

doi:10.3969/j.issn.1672-4348.2014.06.017

# 多种网络业务并存的 IEDF 算法研究

沈俊慧

(福建船政交通职业学院 信息工程系, 福建 福州 350007)

**摘要:** 文章以 EDF 算法为基础,提出一种改进的调度算法 IEDF。在 IEDF 中,提出一种新的带宽调度策略,并在 3 种具体情况下进行业务流调度过程分析。同时,在 SIM 环境下进行仿真实验,仿真结果表明,和原 EDF 算法相比,IEDF 算法有效保证了带宽,降低了时延,提升了 Qos。

**关键词:** EDF 算法; IEDF 算法; 实时业务; 带宽保证; Qos

中图分类号: TP393.02

文献标志码: A

文章编号: 1672-4348(2014)06-0589-06

## IEDF algorithm based on a variety of network businesses

Sheng Junhui

(Information Engineering Department, Fujian Chuanzheng Communication Vocation College, Fuzhou 350007, China)

**Abstract:** An improved earliest deadline first (EDF) scheduling algorithm (IEDF) was proposed, which involves a new scheduling strategy to properly dispatch bandwidth when a variety of network business coexist. The scheduling process of network business bandwidths dispatching in three cases was analysed. Meanwhile, simulation experiments were conducted in SIM environment. The simulation results show that the IEDF algorithm ensures the bandwidth dispatching more effectively, lowers time delay and improves the Qos compared with the original EDF.

**Keywords:** EDF algorithm; IEDF algorithm; real-time business; bandwidth guarantee; Qos

目前,随着用户需求不断增长,计算机网络服务正在从单一数据传输逐步转向视频、语音、图像等多种网络服务发展,网络中的用户不再受时间、地点和网络类型的限制。计算机网络所提供的业务在不断增多,而用户对其所提供的服务质量(Qos)也提出了进一步的要求。如何能够在提供网络服务的同时确保服务质量,是计算机网络服务不断发展过程中关注的重点。

在 RFC238610 中,Qos 是指网络在进行传输时所需满足的用户服务请求,其性能指标主要包括带宽、延迟、数据传输量等<sup>[1]</sup>。分组调度算法是实现 Qos 等级控制的一种核心技术之一。分组调度算法依据一定原则对队列中数据传输顺序和带宽资源分配进行调度,以满足不同等级 Qos 要

求。在不同业务需求环境中,根据具体需求采用不同调度算法是十分重要的。EDF 算法是一种分组调度算法,属于动态调度算法<sup>[2]</sup>。当调度队列中加入新的任务时,EDF 就会依据任务截止时间对队列中任务优先级进行重新判定。EDF 算法是最优单处理器动态调度算法之一<sup>[3]</sup>。

本文以 EDF 算法为基础,提出一种基于网络中多类业务并存的改进 EDF 算法(IEDF 算法)。在 IEDF 算法中,提出基于多业务并存的带宽分配策略,并以 3 种情况为例,进行具体的业务流调度过程分析。最后,在 SIM 环境下,分两种流量模型,对 IEDF 算法和原 EDF 算法进行时延和带宽利用率的仿真测试,并进行对比分析。本文中所指的多类业务主要指当今网络业务中比较主流

收稿日期: 2014-09-27

作者简介: 沈俊慧(1977-),女(汉),福建诏安人,讲师,硕士,研究方向: 计算机网络技术与程序开发。

的实时业务、带宽保证和尽力服务。

1 EDF 算法研究

EDF 是一种采用动态调度的优先级调度算法,在 EDF 进行调度的过程中,其优先级是可变的。当当前队列中添加进新的任务时,优先级就可能会依据各自任务的截止时间,进行优先级重新设定。

下面通过具体实例来分析 EDF 算法的任务调度过程。

假如有如表 1 所示任务。

表 1 EDF 任务表

Tab.1 EDF tasks

任务名称	到达时刻/ ms	执行时间/ ms	绝对时间限制/ ms
A	0	15	35
B	3	4	15
C	6	10	30

EDF 算法的任务调度过程如下:

(1)初始时刻:此时任务列表中只有 A,此刻 A 是唯一等待执行的任务,因此 A 立即执行。

(2)时刻 3:B 到来,由上表可看出 B 的截止时间小于 A,根据 EDF 优先级判断原则,B 任务具有较高的优先级,那么 B 将打断 A 执行进行抢占执行。

(3)时刻 6:C 任务到来,而 C 的截止时间大于 B,相较于 B,C 具有较低优先级,那么 C 只能等到 B 执行完成才能继续执行。在 B 执行完成后,由于 C 截止时间小于 A,因此率先执行 C,C 执行完后,继而执行 A。

EDF 算法的基本假设:EDF 算法的分析是以一系列的假设为基础的,这些假设主要包含以下几个方面:

(1)所要处理的任务都是相互独立的,彼此之间不存在优先约束和优先限制。

(2)没有任务非抢先区域,并且抢先成本极小。

这些假设使 EDF 的分析得到极大简化。假设(1)表明不存在优先约束,也就是说任务被释放的时间和其他任务的完成时间没有关联。假设(2)则表明任务的抢占次数对处理器的影响很小。而任意任务可在任意时刻进行任意任务抢

占。对于不满足以上假设的系统,应采用优先和排斥约束解决问题。

EDF 的可调度性:假定所有的任务都具有周期性,并且每一个任务的时间限制就是它们的周期,那么对任务集进行 EDF 调度测试是非常简单的:若总的任务集合的利用率小于等于 1,那么 EDF 算法就可以在单一处理器上进行任务集的合理调度。对于那些时间限制并不完全等于周期的任务集,就需要 EDF 算法生成一个时间表,进行判断在约定时间内,所有任务的时间限能否被完全满足,在这种情况下,EDF 的可调度测试可先定义以下几个公式:

$$u = \sum_{i=1}^n (e_i/p_i)$$

$$d_{\max} = \max_{1 \leq i \leq n} \{d_i\}$$

$$T = \text{lcm}(T_1, \dots, T_n)$$

其中 lcm 表示最小公倍数。定义  $h_p(p)$  是任务集  $P$  中满足|时间限| <  $p$  的所有任务的执行时间和。而一个由具有  $m$  个任务的任务集不是可行的 EDF 调度的充要条件是:

$$u > 1 \text{ 或者存在一个任务的 } p < \min\{T + d_{\max}, \frac{u}{1-u} \max_{1 \leq i \leq n} \{T_i - d_i\}\}$$

$m$  为任务数量; $e_i$  为任务  $p_i$  的执行时间; $T_i$  为具有周期性任务的周期; $d_i$  为任务  $p_i$  的时间限; $h_p(p)$  为在绝对时间小于等于  $p$  的任务集合  $P$  中,所有重复任务的执行时间和。

由以上分析可得,EDF 在进行单处理器任务调度时具有很好的表现,但在多任务并存进行调度时存在一定不足。基于此,本文提出一种在多种网络业务并存的环境下能进行合理调度的改进 EDF 算法(IEDF 算法)。在 IEDF 算法中,提出基于多业务并存的带宽分配策略,并以 3 种情况为例,进行具体的业务流调度过程研究。

2 IEDF 带宽保证策略

网络资源总是有限的,在多种网络业务并存的环境中,为保证服务质量,资源分配的公平性尤为重要。从考虑公平性出发,在优先实现实时业务流时,应先估计资源是否充足。由于 IEDF 算法需要对多种网络业务进行共同保证应该考虑运行时间长短,带宽是否够用。基于此,本文提出一种基于时间粒度的带宽调度策略。

时间粒度是指在无限可分割的时间内的通信量。例如:若 20 s 内能传输 40 kb 数据,那么通信量可称为 40 kb/20 s,也可称为 20 kb/10 s。时间粒度相同是指在相同的时间单位内传输的数据量相同,与传统意义上的时间相同的概念不同。

在制定 IEDF 算法调度策略之前,假设实时业务流满足上述提到的 EDF 的可调度的条件。实时业务流具有周期性,  $T_i$  是每一个实时业务流的周期,  $M$  是所有实时业务流周期的最小公倍数,所有实时业务流的超周期:

$$M = \text{lcm}(T_1, T_2, \dots, T_i) (\text{lcm 最小公倍数}) \quad (4)$$

IEDF 带宽保证策略如下:在 IEDF 中,为保证所有实时数据在规定的时间内都能实现调度,则优先调度实时业务流,然而在这样的情况下,带宽业务流一般不能够得到全部满足。在带宽保证业务流不能被完全满足的情况下,只要带宽保证业务流在所有实时业务流的  $M$  超周期时间内能够达到相同时间粒度带宽,就认为带宽业务流得到了保证,即使带宽总量可能不够,这就是 IEDF 中的带宽调度策略。

