Journal of Computer Applications
计算机应用,2012,32(10):2916-2919
ISSN 1001-9081
CODEN JYIIDU
2012-10-01
http://www.joca.cn

文章编号:1001-9081(2012)10-2916-04
doi:10.3724/SP.J.1087.2012.02916

基于损失函数的 AdaBoost 改进算法

雷 蕾*,王晓丹
(空军工程大学 防空反导学院,陕西 西安 710051)
(*通信作者电子邮箱 wendyandpaopao@163.com)

摘 要:针对 AdaBoost 集成时难分样本权重扩张导致训练样本在更新时分布失衡的问题,提出一种基于正负类样本损失函数(LF)的权重更新策略。权重的调整不仅与训练误差有关,还考虑到基分类器对不同类别样本的正确分类能力,从而避免训练样本过度集中于某一类的异常现象。实验结果表明,基于 LF 的 AdaBoost 能在提高收敛性能的情况下,提高算法精度,克服样本分布失衡问题。偏差方差分析的结果显示,该算法在改善偏差的情况下,能有效地减小错误率中的方差成分,提高集成的泛化能力。

关键词:AdaBoost 算法;支持向量机;损失函数
中图分类号:TP18;TP391 **文献标志码**:A

Improved AdaBoost ensemble approach based on loss function

LEI Lei*, WANG Xiao-dan
(*School of Air and Missile Defense, Air Force Engineering University, Xi'An Shaanxi* 710051, *China*)

Abstract: As to the issue that the weight expansion for hardest samples can cause imbalance when updating the training sample in AdaBoost algorithm, an improved approach based on the Loss Function (LF) of the different patterns, namely, LF-AdaBoost, was proposed. The weight tuning was affected not only by the training error, but the performance of base classifiers for different classes, thus avoiding the excessive concentration phenomenon. The results based on UCI data sets and different base classifiers have shown that the approach can improve the speed of convergence and overcome the imbalance, as well as promote the generalization ability of ensemble classifier.

Key words: AdaBoost algorithm; Support Vector Machine (SVM); Loss Function (LF)

0 引言

AdaBoost 算法作为经典的二类分类集成算法,能有效地将多个弱分类器,比如决策树、神经网络等,集成得到一个强分类器。目前,AdaBoost 已经被广泛应用到雷达跟踪[1]、人脸识别[2-4]和目标检测[5]等领域。

权值更新能保证学习算法专注于较难处理的训练样本,是 AdaBoost 的最大优点。然而在 AdaBoost 训练后期会出现权重扩张和训练退化的现象,即某些样本在被多次错分后,其权重将不断增大,使得训练子集过度关注某一类的样本,从而导致生成的基分类器训练误差增大,最后在加权投票时危害分类器的正确预测能力。已有的研究大部分通过改善样本权值的更新过程来提高集成算法的性能[3,6-7],取得了较好的效果。如何通过改变基分类器加权参数来解决这一问题,目前研究较少。

针对以上问题,本文按照基分类器加权求解方式的思路,提出了一种基于正负类损失函数的参数求解算法,该算法采用训练好的基分类器对每一次训练子集的补集进行损失估计,根据得到的损失值更新错分样本权值,调整训练样本在不同类别上的关注程度,使其保持平衡,提高对不平衡数据的分类能力。

1 AdaBoost 集成分析

AdaBoost 算法的实现:依次训练一组基分类器,其中每个基分类器的训练集都是选择由其他基分类器给出的"最富信息"的样本组成,最后用线性加权集成这些基分类器,从而获得最终判决结果[7]。在算法开始时,每个训练样本都被赋予同一个权重,表明该样本被弱分类器选作训练集的概率。如果某个训练样本已经被正确分类,则降低它的权重,那么在下一个弱分类器确定训练集时,它被选中的概率就会被降低;相反,如果某个训练样本没有被正确分类,那么它的权重就会得到提高,则它在下一次训练过程中被选中的概率会增加。通过这样的方式,AdaBoost 的训练过程能够聚焦于那些较难分(更富信息)的样本上。

2 基于 LF 的 AdaBoost 算法

在传统的 AdaBoost 算法中,当训练样本集包含一些较难分的困难样本时,其样本权重会被上一次迭代的弱分类器因错分而增大[6]。如果某个目标类中包含了太多的难分样本,则在若干次循环后该目标类上的样本权重会出现扩张的现象,从而使得下一次选择训练集时样本数据分布扭曲,导致算法失效。本文提出的 LF-AdaBoost 改进算法,通过对样本权重更新的调节,减小权重扩张幅度,达到合理分配权重的目的。

