绍：

确定训练样本集合中的特征集合 $\{x_1, x_2, ..., x_n\}$，其中 $x_i \in U \cup P$。

$for(i=1, i<=n, i++)$

$\quad if(freq(x_i,P)/|P| > freq(x_i,U)/|U| \&\& freq(x_i,P)/|P| > \lambda\%)$

$\quad\quad PF = PF \cup \{x_i\}$

$RN = NULL$

for(每一篇文档 $d \in U$)

$\quad if(\forall x_j \in PF \&\& freq(x_j,d) == 0)$

$\quad\quad RN = RN \cup \{d\}$

$\quad\quad U = U - \{d\}$

其中，$|P|$为正例集合P中样本的数量，$|U|$为未标识集合U中样本的数量，$freq(x_i, P)$为特征 x_i 在P集合中出现的次数，$freq(x_i, U)$ 为特征 x_i 在U集合中出现的次数。在我们的构造文本分类器WVC中，就是利用上面提到的改进的 1-DNF算法构造最初的最强反例集合NEG_0。通过试验可看出，使用改进的 1-DNF的算法，可以大大提高初次获得最强反例的数量。

4 用改进的算法构造文本分类器及实验结果分析

4.1 用改进的 1-DNF 算法构造文本分类器

本课题组应用改进的 1-DNF 算法设计了一个针对 PU 问题的文本分类器 WVC，由于受到篇幅限制，本文对文本分类器 WVC 没有进行详细的描述，下面对它进行简单概述：

(1)对文档进行预处理：移去终止词)StopWords)，并采用 *tfc* 权重表示文本向量。*tfc* 权重表示方法与 *tfidf* 相比，它的优点在于它不仅仅考虑一个特征在全局和本文档出现的情况，并且考虑了本篇文档的长度情况。毕竟，一个特征在一个 1 000 字的文档中出现 1 次和其在 100 字的文档出现一次的权重应该是不同的。对于文档的长度，是在形成文档特征向量过程中不可不考虑的因素。文档特征向量形成后，如果没有经过特征抽取，特征的维数是相当大的。为了减少运算的复杂性，缩短运算的时间，应该对文档进行特征降维。我们只是简单地移去在全局文档中出现次数小于 5 的特征，这样就形成了所需的最终特征向量。

(2)用改进的 1－DNF算法获取初始的最强反例集合NEG_0。

(3)利用NEG_0，迭代使用SVM算法构造文本分类器。

4.2 实验结果分析

本文在实验中使用“ Reuters 21578 ”[5] 作为训练和测试的数据集，这个数据集收集了 21 578 个文档，分成了 135 个类别，但是在进行实验时，通常使用的是最高的 10 个类别。本文在试验中用这 10 个类别进行分类器构造及其性能测试。

在构造 PU 问题的过程中，本文采用了和 Liu Bing 一样的做法，随机取每个类别的 70%作为训练集合，剩余的 30%用于测试。为了模拟 PU 问题，本文随机取 γ% 的正例文档(γ 取值范围是从 10～50，步长为 10)和其它的未标识集合组成本文实验所需的 U 集，剩余的(1－γ%)的正例文档作为 P 集，针对不同的 γ 值，分别做实验验证本文的方法在正例集合剧减的情况下的稳定性。实验中给出的各项结果均为在不同的 γ% 情况下的算术平均值。

从 U 集合中寻找最强反例的步骤中，本文对 1－DNF 的改进时引进了参数 λ%，且规定正例特征不仅要满足在正例集合中出现的频率大于在未标识集合中出现的频率，而且还要满足其在正例集合中出现的绝对频率大于 λ%。本文对 λ 取不同值分别进行了实验(λ 的取值范围是 10～90，步长为 10)，最后选取最好的结果作为 λ 的最终值。

下面对实验产生的结果进行分析。首先，对本文改进后的 1-DNF 算法与 1-DNF 算法进行综合比较，从二者获得的反例样本个数、获得的反例中的错误率两方面进行比较。因为篇幅的原因，并没有将每次运行的结果在文中列出，而是将不同 γ% (掺入未标识集合中正例的百分比)得到的结果取算术平均值进行比较，结果见表 1。

表 1 中 UN 表示获得的可信反例个数，(10%～90%)分别表示我们改进的 1-DNF 算法在不同 λ% 的情况， ERR(%) 的计算公式如下：

$$\mathrm{ERR}(\%) = \frac{\text{获得可信反例中包含正例的个数}}{\text{未标识集合中掺入的正例个数}}$$

进一步计算在相同的 λ% (正例特征在正例集合的绝对频率)条件下，所有类别 UN 和 ERR(%)的算术平均值，列表如表 2。从表 2 中可以看出改进的 1-DNF 随着 λ% 的增大，反例获取量逐渐增大，但是错误率也逐渐增大。但是要注意到， λ% ＝10%时，正例的误分率仅为 0.19%，但是分得反例数量却是 1-DNF 的 4.4 倍； λ% ＝20%时，正例的误分率在 2%之内，但是分得的反例数量是 1-DNF 的 7.6 倍，说明本文通过对 1-DNF 算法的改进，在正例误分率很低的情况下，大大扩大了反例集合的数量。