### 3 IEDF 算法调度过程

首先将 IEDF 算法调度过程中所要用的公式列举如下:

$$T_{BE} = L_{BE}/V \quad (5)$$

$$P = \min \left\{ \left\lceil \log_2 \left( 1 + \frac{RTT}{L_{BAMAX}/V} \right) \right\rceil + 1, \left\lceil \log_2 \left( 1 + \frac{Q}{L_{BAMAX}/V} \right) \right\rceil - 1 \right\} \quad (6)$$

$$T_{BA} = \begin{cases} 2 \times \frac{C \times L_{BAMAX}}{BW \times \sqrt{D}} + \frac{Q}{V}, & \text{带宽保证流所需带宽} \leq BW \\ 2RTT + \frac{Q}{V} + P \left[ RTT + \frac{L_{BAMAX}}{V} \right] - (2^P - 1) \frac{L_{BAMAX}}{V}, & \text{其他} \end{cases} \quad (7)$$

在公式(5)中,  $L_{BE}$  表示尽力服务业务流单周期数据包的总长度,  $V$  表示传输速率,  $T_{BE}$  则表示传输数据包的时间。在公式(6)中,  $L_{BAMAX}$  表示带宽保证流数据包最大长度,  $Q$  表示单周期数据包最大长度,  $RTT$  表示往返数据包所需时间。在公式

(7)中,  $T_{BA}$  表示发送带宽保证业务流所需的时间,  $BW$  表示网络所提供的带宽。带宽业务流所需带宽与网络提供带宽存在 3 种关系(1)所需大于提供;(2)所需等于提供;(3)所需小于提供。

IEDF 的业务流调度过程如下所示:在多种网络业务并存,并且实时业务流符合 EDF 调度条件的情况下,根据公式(4)计算出实时业务流的超周期  $M$ ,根据公式(5)计算出尽力服务业务流的  $T_{BE}$ ,根据公式(6)、(7)计算出带宽保证流的传输时间,根据计算结果判断带宽调度策略。在这 3 类网络业务流中,实时业务流具有最高优先级调度,而尽力服务流调度优先级最低,因此,在进行调度策略判断时,只考虑实时业务流和带宽保证流的调度情况,而不考虑尽力服务业务流。

对带宽保证策略的选择,主要分以下 3 种情况进行调度过程分析,在以下 3 种情况中,  $x, y, z$  分别表示实时业务流、带宽保证流和尽力服务流的个数:

$$(1) x = n, y = 1, z = 1$$

在此情况下,有  $n$  个实时业务流,只有一个带宽保证流和尽力服务流。在实时流的超周期  $M$  内,所有实时业务流调度完后,若剩余时间大于  $T_{BA}$ ,即  $H - \left( \sum_{i=1}^n p_i + T_{BA} \right) \geq 0$ ,则表明在  $M$  时间内,带宽保证流所需负载量能够得到完全保证,此时,优先按照 EDF 进行实时业务流调度,继而进行带宽保证流调度,最后进行尽力服务流调度。

若  $H - \left( \sum_{i=1}^n p_i + T_{BA} \right) < 0$ ,则表明在超周期  $M$  期间,所有带宽业务流所需负载不能被完全保证,此时依据 IEDF 算法新的带宽保证策略,只要在  $M$  时间内达到相同时间粒度的带宽,那么就认为满足所有带宽业务流得到了保障。此时取  $\left\lceil \left( H - \sum_{i=1}^n p_i \right) / T_{BA} \right\rceil \times 10 \}$  / 10 倍为约定带宽,以实现尽最大服务进行带宽保证(这里取  $\left\lceil \left( H - \sum_{i=1}^n p_i \right) / T_{BA} \right\rceil \times 10 \}$  / 10 为约定带宽是经过多次实验测试得来的)。

$$(2) x = 1, y = n, z = n$$

在此情况下,实时业务流只有一个,因此只需优先调度实时业务流,而余下时间资源如何分配给带宽保证流和尽力服务流是调度的关键。在

IEDF 新的调度策略中,将带宽保证流看成实时业务流,这时存在优先级高低的问题,谁具备高优先级就先调度谁,这就需要根据约定带宽的大小判定优先级。在公式(6)中,带宽与往返时间成反比,因此,优先保证带宽保证业务流的带宽大小,能够使带宽保证流所用时间减小。情况 1 中,带宽保证所用时间与时间粒度成反比,即  $T_{BA}$  减小,时间粒度会增大,时间粒度增大后,其获得的带宽也就随之增加,这就使带宽和约定带宽比率增加,使带宽保证率得到有效提高。

