

# 不确定性与自然计算并行性的内在关联

王 鹏<sup>1,2</sup> 李建平<sup>2</sup>

(成都信息工程学院并行计算实验室 成都 610225)<sup>1</sup>

(电子科技大学计算机科学与工程学院 成都 610054)<sup>2</sup>

**摘 要** 不确定性是量子力学中已获得证明的自然界基本规律。通过分析自然算法中的并行性与不确定性的内在关联认为,不确定性不但是并行性产生的根源,还是算法智能产生的根源。根据这一认识给出了算法的不确定性原理,并结合 Shannon 的信息熵原理建立了自然计算的不确定性智能模型(UIM)。这一模型认为先验知识信息和由不确定性所提供的信息共同构成了智能系统的基本模型。智能系统本质上就是一个信息系统,先验知识信息保证算法按正确的方向进行搜索,不确定性所提供的信息实现了算法对解空间的并行搜索,通过提高系统信息的含量将有效提高系统的智能水平。这一模型在 Pi 值的计算算法中也得到了印证。

**关键词** 不确定性,自然计算,并行,智能,信息熵,不确定性智能模型

## Relationship between Uncertainty and Parallelism of Natural Computation

WANG Peng<sup>1,2</sup> LI Jian-ping<sup>2</sup>

(Parallel Computing Laboratory, Chengdu Univ. of Info. Tech., Chengdu 610225, China)<sup>1</sup>

(School of Computer Science and Engineering, Univ. of Electron. Sci. & Tech. of China, Chengdu 610054, China)<sup>2</sup>

**Abstract** Uncertainty principle of quantum mechanics is the basic law of nature. Uncertainty is considered as the root of parallelism and intelligence of natural computation by analyzing the relationship between parallelism and uncertainty. According to this understanding the uncertainty principle of algorithm was proposed, and the uncertainty intelligence model (UIM) of natural computation was established. Shannon entropy principle was used in this model. UIM indicates that prior knowledge and uncertainty information constitute the basic intelligent system. Intelligent system is essentially an information system. The information provided by prior knowledge assures the algorithm in the correct direction of search, and the information provided by uncertainty realizes the parallel search in solution space. So we can conclude that improving the entropy will achieve the high level intelligence. UIM is also verified by Monte-Carlo algorithm of Pi in this paper.

**Keywords** Uncertainty, Natural computation, Parallel, Intelligence, Entropy, Uncertainty intelligence model

量子力学描述了自然界最本质的运动规律。量子力学为世界引入了不确定性,并证明了不确定性是自然界的一种本质属性。自从 Heisenberg 在 1927 年提出不确定性原理(Uncertainty Principle)开始, Laplace 的决定论思想就被打破<sup>[1]</sup>。Shannon 将不确定性引入信息的度量<sup>[2]</sup>,奠定了信息论的基础。虽然到目前为止还有部分人仍然不相信“上帝是在掷骰子”,包括伟大的物理学家 Einstein,但大自然还是不断地向人们昭示它的不确定性所产生的巨大威力,甚至从某种程度上讲智能的产生亦来源于自然界的不确定性。自然界强大的信息处理能力也来自不确定性。

人类计算的发展就是不断地探索自然、学习自然的过程。自然计算就是指人工智能研究领域的一类通过模仿自然界的特点,具有自适应、自组织、自学习能力的模型与算法,它们能够解决传统计算方法难以解决的各种复杂问题。而且自然计算中的算法,比如遗传算法、模拟退火算法和人工神经网络算法<sup>[3]</sup>,通常都具有较好的并行能力。研究人工智能中的不确

定性问题,是计算机科学中一项重要而前沿的工作。李德毅<sup>[4-6]</sup>等人对不确定性人工智能进行了较为详细的研究,指出随机性和模糊性是不确定性的基本内涵,并认为不确定性人工智能将来会使电脑具备人脑所具有的不确定知识表示能力和处理能力。

本文以自然计算中不确定性所导致的算法内在并行性入手,研究了不确定性与智能算法内在并行性的关联,给出了算法的不确定性原理,建立了算法的不确定性智能模型  $I(f, U)$ 。此模型认为不确定性在智能算法中扮演着重要的角色,决定着算法的许多基本特性,为人工智能算法中的许多特性做出合理的解释。