但是这种改进是否能增加最终分类器的分类准确率，还要看实验得到的分类器性能，下面将本课题组实现的分类器 WVC 的性能)F_1 指标[4]) 与 PEBL 算法及其仅基于正例的

One-Class SVM算法[6]进行比较，实验结果见表 3。

表 1 改进的 1-DNF 算法与 1-DNF 算法提取反例个数与错误率结果

		Acq	Corn	Crude	Earn	Grain	Interest	Money	Ship	Trade	wheat
λ%=10%	UN	792.4	802.2	480.2	787.0	890.8	1 250.0	1308.6	1 356.4	383.4	975.6
	ERR(%)	0.98	0.00	0.22	0.26	0.00	0.11	0.00	0.33	0.00	0.00
λ%=20%	UN	1 003.8	1 343.0	1 189.6	1 200.4	1 570.4	2 185.2	2 087.2	2 128.2	1 034.2	1 791.0
	ERR(%)	2.52	0.00	0.63	0.66	0.91	2.88	2.67	2.28	0.44	0.00
λ%=30%	UN	1 115.4	1 577.8	1 296.6	1 689.0	1 660.0	2 267.6	2 245.2	2 600.8	1 164.6	2 017.0
	ERR(%)	4.07	0.00	0.63	3.22	0.91	3.32	5.19	4.51	0.90	0.00
λ%=40%	UN	1 151.8	1 731.4	1 678.8	1 706.8	2 523.6	2 330.6	2 393.0	2 680.8	1 305.8	2 562.0
	ERR(%)	4.51	0.48	0.63	3.35	2.26	3.32	7.83	12.50	3.24	0.25
λ%=50%	UN	1 603.0	2 570.4	2 599.8	2 622.8	2 535.0	2 356.8	2 491.6	2 683.2	1 813.8	2 568.2
	ERR(%)	6.77	0.48	4.53	5.60	7.59	4.52	9.93	13.10	3.58	0.25
λ%=60%	UN	2 593.6	2 586.4	2 601.6	2 636.6	2 557.8	2 392.2	2 560.8	2 683.2	2 635.8	2 589.0
	ERR(%)	9.37	2.64	6.34	5.91	19.10	4.67	13.18	13.10	3.58	0.25
λ%=70%	UN	2 593.6	2 619.6	2 601.6	4 198.8	2 557.8	2 478.4	2 629.8	2 683.2	2 635.8	2 617.2
	ERR(%)	9.37	16.91	6.34	15.93	19.10	7.31	17.18	13.10	3.58	0.25
λ%=80%	UN	2 593.6	5 844.6	2 601.6	5 023.2	6 314.0	5 482.6	2 629.8	2 683.2	2 635.8	2 617.2
	ERR(%)	9.37	85.75	6.34	100.00	100.00	70.23	17.18	13.10	3.58	0.25
λ%=90%	UN	4 543.2	6 652.4	6 246.4	5 023.2	6 314.0	6 622.2	6 481.2	6 755.0	5 517.2	6 476.2
	ERR(%)	64.14	100.00	10.13	100.00	100.00	100.00	100.00	100.00	6.52	1.50
1-DNF	UN	161.0	203.4	116.6	269.6	128.0	299.0	269.6	324.4	70.8	200.2
	ERR(%)	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.33	0.00	0.00

表 2 所有类别 UN 和 ERR)%)的算术平均值

	1-DNF	10%	20%	30%	40%	50%	60%	70%	80%	90%
UN	204.26	902.66	1 555.3	1 763.4	2 006.46	2 384.46	2 583.7	2 761.58	3 842.56	6 063.1
ERR(%)	0.035	0.19	1.299	2.275	3.837	5.635	7.814	10.907	40.58	68.229

表 3 WVC 与 PEBL、OCS 的性能对比

	Acq	Corn	Crude	Earn	Grain	Interest	Money	Ship	Trade	Wheat	ALL
WVC (λ%=20%)	0.964 8	0.734 6	0.889 1	0.981 3	0.941 6	0.852 8	0.885 6	0.806 7	0.883 6	0.797 1	0.873 72
PEBL	0.953 8	0.721 4	0.856 4	0.979 9	0.908 3	0.839 0	0.868 6	0.774 9	0.863 3	0.788 0	0.855 38
OCS	0.708 2	0.427 1	0.676 6	0.914	0.664 1	0.713 1	0.718	0.422	0.6945	0.521	0.645 86

