高技术通讯 2010 年 第 20 卷 第 1 期:68 ~ 74

doi:10.3772/j.issn.1002-0470.2010.01.012

使用支持向量机的微处理器验证向量优化方法①

王朋宇② 郭 崎 沈海华 陈云霁 张 珩

(中国科学院计算技术研究所计算机系统结构重点实验室 北京 100190)
(中国科学院研究生院 北京 100039)

摘 要 为了解决微处理器仿真验证中随机验证向量质量不高的问题,提出了一种基于支持向量机(SVM)的验证向量优化方法。该方法将已仿真运行的验证向量及其覆盖率信息作为支持向量机的样本进行有监督学习,得到验证向量关于功能覆盖点的分类器。利用训练后的分类器对于新产生的验证向量进行预测,并丢弃预测中不能提高覆盖率的冗余验证向量。实验数据表明该方法能准确地过滤冗余验证向量,提高仿真运行的验证向量的质量。和完全随机的验证向量生成方法相比,该方法达到相同的功能覆盖率仅需要前者 1/3 的验证向量。

关键词 支持向量机(SVM),功能覆盖率模型,微处理器验证,仿真验证,验证向量优化

0 引 言

功能验证已经成为微处理器设计中最大的瓶颈。仿真验证作为最主要的功能验证方法,通过输入验证向量和检查输出结果来完成对设计的验证。不具有完备性,而仅能通过高质量的验证向量来尽量提高验证效率。在微处理器的验证中,验证向量通常是按照一定约束条件随机产生。整个验证过程中 3/4 以上的设计错误是通过运行随机验证向量发现的[1]。覆盖率驱动的验证方法根据随机验证向量和功能点覆盖比例(覆盖率)之间的关系来调整验证向量的生成策略,提高生成的验证向量的质量和验证的效率。覆盖率驱动的验证方法主要分为两类:基于反馈的方法和基于构建的方法。基于反馈的方法动态分析反馈覆盖率信息,根据分析结果自动调整验证向量的生成。这类方法的一个代表是基于贝叶斯网络的方法。该方法在覆盖率和验证向量生成指导之间建立贝叶斯网络,动态调整验证向量生成指导,生成更高质量的验证向量[2]。基于反馈的方法还包括利用基因算法和推理逻辑编程等[3,4]。这类方法需要针对特定的设计建立复杂的模型,学习过程需要人工参与,学习过程比较长。基于构建的方法静态分析被验证设计(design under verification, DUV),从 DUV 中抽取抽象级更高的模型如有限状态机模型,然后从高级模型生成覆盖任务,根据覆盖任务产生验证向量[5]。这类方法存在状态空间爆炸的问题,因此仅适用于模块级的验证。Romero 等人用过滤的方法来提高验证效率[6],他们基于参考模型的仿真速度比实际寄存器传输级(register transfer level,RTL)设计快的条件,用参考模型对验证向量过滤。但实际上参考模型跟实际的 RTL 设计差别比较大,参考模型的覆盖率模型跟实际的 RTL 设计很难对应到相同的验证向量。本文研究一种基于支持向量机(support vector machine,SVM)和功能覆盖率的验证向量优化方法。该方法把功能覆盖率模型中的覆盖点作为 SVM 的分类器标准,将训练验证向量及其覆盖率信息输入到支持向量机中,通过学习得到功能覆盖点的分类器,对于新的验证向量用学习得到的分类器进行过滤。如果分类器的预测结果显示某验证向量能覆盖到的功能点对覆盖率的提高没有帮助,则将其判定为冗余验证向量不予仿真运行。该方法能有效地提高验证向量的质量,缩小实际仿真运行的验证向量的规模。

1 功能覆盖率模型

覆盖率作为衡量验证充分程度的标准,一般分

① 国家自然科学基金(60603049,60673146),863 计划(2007AA01Z112,2008AA110901),973 计划(2005CB321600)和北京市自然科学基金(4072024)资助项目。
② 男,1979 年生,博士;研究方向:处理器设计与验证;联系人,E-mail: wangpengyu@ict.ac.cn
(收稿日期:2009-01-16)