收稿日期:2012-04-25;**修回日期**:2012-06-14。 **基金项目**:国家自然科学基金资助项目(60975026)。
作者简介:雷蕾(1988-),女,四川南充人,硕士研究生,主要研究方向:模式识别、智能信息处理; 王晓丹(1966-),女,陕西汉中人,教授,博士生导师,主要研究方向:智能信息处理、机器学习。

给定有标记的训练样本集 $D:(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)$,其中 $x_i \in X, X \in \mathbf{R}^d, i=1,\cdots,n$。$y_i$ 为类标签,$y_i \in \{\omega_k\}$,$k=1,\cdots,K$。$h_t(x_i)$ 为基分类器 C_t 对样本 x_i 的预测输出。定义损失函数 $l_t(x_i)$ 如下:

$$l_t(x_i)=\begin{cases}0, & h_t(x_i)=y_i\\ 1, & h_t(x_i)\neq y_i\end{cases} \tag{1}$$

其补函数为:$\bar{l}_t(x_i)=1-l_t(x_i)$。

假设 T 为集成算法的迭代次数。在第 t 次迭代时,通过上一次样本分布随机抽取的训练子集 D_t 得到基分类器 C_t,利用 C_t 对本次训练子集的补集 $\tilde{D}_t=D-D_t$ 中的样本进行分类,则 C_t 在此次迭代时的权值可定义为:

$$l_t=\frac{1}{f_t}\left|\sum_{\substack{x_i\in\{\omega_k\}\\ x_j\notin\{\omega_k\}}} l_t(x_i)l_t(x_j)-\sum_{\substack{x_i\in\{\omega_k\}\\ x_j\notin\{\omega_k\}}}\bar{l}_t(x_i)\bar{l}_t(x_j)\right| \tag{2}$$

f_t 为归一化系数,以保证 $\sum_{t=1}^{T} l_t=1$。从式(2)可以看出,l_t 反映了基分类器对属于 ω_k 类样本和不属于 ω_k 类样本的分类能力,保证在正确分类一个 ω_k 类样本时,对一个非 ω_k 类的样本也要正确识别。特别地,当 $i=1,2$ 时,即针对特殊情况下的两类分类问题时,式(2)可以写为:

$$l_t=\frac{1}{f_t}\left|\sum_{\substack{x_i\in\{\omega_1\}\\ x_j\in\{\omega_2\}}} l_t(x_i)l_t(x_j)-\sum_{\substack{x_i\in\{\omega_1\}\\ x_j\in\{\omega_2\}}}\bar{l}_t(x_i)\bar{l}_t(x_j)\right| \tag{3}$$

基于 LF 的基分类器权值更新步骤如下:

输入:基分类器 C_t,训练子集 D_t,迭代次数 T,类别数 K。

输出:权值向量 $\boldsymbol{L}\in[0,1]^{1\times T}$

利用 C_t 对训练子集的补集 $\tilde{D}_t$ 中的样本进行分类

初始化 $\boldsymbol{L}=0$

for $k=1$ to K, N_1 为属于 ω_k 的样本数,N_2 为不属于 ω_k 的样本数

  for $i=1$ to N_1

    for $j=1$ to N_2

      if $h_t(x_i)=y_i$ and $h_t(x_j)=y_j$, $l_t=l_t+1$

      if $h_t(x_i)\neq y_i$ and $h_t(x_j)\neq y_j$, $l_t=l_t-1$

    end

  end

end

归一化向量 $\boldsymbol{L}$

将得到的权值融入到算法中,则 LF-AdaBoost 算法的加权参数求解公式可以写为:

$$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)+l_t \tag{4}$$

传统的 AdaBoost 算法是基于正确率分布对称的分类问题的,即正样本的分类错误率和负样本的错误率在整个训练过程中处于均匀和平等的地位。但因现实中数据集分布的复杂性,训练后期总是过度偏重于某一类的样本,从而使得错误率增大。从式(4)可以看出,LF-AdaBoost 通过对不同类别样本的分类性能来调整训练样本在整个数据集上的偏重程度。当基分类器对正负类中的样本同时分类正确时,l_t 加 1;分类错误时,l_t 就减 1,消除了不同类别上分类错误的比例不平衡对分类器集成的影响。同时,在错误率 ε_t 相同的情况下,对那些对两类样本识别能力更强的基分类器赋予更大的权重,从而提高集成的正确率。