## 1 自然界的不确定性原理

### 1.1 量子测量的不确定性原理

自然界的不确定性原理在量子力学中得到了全面的证明。量子力学中的不确定性原理描述了一个处于不确定状态

到稿日期:2009-03-30 返修日期:2009-07-01 本文受国家自然科学基金(批准号:60702075),中国博士后科学基金(批准号:20070410385),四川省教育厅自然科学重点项目(批准号:07ZA014),成都信息工程学院发展基金(批准号:KYTZ200819)资助。

王 鹏(1975-),男,副教授,主要研究方向为量子计算、并行计算, E-mail: wp002005@163.com.

下的世界景象,因此我们首先对量子力学中的不确定性原理进行说明。量子系统中测不准关系本质上是由于微观粒子的波粒二重性导致的。

定义力学量算符 $\hat{A}$ 的平均值:

$$\langle \psi, \hat{A} \psi \rangle = \int \psi^*(r) \hat{A} \psi(r) d^3 r = \bar{A} \quad (1)$$

$\bar{A}$ 为力学量算符 $\hat{A}$ 的平均值。则满足 $(\psi, \hat{A} \varphi) = (\hat{A} \psi, \varphi)$ 的算符 $\hat{A}$ 称为厄米算符。厄米算符的平均值为实数,实验上的可观测量在任何态下的平均值一定为实数,因此相应的算符必为厄米算符。

为说明不确定性原理,引入了力学量算符的对易式(Commutator): $[\hat{A}, \hat{B}] = \hat{A}\hat{B} - \hat{B}\hat{A}$ 。如果 $[\hat{A}, \hat{B}] = 0$ 则说算符 $\hat{A}$ 与算符 $\hat{B}$ 对易。

基于以上定义可以严格证明不确定性原理:

$$\sqrt{(\Delta A)^2} \cdot \sqrt{(\Delta B)^2} \geq \frac{1}{2} |[\hat{A}, \hat{B}]| \quad (2)$$

由式(2)可以看出,如果两个力学量 $A$ 与 $B$ 不对易,则 $A$ 与 $B$ 不能同时测定,或者说它们没有共同本征态。

在文献[7,8]中也证明了自然界中广泛存在的信号具有不确定性关系,并对信号的量子本质进行了描述。我们不能同时对信号的时间和频率进行精确的测量,并由此定义了信号的频率算符 $\hat{\omega} = -i \frac{\partial}{\partial t}$ ,信号的不确定性可以描述如下:

$$\nabla t \nabla \omega \geq \frac{1}{2} |[\hat{t}, \hat{\omega}]| = \frac{1}{2} \quad (3)$$

其中, $\nabla t$ 和 $\nabla \omega$ 为测量误差。

这证明了自然界本身就是一个不确定性的系统。正是由于不确定性的存在才造就了丰富多彩的自然界,从而自然算法对自然界的仿真也打上了不确定性的烙印。离开了不确定性,就无法对自然界进行模仿。

## 1.2 量子迭加态不确定性与量子并行性

量子计算所具有的 $2^n$ 级并行能力,同样也来自于量子系统的不确定性。量子计算机对迭加态数据的处理是量子计算并行性的根源,而量子迭加态中各本征态是以概率的形式存在于迭加态中的,每次对迭加态的测量都可能得到不同的结果<sup>[9,10]</sup>。

对于有 $n$ 个量子比特的量子存储器,同一时刻可以存储 $2^n$ 个信息,如下式:

$$|\Psi_n\rangle = a_1 |00 \cdots 00\rangle + a_2 |00 \cdots 01\rangle + \cdots + a_{2^n} |11 \cdots 11\rangle \quad (4)$$

而在经典情况下,同一时刻只能存储 $2^n$ 个态中的一个。量子计算机对 $n$ 个量子比特存储器实行一次操作,即同时对所存储的 $2^n$ 个数据进行数学运算,等于经典计算机重复实施 $2^n$ 次操作,或等效于同时采用 $2^n$ 个处理器并行处理。