为代码覆盖率和功能覆盖率两种。基于程序的代码覆盖率包括行覆盖率、分支覆盖率、条件覆盖率等,它是检测设计代码本身被执行的程度,100%的代码覆盖率并不能保证设计功能的正确。功能覆盖率是基于设计和实现的,它侧重于设计的功能,通常用来检测设计功能的所有重要的方面。功能覆盖率模型只能由验证工程师根据具体的设计或者设计规范手工定义。交叉矢量(Cross-product,CP)功能覆盖率模型[7]是一种基本功能覆盖率模型,它由三部分组成:一个对被覆盖对象的语义描述;该语义描述中的所有属性列表;属性的所有可能取值。利用交叉矢量功能覆盖率模型可以给出另外的功能覆盖率模型的表示方法,包括用矩阵表示、用树状图表示、用混合模型表示等[8]。用矩阵、树状图和混合模型可以灵活定义各种功能覆盖率模型。对于属性比较少且属性的值是连续的可以用矩阵来定义功能覆盖率模型;对于属性较多且属性的取值是离散的可以用树状图定义功能覆盖率模型;而混合模型可以定义更复杂的功能覆盖率模型。图 1 是一个树状图表示的功能覆盖率模型实例。该功能覆盖率模型描述了指令的操作数来源。属性 A 为指令,其值有单操作数指令 A0、双操作数指令 A1 和三操作数指令 A2。属性 B 为操作数,其值分为第一个操作数 B0、第二个操作数 B1 和第三个操作数 B2。属性 C 为操作数来源,其值有来自保留站 C0、来自寄存器 C1。

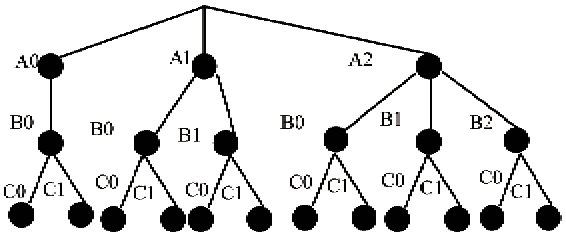

图 1　树状图功能覆盖率模型

该功能模型中的功能点为属性三元组 <A,B,C>,模型一共定义了 12 个功能点。

功能覆盖率是指验证中被充分验证到的功能点与功能覆盖率模型中总的功能点的比率。功能覆盖率作为衡量验证程度的标准表现为:如果某个功能点被覆盖到(一定的次数),则说明该功能点已经被充分的验证;如果功能覆盖率达到预定的要求,则说明对设计的验证已经很充分。

随机验证向量具有自动化生成和易于覆盖到“边角”情况的优点,是功能仿真验证中最主要的验证向量。但它有一个很大的特点就是存在冗余:大量随机产生的验证向量会造成验证功能点的重复覆盖,验证向量的无谓重复只会延长功能验证的收敛时间,降低验证效率。在基于覆盖率的仿真验证中,可以自然地将验证向量分为两类:覆盖到某一个功能点的验证向量和没有覆盖到该功能点的验证向量。通过分类,我们可以将对覆盖率增长没有帮助的验证向量过滤掉,能够提高随机验证向量的质量。

2　支持向量机理论基础

支持向量机(SVM)是根据 Vapnic 的统计学习理论[9]产生的一种模式分类方法,它在字符识别[10]、人脸识别[11]、文本识别[12]等模式识别领域有着大量的应用。它引入了结构风险最小化原则和核映射的思想,不仅克服了传统方法的大样本要求,而且有效克服了维数灾难及局部极小问题。本文使用 SVM 技术研究微处理器验证中随机验证向量的优化问题,采用该技术作为解决方案,是因为 SVM 是一种有效的模式分类方法。