则 LF-AdaBoost 算法的实现流程为:

输入:给定标记的训练样本集 D,迭代次数为 T。

初始化:对样本权值进行初始化:$w_1(i)=1/n$, $i=1,2,\cdots,n$。

For $t=1,2,\cdots,T$

在当前样本分布 $w_t(i)$ 下,随机采样,得到基分类器的 C_t 的训练样本集 D_t;

计算 C_t 的训练误差 $\varepsilon_t=\sum_{i=1}^{n} w_t(i), y_i\neq h_t(x_i)$,即错分样本的权值 w_t 之和;

将在基分类器 C_t 作用于 $\tilde{D}_t$,根据基于损失函数的权值更新方法,得到权值 l_t;

令 $\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)+l_t$,更新训练样本的权值:

$w_{t+1}(i)=\frac{w_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$,其中 Z_t 为归一化系数,使 $\sum_{i=1}^{n} w_t(i)=1$;

输出:集成分类器的判决函数值:

$H(x)=\operatorname{sign}\left(\sum_{i=1}^{n}\alpha_i h_i(x)\right)$

3　实验结果及分析

3.1　实验数据

实验所用的数据来自公共数据集。表 1 为所用数据集属性描述。

表 1　各数据集属性描述

Problem	#Train	#Atts	#Classes
Breast-w	699	9	2
Balance	625	4	3
Diabetes	768	8	2
Glass	214	9	7
Heart-statlog	270	13	2
Ionosphere	351	34	2
Segment	2310	19	7
Sonar	208	60	2
Soybean	683	35	19

3.2　实验设计

实验通过两种不同的基分类器,决策树和 SVM 来验证 LF-AdaBoost 算法的有效性。

SVM 来自 PRTool 工具箱,采用径向基核函数(Radial Basis Function, RBF)。RBFSVM 有高斯宽度 σ 和惩罚因子 C 两个参数,任何一个的改变都导致分类器性能的改变。通过选取合适的模型参数可以有效地避免产生过适应现象。对 RBFSVM 的性能分析发现,当 C 取值很小时,RBFSVM 的学习性能很差,但如果 C 取值较为合适时,RBFSVM 的学习性能主要依赖于 σ 值的变化,即此时 σ 的取值对 SVM 的性能影响更大。对此,我们通过改变径向基核函数的参数 σ 的取值来增大 RBFSVM 个体分类器之间的差异性[9],避免了参数 σ 在所有基分类器中取值相同带来的问题。因此在本文中采用文献[7]的做法,把训练每个基分类器的样本集的标准差作为该基分类器的 σ 值。惩罚参数 $C=1$。实验分别对基于 SVM 的

AdaBoost 算法(SVMAda)和结合 SVM 与损失函数的 AdaBoost 算法(SVM + LF),基于决策树的 AdaBoost 算法(TreeAda)和结合决策树与损失函数的 AdaBosot 算法(Tree + LF)进行了比较实验。采用交叉验证法来确定训练集和测试集。迭代次数为 20 次。

在估计分类错误率时采用十重交叉验证来进行,并利用双边估计 t 检验法来计算置信水平为 0.95 的分类错误率置信区间作为最终结果,计算公式如下:

$$\frac{|\bar{x}-\mu|}{\sigma/\sqrt{n}} \geqslant t_{0.025}(n-1) \tag{5}$$

其中 μ,σ 分别表示十重交叉验证的均值和标准差,$t_{0.025}(9)=2.2622$。

3.3 实验结果和分析

3.3.1 实验结果

表 2 为部分公共数据集在 4 种分类算法下的分类性能比较。从表 2 横向可以看出 LF-AdaBoost 算法的集成分类精度普遍要大于传统的 AdaBoost 集成方法。由此可见其对 AdaBoost 算法确实能产生积极影响。通过基于损失函数的权值更新能更好的平衡训练样本在不同类别上的分布,保证算法顺利进行,提高集成分类精度。而基于决策树的 AdaBoost 对集成性能的提升程度高于基于 SVM 的集成。