表 4 WVC 与 PEBL 中 SVM 算法的迭代次数对比

	Acq	Corn	Crude	Earn	Grain	Interest	Money	Ship	Trade	Wheat	ALL
WVC (λ%=20%)	8.8	6.2	7.6	7.2	6.6	6.6	10.4	6.8	6.6	7.0	7.38
PEBL	9.8	7.2	9.4	7.4	7.8	7.6	10.4	8.6	9.0	9.0	8.62

PEBL表示使用PEBL算法的到最终分类器，OCS表示One-Class SVM算法产生的最终分类器。其中F_1指标的运算方法和UN的计算相同：都是取不同γ% 情况下的算术平均值值。为了有利于观察，上面表格中进一步计算了相同的λ% (正例特征在正例集合的绝对频率)条件下，所有类别的F_1指标的算术平均值——“ ALL 列”。

由表 2 的实验结果可以看出，当λ% ＝20%时，WVC 分类器的分类效果比 PEBL 高 1.73 百分点，One-Class SVM 的效果最差，比 WVC 分类器低 22.7 个百分点，比 PEBL 低 20.9 个百分点。

由于对 1-DNF 算法的改进，使最强反例的获取数量大大增加，这导致本文的第(2)步迭代算法的收敛速度比 PEBL 算法的收敛速度快(这两种分类器构造方法的第(2)步都是采用 SVM 算法)。实验结果得到的最终的迭代次数对比表见表 4。

表 4 中最后一列“ ALL ”是在相同λ% 情况下，对每一类别的迭代次数所取的算术平均值。从表 4 中可看出，采用改进的 1-DNF，能够有效地减少迭代次数，比采用 1-DNF 算法的 PEBL 平均迭代次数少 1.32 次，加快了算法的收敛速度。

参考文献

1 Denis F. PAC Learn ing from Positive Statistical Queries[C]//Proceedings of the 9th International Conference on Algorithmic Learning Theory, Otzenhausen, Germany. 1998-10.

2 Liu B, Lee W S, Yu P. Partially Supervised Classification of Text Documents[C]//Proceedings of the 9th International Conference on Machine Learning, Sydney, Australia. 2002-07: 387-394.

3 Liu B, Dai Y, Li X. Building Text Classifiers Using Positive and Unlabeled Examples[C]//Proceedings of the 3rd IEEE International Conference on Data Mining, Florida, USA. 2003-11: 179-188.

4 Yu H, Han J, Chang K. PEBL: Web Page Classification Without Negative Examples[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(1): 70-81.

5 Tseng Yuen Hsien. Tools for Reuters-21578 Text Categorization Dataset[EB/OL]. 2002-11. http://www.lins.fju.edu.tw/~tseng/Collections/ Reuters-21578.html.

6 Larry M. Manevitz, Malik Yousef. One-class svms for Document Classification[J]. Journal of Machine Learning Research, 2002, 2(2): 139 -154.

（上接第 190 页）

4 结论

本文提出了利用 GA+HR 算法求解矩形装箱问题。结果表明此近似算法能够比其它算法更好。特别是对数据量大的测试问题，执行得更好。因此，GA+HR 在许多工程领域中，如木材、玻璃、造纸工业、船舶制造工业、纺织业和皮革工业中的矩形物体的合理规划有很大的实用价值。随着人们对装箱问题的不断深入研究，将使结果越来越趋近最优解，计算速度也越来越快，装箱算法会在实际生活中得到真正的应用。我们将来的工作就是改进现有的算法，使其能更快更好地解决装箱问题。

参考文献

1 Zhang Defu, Kang Yan, Deng Ansheng. A New Heuristic Recursive Algorithm for the Strip Rectangular Packing Problem[J]. Computers & Operations Research, 2006, 33(8): 2209-2217.

2 Hopper E, Turton B C H. An Empirical Investigation of Meta-heuristic and Heuristic Algorithms for a 2D Packing Problem[J]. European Journal of Operational Research, 2001, 128(1): 34-57.

3 Hadjiconstantinou E, Christofides N. An Exact Algorithm for the Orthogonal 2-D Cutting Problems Using Guillotine Cuts[J]. European Journal of Operational Research, 1995, 83(1): 21-38.

4 Berkey J O, Wang P Y. Two-dimensional Finite Bin Packing Algorithms[J]. Journal of the Operational Research Society, 1987, 38(5): 423-429.

5 Liu D, Teng H. An Improved BL-algorithm for Genetic Algorithm of the Orthogonal Packing of Rectangles[J]. European Journal of Operational Research, 1999, 112(2): 413-419.