SVM 是从线性可分情况下对分类超平面的最优化发展而来的,它主要包括两类问题:线性可分问题以及线性不可分问题。对于线性可分问题,考虑样本集 $(x_i, y_i), i = 1,2,\cdots,n, x_i \in R^n$ 为 n 维向量,$y_i \in \{+1, -1\}$ 用来标识该模式向量的类别。在样本线性可分的情况下,目的是找到一个最优超平面将两个类别的点分离开。最优分类器就是要求分类器不但能将两类正确分开,而且使得分类间隔最大。这个最优超平面被定义为

$$(w \cdot x) + b = 0 \quad (x \in R^n, w \in R^n, b \in R) \tag{1}$$

要找到这个超平面,需要求解下面二次规划问题:

$$\min \frac{1}{2} \| w \|^2 \tag{2}$$

$$s.t.\ y_i((w \cdot x_i) + b) \geqslant 1 \tag{3}$$

该约束问题可以用 Lagrange 方法求解,解上述问题后得到的最优分类函数是

$$f(x) = \mathrm{sgn}\{\sum_{i=1}^{n} a_i^* y_i < x_i \cdot x > + b^*\} \tag{4}$$

大多数样本集在原始空间内都是线性不可分的。这类问题的一般解决方法是采用非线性映射,将原始空间的样本映射到高维特征空间,使样本在高维空间内线性可分。假设 $x \in R^n$ 经非线性函数 $\phi(x)$ 变换后,得到 $\phi(x_i) \in R^m, m > n$。称 x_i 所属

的 n 维空间为输入空间，$\phi(x_i)$ 所属的 m 维空间为特征空间。此时的内积运算为 $<\phi(x_i)\cdot\phi(x_j)>$，是在高维空间中进行的，可能会遭遇维数灾难问题，使计算成为不可能。核函数的提出成功地解决了这一问题。如果能找到一个函数 $K(x_i\cdot x_j)$ 满足 $K(x_i\cdot x_j)=<\phi(x_i)\cdot\phi(x_j)>$，则高维空间的内积运算就可以用在原空间中的函数来实现，从而绕过维数灾难问题，使样本集在特征空间可分。引入核函数后，分类函数变为

$$f(x)=\mathrm{sgn}\{\sum_{i=1}^{n}a_i^*y_iK(x_i\cdot x_j)+b^*\} \qquad (5)$$

3 SVM 分类器的学习获取

根据覆盖率信息对验证向量进行分类可以表达为多个二类问题：验证向量或者覆盖到某个功能点，或者没有覆盖到。标准的 SVM 分类器是二值分类问题，因此可以将标准的 SVM 分类器用于验证向量的在单个功能点上的分类预测。

如何得到精确的分类器，是验证向量过滤优化的关键。基于核函数的 SVM 机器学习方法具有出色的学习性能，能够在较少样本的情况下训练出具有相对较高性能的分类器。用 SVM 学习得到分类器的流程如图 2 所示。首先对训练样本进行特征提取，得到训练样本的特征向量，然后将特征向量输入到 SVM 学习器中，并选择合适的核函数进行学习，得到分类器。训练样本不仅包含训练验证向量，还包含验证向量在仿真后得到的功能点覆盖信息。

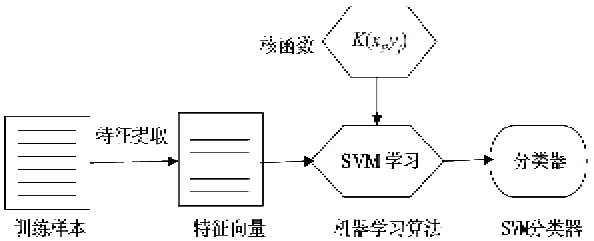

图 2 SVM 学习得到分类器流程

如果一个验证向量 s_i 覆盖到了 $t_1,t_2,\cdots,t_m$ 共 m 个功能点，对于每一个功能点 t_i 都与验证向量 s_i 构成了一个训练样本 (s_i,y_i)，$y_i\in\{+1,-1\}$ 为标签，表示功能点 t_i 是否被验证向量 s_i 覆盖到。图 2 中分类器的学习是针对某一个具体的功能点 t_i，假定训练样本中包含了 $s_1,s_2,\cdots,s_n$ 共 n 个训练验证向量，对每个训练验证向量 s_i 进行特征提取，得到该训练验证向量的特征向量 x_i，与训练验证向量对于功能点 t_i 的覆盖信息 y_i 构成 SVM 的输入数据结构 (x_i,y_i)，输入到 SVM 学习器中，学习得到该功能点的分类器。对功能点逐一进行学习，得到每个功能点的分类器，对于有 m 个功能点的情况，我们共需要建立 m 个 SVM 分类器。一个验证向量可能会覆盖到多个功能点，但如果分类器的预测结果显示它能覆盖的任意一个功能点都是已经充分验证过的功能点，则该验证向量被判定为冗余验证向量。