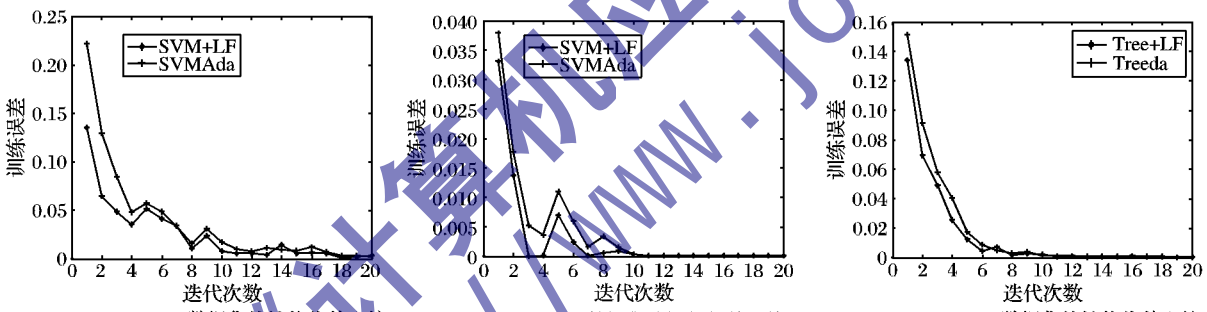
表 2 5 组 UCI 数据集在各分类方法下的分类正确率 %

数据集	分类正确率 ± 置信区间			
	SVM + LF	SVMAda	Tree + LF	TreeAda
Sonar	53.37 ± 0.41	46.63 ± 0.41	74.03 ± 1.39	67.84 ± 2.03
Ionosphere	65.24 ± 5.74	64.10 ± 1.30	90.89 ± 3.11	90.31 ± 6.89
Diabetes	65.10 ± 3.31	50.78 ± 3.31	69.28 ± 11.58	64.22 ± 3.03
Breast-w	95.42 ± 7.19	68.88 ± 3.37	96.57 ± 10.84	87.58 ± 2.16
Heart-statlog	67.04 ± 2.24	59.26 ± 4.71	70.74 ± 1.77	72.22 ± 4.71

然后,本文随机选取不同数据集进行实验来验证基于损失函数的 AdaBoost 的收敛性能。如图 1 所示。

抛开数据集和基分类器差异的影响,大部分情况下基于损失函数的 AdaBoost 集成算法在收敛速度和训练误差上都要好于传统的 AdaBoost 算法。同时,基于 Breast-w 和 Heart-statlog 数据集的实验表明,随着迭代次数的增加,训练误差会明显降低,当达到最小值时偶尔会有所上升,并最终趋于稳定。因为所用数据集的数据不是很庞大,在实验中发现,当迭代次数达到 20 次时,测试误差就趋于理想值。

因缺乏文本实验数据和源代码,实验将本文方法与文献[13]中 MWBoost 方法进行分类结果比较。基分类器选择决策树。表 3 为分类正确率比较结果,分类精度最高的算法以下划线标出。



(a) heart-statlog数据集的性能收敛比较 (b) Breast-w数据集的性能收敛比较 (c) ionosphere数据集的性能收敛比较

图 1 基于损失函数的 AdaBoost 在不同数据集上的性能收敛比较

表 3 三种方法分类正确率比较 %

算法	Breast-w	Balance	Diabetes	Glass	Heart-statlog	Ionosphere	Segment	Sonar	Soybean
AdaBoost	96.31	85.32	67.97	82.55	74.07	91.02	91.22	75.69	92.99
MWBoost	96.14	77.12	71.48	74.30	80.74	93.45	98.14	79.33	93.27
LF-AdaBoost	95.49	88.76	76.09	83.93	82.19	86.94	94.37	81.53	86.92

从实验结果可以看出,在 9 个数据集上,LF-AdaBoost 在 5 个数据集上分类效果最好,在其他分类效果不是最好的情况下,其分类精度也相差不大。而与 MWBoost 相比,在同类数据集上,LF-AdaBoost 的收敛速度更快。