和情况 1 的分析过程相似,在超周期  $M$  期间,所有实时业务流调度完后,若剩余时间大于所有  $T_{BA}$  的和,即  $H - (\sum_{i=1}^n p_i + T_{BA}) \geq 0$  则表明依据 IEDF 带宽调度策略,在  $M$  时间段内能够完全满足所有带宽保证业务流所需负载量,在此种情况下,优先调度实时业务流,然后依照最大约定带宽优先原则进行带宽保证流调度,最后进行尽力服务业务流调度。

若  $H - (\sum_{i=1}^n p_i + T_{BA}) < 0$ ,则表明在超周期  $M$  期间,所有带宽保证业务流所需负载量不能被完全满足。在此情况下,依据 IEDF 算法带宽保证策略,将所有带宽保证业务流依据最大约定带宽优先原则进行调度,依据此原则,能够满足所有带宽保证业务流带宽就进行调度。否则,根据  $\{[(H - \sum_{i=1}^n p_i) \times 10] / 10\}$  倍约定带宽进行带宽保证业务流调度,来尽最大能力满足带宽保证。

(3)  $x = y = z = n$

在此情况下,先进行 EDF 实时业务流调用,实时业务流调用完成后,再结合第一、第二种情况进行带宽保证流和尽力服务流调用分析。

若在所有实时业务流的超周期  $M$  时间内,调用完实时业务流调度后,剩余时间大于所有带宽保证业务流的  $T_{BA}$  的和,即  $H - (\sum_{i=1}^n p_i + T_{BA}) < 0$ ,则表明,在  $M$  时间段内,带宽保证业务流所需负载量能够完全满足,在此情况下,优先按照 EDF 进行所有实时业务流调用,然后依据约束带宽优先原则进行带宽保证流调用。

若  $H - (\sum_{i=1}^n p_i + T_{BA}) < 0$ ,则表明在  $M$  时间段中,所有带宽保证流所需负载量不能被完全满足,

在此情况下,依据 IEDF 算法带宽保证策略,只要在  $M$  时间段内达到相同时间粒度的带宽,就可认为满足所有带宽业务流得到保障,此时取最接近的比率倍为时间粒度,也就是取  $\{[(H - \sum_{i=1}^n p_i + T_{BA})] \times 10\} / 10$  为约定带宽,以实现尽最大服务进行带宽保证。

## 4 实验仿真

在实验仿真模块,采用 SIM 模拟器进行仿真测试。SIM 是由斯坦福大学研究开发的交换机模拟软件,由 C 语言编写,运行在 Linux 环境下。在 SIM 中能够提供多种流量模型,并能对不同技术实现的交换机进行功能模拟。

在模拟实验中,将改进的 IEDF 算法添加到 SIM 模拟器中。同时与原 EDF 算法进行对比分析。分别统计贝努利独立同分布和突发流量两种不同模型在相同负载下的时延(端对端、输入端)以及带宽业务流的带宽保证率(实际保证带宽/约定带宽)。

### 4.1 仿真环境搭建

(1)使用  $4 \times 4$  型交换机对同一种算法进行不同环境下的仿真模拟。可以通过 Minputs0 和 NumoutPuts0 参数进行交换机类型的指定。

(2)通过配置文件中的 Algorithm 参数来指定进行仿真模拟所要调度的算法。在本文的仿真中,采用 EDF 和 IEDF 两种算法来进行仿真模拟。

(3)采用两种流量模型进行仿真测试,分别是贝努利独立同分布和突发流量两种流量模型。模拟时间是 1 000 个时间槽,模拟中采用两个优先级,贝努利独立同分布模型下设置交换机负载为 94%,设置突发流量下交换机负载为 48%。流量模型的合适选择是进行精确流量预测的基础。

### 4.2 实验结果

在端口为  $4 \times 4$  交换机中进行 IEDF 和原 EDF 算法仿真测试,测试时延(端对端、输入端)以及带宽保证率。为确保仿真结果的可信性,进行多次测试,得到仿真结果如下所示:

两种流量模型下的端对端时延仿真结果如图 1 和 2 所示:

图 1 所示,在贝努利独立同分布模型下,当最高优先级业务占有率在 25% 以下时,其对传输延迟影响很小,随着占有率的不断增加,相较于原

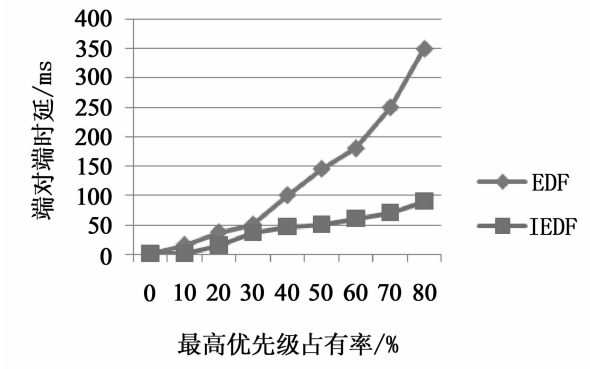


图 1 贝努利独立同分布端对端时延对比

Fig.1 Terminal to terminal time delay comparisons of Bernoulli independent iso-distribution

EDF 算法,端对端传输时延明显降低,平均降低 25.3%,最高时能降低 45.1%。图 2 所示,在流量突发模型下,在最高优先级占有率在 25% 以下时,原 EDF 算法对传输时延有一定影响,随着占有率的增加,IEDF 算法对传输时延几乎没有影响,反观 EDF 算法,随着占有率的增加,时延也逐渐增加,这是因为流量激增,出现大量最高优先级信元,EDF 算法会应用一定的算法对其进行特殊处理,这在一定程度上增加了时延。

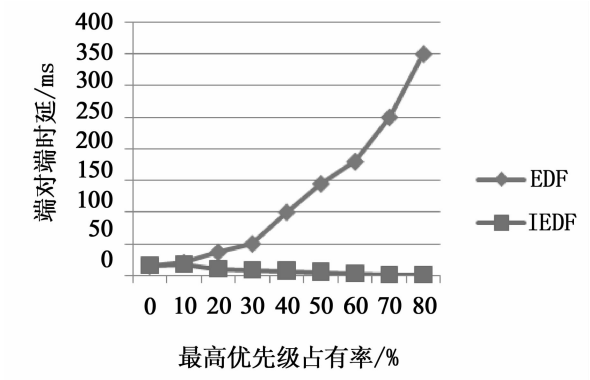


图 2 突发流量端对端时延对比

Fig.2 Time delay comparison of instant flowing bandwidth

两种流量模型下的传输时延仿真结果如图 3 和 4 所示:

如图 3 所示,在贝努利独立同分布模型下,当最高优先级业务占有率较低时,IEDF 算法的输入延迟较 EDF 算法低,能够实现接收到的数据立即进行传输。而随着占有率的增加,IEDF 的输入延迟逐渐加大,这是因为要对大量最高优先级信