3.1 训练验证向量特征提取

在用 SVM 对训练样本进行学习得到分类器之前，必须对训练样本进行特征提取，减少 SVM 中输入的向量维数，提高分类效率。

微处理器的验证向量一般是指令序列，指令序列验证向量可以有多种表示方式，如汇编代码、机器码等。表 1 为 MIPS 微处理器一个验证向量片段的汇编代码和十六进制的机器码表示。实际验证中的指令序列长度一般为几千条。

表 1 验证向量的汇编码和十六进制码表示

	汇编码表示	二进制码表示
mfhi	$ 10	10 50 00 00
lb	$31, 24251($2)	bb 5e 5f 80
sub	$5, $13, $11	22 28 ab 01
addu	$19, $20, $24	21 98 98 02
swl	$10, -15728($2)	90 c2 49 a8

3.1.1 基于 N-gram 的预处理

N-gram 是文本信息统计的重要建模工具，它将文本看作字符串，按照窗口宽度为 n，步长为 1 的滑动窗口从该字符串中提取子串。它的基本思想是假定一个词的出现概率只与和它前面紧邻的 $n-1$ 个词有关。N-gram 模型被称为一阶马尔科夫链，它的时间复杂度随着窗口的宽度 n 增大呈指数增长，因此模型中 n 的值不能太大。

对于表 1 中指令助记符文本，对应的一个 2-gram 为：mfhilb，lbsub，subaddu，adduswl。在实际中，滑动窗口的大小依赖具体的应用。本文将 n-gram 分析应用于汇编码表示的验证向量中指令助记符表示的文本，滑动窗口的宽度定为 2 个指令助记符的长度。对训练样本中所有训练验证向量的指令助记符进行预处理，可以获得相应的频率矩阵。表 2 为一个频率矩阵实例。

表 2　特征向量频率矩阵实例

特征	频率	
	正例样本数	负例样本数
addsub	140	114
addsw	230	123
subadd	113	211
lbsub	121	152
lbsw	89	78

上述频率矩阵中特征的正例样本数和负例样本数是针对每个功能点而言的。正例样本数表明该特征在覆盖到这个功能点的训练验证向量中出现的次数,负例样本数表明该特征在没有覆盖到这个功能点的训练验证向量中出现的次数。

3.1.2　基于信息增益的特征选择

一个文本中包含的 n-gram 信息非常丰富,但并不是所有出现在文本中的 n-gram 信息项对分类都有益。信息增益(information gain,IG)是一种有效的特征选择方法,它反映了一个特征在区分不同类别时的重要性。通过计算信息增益并把具有较低增益值的特征忽略掉,是一种有效的降低特征空间维数的方法。信息增益是从信息论中信息熵得到的,根据信息论,变量 X 的信息熵为

$$H(X) = -\sum_{i=1}^{n} P(x_i) \cdot \log_2(P(x_i)) \tag{6}$$

变量 X 关于变量 Y 的条件信息熵是观测信息 Y 后,信息空间 X 的不确定性。信息增益是信息熵的差,定义为

$$IG(X,Y) = H(X) - H(X \mid Y) \tag{7}$$

对应到我们的实际问题,对于验证向量集 S,针对某个功能点,覆盖到该功能点的样本数为 p,未覆盖到该功能点的样本数为 n,总的样本数 $t = p + n$,则该特征 f 在样本集 S 中的信息增益表示为

$$IG(S,f) = H(S) - \sum_{v \in values(f)} \frac{p_v + n_v}{p + n} H(I_v) \tag{8}$$