3.3.2 偏差方差分析

为深入了解基于 LF 的 AdaBoost 方法在提高分类效果方面的本质,偏差-方差分析法将被引入以分析其深层次的原因。从理论上讲,一个分类器的误差应该分解为三部分,即贝叶斯误差、偏差和方差。而在实际的学习任务中,类的真实分布一般是未知的,这使我们很难估计出内部误差。考虑到在一个学习任务中,内部误差对几个学习算法是不变的,它不会影响到学习算法之间的相对效率。定义的误差的偏差和方差分解中,内部误差被融入到了偏差和方差中。令 T 为训练集的分布,D 为来自分布 T 的训练样本,L 为基学习算法,$L(D)$ 表示用 L 训练的分类器,则对于验证样本点(x,y),其偏差和方差的定义为:

$$\begin{cases} Bias(x) = P_{D\sim T}(L(D)(x) \neq y \,\&\, L(D)(x) = y*) \\ Var(x) = P_{D\sim T}(L(D)(x) \neq y \,\&\, L(D)(x) \neq y*) \end{cases} \tag{6}$$

其中,$y*$ 是由来自 T 的不同训练集 D 训练的分类器对样本 x 所预测的类标签的中心趋势。

为了计算偏差和方差的值首先需要知道训练数据的真实分布 T。由于在实际问题中,T 一般是未知的,因此只能用某种方法来估计偏差和方差。在此采用 Webb[10] 所提方法来估计偏差和方差。具体步骤如下:首先将数据 D 随机地分为 3 个大小基本一致的集合 f_1,f_2,f_3,并将该过程重复进行 10 次得到 30 个不同集合 $f_1^1,f_2^1,f_3^1,f_1^2,f_2^2,f_3^2,\cdots,f_1^{10},f_2^{10},f_3^{10}$。在每次实验中,将 3 个集合中的每个集合都用作一次验证集,与

其对应的另外两个集合用作训练集。为了简化记号,令 $D_1^i = f_2^i \cup f_3^i, D_2^i = f_1^i \cup f_3^i, D_3^i = f_1^i \cup f_2^i$,则对于 $x \in D$,其类标签的中心趋势可以估计为:

$$\text{argmax}\left(\sum_{i=1}^{10}\sum_{j=1}^{3} I(x \in f_j^i \,\&\, T(D_j^i)(x) = y)\right) \quad (7)$$

为了更直观地比较每种方法对偏差和方差的减小程度,图 2 给出了基于决策树的 LF-AdaBoost 算法和传统 AdaBoost 算法在每个数据集上所对应的偏差和方差。在某些情况下,偏差和方差之和并不严格等于相应的误差,主要是因为对一些值进行了四舍五入。从图 2 中可以看出,与传统的 AdaBoost 算法相比,基于 LF 的改进算法在偏差改善不是很大的情况下,能有效地减小错误率中的方差成分,提高了算法的泛化性能。

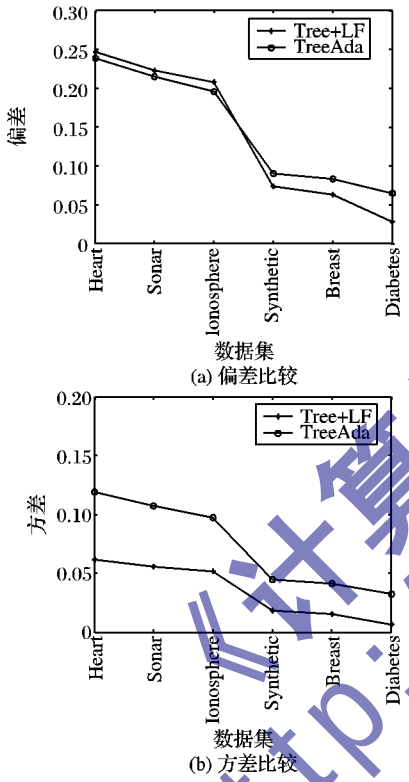


图 2　偏差方差分解示意图

4　结语

本文提出了一种改进的 AdaBoost 算法,利用损失函数维护正负类样本分类正确率的分布从而避免训练样本过度集中于某一目标类。从不同数据集的收敛结果可以看出,基于损失函数的 AdaBoost 算法在性能方面和精度方面明显要好于传统的 AdaBoost 算法,提升了泛化性能,具有潜在的应用价值。

参考文献:

[1] 陈炜, 周晓, 叶菲, 等. AdaBoost-NN 在雷达信号识别中的应用[J]. 电子对抗技术, 2005, 20(1): 29 – 33.

[2] 武勃, 黄畅, 艾海舟, 等. 基于连续 AdaBoost 算法的多视角人脸检测[J]. 计算机研究与发展, 2005, 42(9): 1612 – 1621.