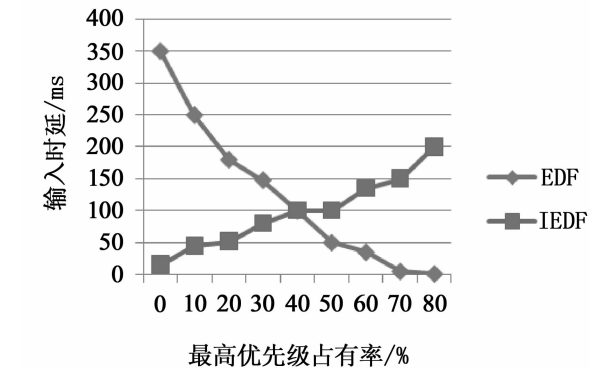


图 3 贝努利独立同分布输入时延对比

Fig.3 Input time delay comparison of Bernoulli independent iso-distribution

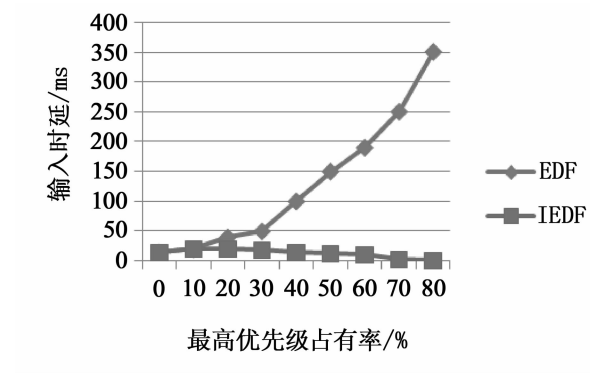


图 4 突发流量输入时延对比

Fig.4 Input time delay comparison of instant flowing bandwidth

元进行特殊处理。如图 4 所示,在流量突发模型下,IEDF 较之于 EDF 的输入时延有明显降低,随着最高占有率的不断增加,传输时延基本上贴近于 x 轴,这表明 IEDF 能够对接收到的数据实现无延迟的发送,即时收到,即时发送,相较于原 EDF 有很大改善。

从两种模型下的端对端时延和传输时延来看,在低负载情况下,IEDF 算法在时延上表现良好,在一定很大程度上降低了时延,甚至能够实现几乎零延迟的即时传输。在最高级占有率增加时,因为需要对其进行特殊处理而导致了时延增加,但相对于原有 EDF 算法,其对时延的控制也较佳。

带宽保证率等于网络所提供带宽/约定带宽。4×4 交换机下,两种算法的带宽保证率如表 2 所示。

表 2 4×4 交换机下带宽保证率

Tab.2 Bandwidth guaranteeing rate with 4×4 exchanger

约定带宽/b	IEDF/b	EDF/b
1	75	55
2	60	43
3	57	33

如表 2 所示,在相同的约定带宽下,IEDF 算法的带宽保证率较原 EDF 算法平均提高 19.7%,有效保证了带宽。

参考文献:

[1] 林闯,单志广,任丰原. 计算机网络的服务质量(QoS)[M]. 北京:清华大学出版社,2004;210-213.

[2] 王强,王宏安,金宏,等. 实时系统中的非定期任务调度算法综述[J]. 计算机研究与发展,2004,41(3):385-392.

[3] 郭艾寅,钱光明,徐云剑. 基于多种服务并存网络的实时队列时延研究[J]. 计算机工程与应用,2009,44(17):101-103.

(责任编辑:肖锡湘)

(上接第 576 页)

目标有遮挡的过程,其跟踪效果不佳,因此还可以考虑通过运动光流法对目标跟踪算法进行改进。

参考文献:

[1] Magee D. Tracking multiple vehicle using foreground, background and motion models[J]. Image and Vision Computation, 2004,22(2):143-155.

[2] Jia L H, J H Z. Improved background subtraction based on novel thresholding[J]. Acta Automatica Sinica,2009,35(4):467-473.

[3] Zhang K, Zhang L, Yang M H. Real-time compressive tracking[C]// Proceedings of the 12th European Computer Vision(III)ECCV 2012. Berlin:Springer,2012:864-877.

[4] Candes E J, Romberg J, Tao T. Robust uncertainty principles: Exact signal recognition from highly incomplete frequency information[J]. IEEE Trans on Info Theory, 2006,52(2):489-509.

[5] Donoho D L. Compressed sensing[J]. IEEE Trans on Info Theory, 2006,52(4):1289-1306.

[6] Candes E J, Tao T. Near optimal signal recovery from random projections: Universal encoding strategies[J]. IEEE Trans on Info Theory, 2006,52(12):5406-5425.

[7] 朱明,高文,郭立强. 压缩感知理论在图像处理领域的应用[J]. 中国光学,2011,4(5):441-447.

[8] Achlioptas D. Database-friendly random projections: Johnson-Lindenstrauss with binary coins[J]. Journal of Computer and System Sciences,2003,66(4):671-687.

[9] Baraniuk R, Davenport M, Devore R, et al. A simple proof of the restricted isometry property for random matrices[J]. Constructive Approximation,2008,28(3):253-263.

[10] Li P, Hastie T J, Church K W. Very sparse random projections[C]// Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2006;287-296.

[11] Babenko B, Yang M H, Belongie S. Robust object tracking with online multiple instance learning[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence,2011,33(8):1619-1632.

[12] Diaconis P, Freedman D. Asymptotics of graphical projection pursuit[J]. The Annals of Statistics,1984,12(3):793-815.

(责任编辑:肖锡湘)