由于这里是二分类问题,S 的取值只有正样例和负样例两种情况,分别对应覆盖到某个功能点的样例和未覆盖到某个功能点的样例。此外,特征 f 只有两种取值,$v \in values(f) = \{0,1\}$,分别表示属性 f 出现在某个样例中和不在某个样例中出现的情况。由此得到各个符号具体含义如下:

p_0:不包含特征 f 的正样例;

n_0:不包含特征 f 的负样例;

I_0:不包含特征 f 的所有样例,即 $p_0 + n_0$;

p_1:包含特征 f 的正样例;

n_1:包含特征 f 的负样例;

I_1:包含特征 f 的所有样例,即 $p_1 + n_1$。

根据特征项的信息增益,我们把具有较低信息增益的特征项剔除,然后将训练验证向量映射到剩余的特征项上,得到维数适当同时又能很好反映训练验证向量与覆盖率关系的特征向量。特征向量提取算法如图 3 所示。

算法:ExtractCV
输入:验证向量及覆盖率信息和特征向量的维数 n
输出:特征向量 S
1) 将用汇编码表示的验证向量文本文件中的指令助记符提取出来,保存到一个数组中;
2) 用 N-gram 提取特征,得到特征的频率矩阵;
3) 用信息增益算出频率矩阵中各个特征的信息增益值;
4) 根据输入的特征向量的维数,将信息增益值最大的前 n 个特征选取出;
5) 依次将每个验证向量即指令序列映射到选取出的 n 个特征上,得到一个 n 维的特征向量

图 3　验证向量特征提取算法描述

3.2　核函数选择

核函数是 SVM 算法中一个重要的参数,不同的核函数有不同的分类效果。目前常见的核函数有多项式核函数、径向基函数、Sigmoid 函数等:

多项式核函数:$K(x,y) = [(x \cdot y) + 1]^d$

径向基函数:$K(x,y) = \exp\left\{\frac{|x - y|^2}{2\sigma^2}\right\}$

Sigmoid 函数:$K(x,y) = \tanh(kx \cdot y - \delta)$

核函数的形式及其参数确定了分类器的类型和复杂程度,它也是分类器性能控制的手段。多项式核函数是一个应用比较广泛的非线性映射函数,其对目标函数的逼近能力是由多项式的阶数决定的,在实际应用中 d 值必须控制在一定范围内。径向基函数具有很好的学习性能,但它的性能优劣直接受参数 σ 大小的影响。虽然和神经网络联系在一起的 Sigmoid 函数是一个很好的逼近函数,但它只是部分满足支持向量机的核函数必须满足的 Mercer 条件,Sigmoid 函数没有特别的优势并且参数选择比较困难。目前关于核函数选择的理论依据不多,在我们的试验中采用了多项式和径向基两种核函数,并根据实验结果进行比较。

最后,将提取的特征向量按 Libsvm SVM 软件[13]要求的数据文件格式输入到 SVM 学习器中,选择适当的核函数,学习得到验证向量的覆盖率信息中相对应于每个功能点分类器。

4 基于 SVM 和覆盖率的验证向量优化

在传统的仿真验证中,新产生的验证向量不经过任何处理,就直接输入到被验证设计(DUV)中进行仿真验证。在基于支持向量机和功能覆盖率的验证向量优化中,验证向量在输入到 DUV 之前会经过两步处理:第一步对验证向量做覆盖率结果预测;第二步对验证向量进行冗余检测和过滤。工作流程如图 4 所示。