[3] 张君昌, 李倩, 贾靖. 基于分类器相关性的 AdaBoost 人脸检测算法[J]. 计算机应用, 2009, 29(12): 3346 – 3348.

[4] 王艳, 公维军. 双阈值级联分类器的加速人脸检测算法[J]. 计算机应用, 2011, 31(7): 1822 – 1830.

[5] 李闯, 丁晓青, 吴佑寿. 一种改进的 AdaBoost 算法 – AD AdaBoost[J]. 计算机学报, 2007, 30(1): 103 – 109.

[6] 李文辉, 倪洪印. 一种改进的 AdaBoost 训练算法[J]. 吉林大学学报: 理学版, 2011, 49(3): 498 – 504.

[7] 王晓丹, 孙东延, 郑春颖, 等. 一种基于 AdaBoost 的 SVM 分类器[J]. 空军工程大学学报: 自然科学版, 2006, 7(6): 54 – 57.

[8] LI XUCHUN, WANG LEI, SUNG E. AdaBoost with SVM-based component classifiers[J]. Engineering Applications of Artificial Intelligence, 2008, 21(5): 785 – 795.

[9] BAUDAT G, ANOUAR F. Generalized discriminant analysis using a kernel approach [J]. Neural Computation, 2000, 12(10): 2385 – 2404.

[10] WEBB G I. MultiBoosting: A technique for combining boosting and wagging [J]. Machine Learning, 2000, 40(2): 159 – 196.

[11] 付忠良. 多分类问题代价敏感 AdaBoost 算法[J]. 自动化学报, 2011, 37(8): 973 – 983.

[12] 唐焕玲, 鲁明羽, 邬俊. 基于投票信息熵的 AdaBoost 改进算法[J]. 控制与决策, 2010, 25(4): 487 – 492.

[13] 张家红, 张化祥, 刘伟. 标记错分样本的 AdaBoost 算法[J]. 计算机工程与设计, 2010, 31(6): 1294 – 1296.

(上接第 2915 页)

[3] 贺毅朝, 王熙照, 刘坤起, 等. 差分演化的收敛性分析与算法改进[J]. 软件学报, 2010, 21(5): 875 – 885.

[4] FAN HUI-YUAN, LAMPINEN J. A trigonometric mutation operation to differential evolution[J]. Journal of Global Optimization, 2003, 27(1): 105 – 129.

[5] RAHNAMAYAN S. Opposition-based differential evolution[D]. Waterloo, Ontario: University of Waterloo, 2007.

[6] 张利彪, 周春光, 马铭, 等. 基于极大极小距离密度的多目标微分进化算法[J]. 计算机研究与发展, 2007, 44(1): 177 – 184.

[7] 贺毅朝, 王熙照, 寇应展. 一种具有混合编码的二进制差分演化算法[J]. 计算机研究与发展, 2007, 44(9): 1476 – 1484.

[8] 赵光权, 彭喜元, 孙宁. 带局部增强算子的微分进化改进算法[J]. 电子学报, 2007, 35(5): 849 – 853.

[9] RAHNAMAYAN S, TIZHOOSH H R, SALAMA M M A. Opposition-based Differential Evolution (ODE) with variable jumping rate [C]// FOCI'07: Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence. Washington, DC: IEEE Computer Society, 2007: 81 – 88.

[10] 张晓伟, 刘三阳. 免比例因子 F 的差分进化算法[J]. 电子学报, 2009, 36(6): 1318 – 1323.

[11] HANSEN P, MLADENOVIC N. Variable neighborhood search: Principles and applications[J]. European Journal of Operational Research, 2001, 130(3): 449 – 467.

[12] 汪定伟, 王俊伟, 王洪峰, 等. 智能优化方法[M]. 北京: 高等教育出版社, 2007.

[13] CHAKRABORTY U K, DAS S, KONAR A. Differential evolution with local neighborhood [C]// CEC'06: Proceedings of IEEE Congress on Evolutionary Computation. New York: IEEE Press, 2006: 2042 – 2049.

[14] KINCAID D, CHENEY W. 数值分析[M]. 王国荣, 俞耀明, 徐兆亮, 译. 北京: 机械工业出版社, 2006.

[15] SEDGEWICK R, FLAJOLET P. An introduction to the analysis of algorithms[M]. Boston: Addison-Wesley Publishing Company Inc., 1999.