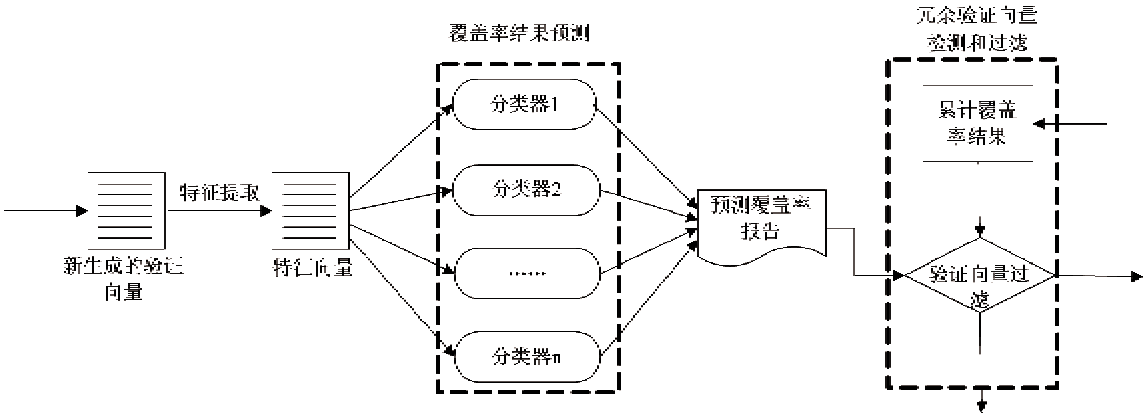

图 4 基于支持向量机和覆盖率的验证向量过滤

本文的验证向量生成采用约束求解的随机方法。由于本文主要研究冗余验证向量的检测和过滤,我们用参考文献[14]中的方法产生分布均匀的指令级随机验证向量。

4.1 验证向量覆盖率结果预测

在验证向量被仿真之前预测到它的覆盖率结果是一件非常困难的事情。我们用第 3 节所述的用 SVM 学习得到的分类器对验证向量进行覆盖率结果预测。在新的验证向量生成后,首先对其进行特征向量提取得到新验证向量的特征向量,然后将特征向量输入到分类器中进行覆盖率结果预测。假设共有 n 个功能覆盖点 $\{t_1, t_2, \cdots, t_n\}$,每个功能覆盖点 t_i 对应一个分类器 m_i,式 m_i 对输入的特征向量进行判断,判断其能否覆盖到功能点 t_i。将验证向量在所有的分类器上判断,得到完整的覆盖率结果预测信息,将这些信息收集组合起来作为下一步的输入。

4.2 验证向量冗余检测和过滤

对每一个功能点来说,其覆盖率体现在这个功能点是否被覆盖到、被覆盖了多少次。总的覆盖率体现在被充分验证的功能点占总功能点的比例。这样的统计信息可以由仿真工具在仿真过程中累计给出。对于每一个功能点,验证工程师给出这个功能点被充分验证的标准:该功能点被覆盖到(多少次)。当验证向量经过覆盖率预测后,覆盖率预测结果与仿真工具给出的功能点覆盖率统计信息比较,就会知道该验证向量能否提高覆盖率。如果验证向量能覆盖到的功能点的验证程度达不到验证工程师的预定要求,则验证向量能提高覆盖率,否则验证向量为冗余的验证向量。冗余的验证向量在这个步骤中被发现并且过滤掉。

验证向量的特征提取只取决于验证向量本身和其覆盖率信息,与设计规模没有关系。当设计规模大、设计复杂时,设计的功能点就会增多,基于 SVM 的分类器数量将会增加。为了降低训练的计算代价,我们可以并行地训练这些相互独立的分类器。

5 实验结果

我们在 AMD 平台上实现了基于 SVM 和覆盖率的验证向量过滤工作流程。实验平台的配置为 AMD 速龙 64 3200 + 处理器,64GB 内存,操作系统为 Linux,模拟器为 Synopsys 公司的 VCS7.0。本文中对龙芯一号 Soc 的处理器核进行验证,该处理器核是龙芯一号处理器[15]核的增强版本。覆盖率模型的具体实现采用 System Verilog 语言,一共定义了 10000 多个功能点。功能覆盖点大多数具有关联性,我们对功能覆盖点进行分类,选取了其中的 1000 个功能点做实验。

图 5 是两种核函数在不同特征向量维数时分类

正确率的平均值表示。从图中可以看出,径向基核函数的分类准确度略好于多项式核函数,但总体上两种核函数的分类方法效果差距不大,特征项维数才是影响分类正确性的主要因素。直观上特征项的数量越多,模型的精确度就越高。但从图 5 中反映的结果可以看到,用信息增益对特征进行过滤,在特征向量达到一定的维数前(图中为 800 维),支持向量机分类器的正确性逐渐提高。但当特征向量达到一定的维数后,SVM 给出的结果开始变差。这是因为随着特征向量维数的增加,学习算法的复杂度也会增加,学习的时间显著增加。过高的特征向量维数会带来很多噪声,对学习算法的能力产生负面的影响,降低分类算法的分类准确度。

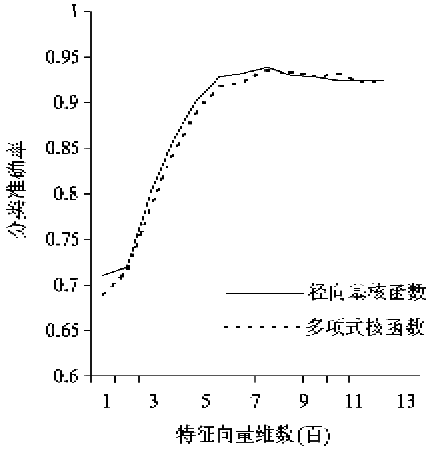

图 5　特征向量维数对分类正确率的影响

验证向量优化的目的就是过滤掉冗余的验证向量,提高验证向量的质量和验证的效率。图 6 给出了用 SVM 优化验证向量的效果图。从图 6 中可以看到,当覆盖率达到一定程度时,随机产生的验证向量很难再快速地提高覆盖率。而用优化过的验证向量则可以使覆盖率快速地增长,达到与原始随机验证向量相同的覆盖率,验证向量的规模大概缩减为原来的 1/3。当覆盖率达到一定程度时,用优化的验证向量覆盖率的增长(图中虚线的斜率)没有刚开始快,一是因为开始阶段的功能点是比较容易覆盖到,二是因为新的验证向量可能也会包含冗余,因为训练验证向量不可能覆盖到所有的功能覆盖点,因此不能对所有的功能点都建立一个分类器,对那些没有建立分类器的功能点,验证向量的冗余是无法检测到的。因此该方法对训练验证向量的有一定的要求,要求训练验证向量对功能覆盖点有一定的覆盖度。

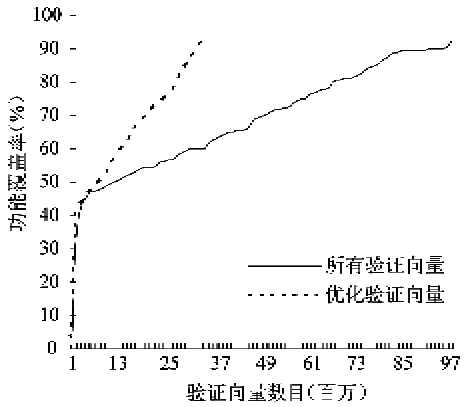

图 6　原始验证向量与优化验证向量效果图

6　结　论

为了解决仿真验证中验证向量质量不高的问题,本文提出一种基于支持向量机(SVM)的冗余验证向量过滤技术。该方法把功能覆盖点作为 SVM 的分类器标准。通过 SVM 对训练验证向量及其覆盖率进行学习,得到分类器。对于新的验证向量用学习得到的分类器进行过滤,如果新的验证向量能覆盖到的功能点对覆盖率的提高没有帮助,则为需要过滤掉的冗余验证向量。该方法能有效提高验证向量的质量,缩小验证向量的规模。龙芯一号 SOC 验证的实验结果表明,使用该方法有效地提高验证向量的质量,缩小了仿真运行的验证向量的规模。

未来还有进一步的工作需要完成。首先,因为训练验证向量只能覆盖到覆盖率模型中的一部分功能点,如果只是用训练验证向量学习到的结果进行分类,只能过滤掉与训练验证向量覆盖率相同的验证向量。新验证向量还是会有冗余问题。因此需要一种在线学习机制,将新增的对覆盖率有贡献的验证向量添加到训练验证向量中,重新用 SVM 学习动态生成新的分类器。其次,验证向量的特征提取没有考虑设计的具体实现,只是简单地从向量和覆盖率结果得到。在以后的工作中需要把领域知识加入到特征提取中,使提取的特征能更好地反映覆盖率信息,增加分类的正确性。

参考文献

[1] Taylor S, Quinn M, Brown D, et al. Functional verification of a multiple-issue, out-of-order, superscalar Alpha processor-the DEC Alpha 21264 microprocessor. In: Proceedings of the 35th Design Automation Conference, California, USA, 1998. 638-643

[2] Fine S, Zlv A. Coverage directed test generation for function-

al verification using Bayesian networks. In: Proceedings of the 40th Design Automation Conference, Anahemi, California, USA, 2003. 286-291

[3] Hsiou-Wen H, Eder K. Test directive generation for functional coverage closure using inductive logic programming. In: Proceedings of the 11th High-Level Design Validation and Test Workshop, Monterey, California, USA, 2006. 11-18

[4] Samarah A, Habibi A, Tahar S, et al. Automated coverage directed test generation using a cell-based genetic algorithm. In: Proceedings of the 11th High-Level Design Validation and Test Workshop, Monterey, California, USA, 2006. 19-26

[5] Ur S, Yadin Y. Micro-architecture coverage directed generation of test programs. In: Proceedings of the 36th Design Automation Conference, New Orleans, Louisiana, USA, 1999. 175-180

[6] Romero E L, Strum M, Wang J C. Comparing two testbench methods for hierarchical functional verification of a bluetooth baseband adaptor. In: Proceedings of the 3rd Hardware/Softerware Codesign and System Synthesis, Jesey City, New Jersey, USA, 2005. 327-332

[7] Grinwald R, Harel E, Gore A, et al. User defined coverage-a tool supported methodology for design verification. In: Proceedings of the 35th Design Automation Conference, California, USA, 1998. 158-165

[8] Piziali A. Functional Verification Coverage Measurement and Analysis. Boston: Kluwer Acad, 2004. 56-94

[9] Vapnik V. The Nature of Statistical Learning Theory. New York: Springer, 1995. 91-115

[10] Ahmad A R, Khalid M, Yusof R. Kernel methods and support vector machines for handwriting recognition. In: Proceedings of the Student Conference on Research and Development, Shah Alam, Selangor, Malaysia, 2002. 309-312

[11] Buciu F, Kotropoulos C, Pitas I. Combining support vector machines for accurate face detection. In: Proceedings of International Conference on Image Processing 2001, Thessaloniki, Greece, 2001. 1054-1057

[12] Joachims Thorsten. Text categorization with support vector machines: Learning with many relevant features. In: Proceedings of the 10th European Conference on Machine Learning, Dorint-Parkhotel, Chemnitz, Germany, 1998. 137-142

[13] Chang C C, Lin C J. LIBSVM : a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm: National Taiwan University, 2001

[14] 李潮基. 约束求解与随机测试程序生成研究:[硕士学位论文]. 北京: 中国科学院计算技术研究所, 2006. 21-30

[15] 胡伟武,唐志敏. 龙芯1号处理器结构设计. 计算机学报,2003,26(4): 385-394

An approach to microprocessor simulation vector optimization using SVM

Wang Pengyu, Guo Qi, Shen Haihua, Chen YunJi, Zhang Heng

(Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

(Graduate School of Chinese Academy of Sciences, Beijing 100039)

Abstract

This paper proposes a simulation vector filter method based on support vector machines (SVM) and functional coverage to cope with the problem that most random simulation vectors are redundant in simulation-based microprocessor verification. A SVM is used to learn the trained simulation vectors and their coverage information, and the new generated simulation vectors are filtered by the learned result. Those which can not improve the functional coverage are redundant and should be discarded. The experimental results based on application of the proposed methodology demonstrate that this technique can precisely filter the redundant simulation vectors to improve the efficiency of verification. Compared with the totally random method, only one third of the simulation vectors are needed to be simulated to reach the same functional coverage.

Key words: support vector machine (SVM), functional coverage model, microprocessor verification, simulation verification, simulation vector optimization