量子系统的并行特性是自然界并行特性的体现,自然界的处理信息能力是十分惊人的。但处于迭加态的量子系统是一个不确定性系统,迭加态 $|\Psi_n\rangle$ 中的所有本征态都是以概率 $a_i$ 出现在测量结果中的。事实上我们通常本不能准确地得到所需结果。但正是这种解空间中大量解的迭加,并形成了一个以概率为基础的不确定性系统,才使量子计算具备了 $2^n$ 量级的并行计算能力。

## 2 自然计算的内在并行性

人工智能算法应具备两个重要的条件:1)向正确的方向

搜索;2)具备对庞大解空间的高速搜索能力。第2个条件表明人工智能算法必须有高效的并行计算能力。自然计算在很大程度上就是模仿了自然的高效并行能力。并行计算能力是几乎所有自然算法都具备的,如遗传算法的隐含并行性。那么自然算法的并行性是如何产生的呢?我们分析了遗传算法、模拟退火算法和神经网络算法,发现它们都不同程度地应用了随机性原理。这应该不是巧合,因为随机性就带来了不确定性。随机方法的使用正是这类算法对自然界不确定性的模仿,并因此具备了强大的并行数据处理能力。

### 2.1 遗传算法的并行性

20世纪50年代Holland模拟生物遗传特性提出遗传算法后<sup>[11,12]</sup>,为求解复杂系统优化问题提出了一个通用框架,并广泛应用于函数优化、组合优化和自动控制。遗传算法的隐含并行性为求解这类复杂问题提供了一个重要的全局化方法。实践证明,遗传算法对于组合优化中NP完全问题非常有效。

遗传算法对于NP类问题的强大处理能力来自于它的隐含并行性。遗传算法的隐含并行性(Implicit Parallelism)表明:遗传算法每一代除了仅对 $n$ 个串的处理外,实际上还处理大约 $O(n^3)$ 个模式,从而每代只执行与群体规模成比例的计算量,就可以同时达到并行地对大约 $O(n^3)$ 个模式进行有效处理的目的,并且无额外的存储。在遗传算法的隐含并行性作用下,根据模式定理(Schema Theorem)具有短的定义长度、低阶并且适应值在群体平均适应值以上的模式在遗传算法迭代过程中将按指数增长率被采样,从而较快地找到问题的结果。

### 2.2 模拟退火算法的并行性

模拟退火算法(Simulated Annealing,简称SA)由Kirkpatrick于1983年在“Science”上正式提出<sup>[13]</sup>,其出发点是基于物理中固体物质的退火过程与一般的组合优化问题之间的相似性,并逐渐发展成一种迭代自适应启发式概率性搜索算法。

Metropolis准则是模拟退火算法中的一个重要数据处理方法。新解的产生按照Metropolis准则,一个新的状态从现有的状态中产生。这个法则能以一定的概率接受能量上升(即产生恶化解)的新状态,而能量下降是优化的总目的。Metropolis准则描述如下:

在温度 $T$ 下由初始解产生新解 $S'$ ,计算增量 $\Delta' = C(S') - C(S)$ ,其中 $C(S)$ 为评价函数,如果 $\nabla t' < 0$ ,则接受 $S'$ 作为当前新的解,否则以概率 $e^{(-\nabla t'/T)}$ 接受 $S'$ 作为新的当前解。

模拟退火算法通过Metropolis准则以一定的概率接受恶化解,实现了对解空间的并行搜索。每一个根据Metropolis准则所录用的恶化解就代表了一类具备一定特征的解空间模式集合,这与遗传算法相似,相当于实现了在对大量隐含模式的搜索,因而模拟退火算法能实现对TSP问题等NP问题的求解。我们可以认为,模拟退火算法所呈现的智能性是由Metropolis准则所提供的算法并行搜索能力造成的。

## 3 不确定性与自然计算并行性的关联

通过对遗传算法、模拟退火算法的分析,我们发现自然计算通常都具有对问题进行并行处理的能力,因而能对部分NP问题给出快速的求解方法。但往往又不能对问题进行绝对精确的处理,即不能保证得到问题的最优解。自然计算的不确定性与并行性具有一种内在的关联性。

### 3.1 自然算法并行性产生的原因

遗传算法、模拟退火算法在算法设计时都采用了概率的方法。遗传算法在进行遗传选择时是基于个体的适应值按概率进行随机选择进入交配池的,并通过变异操作随机地改变个体的基因,从而使整个遗传系统处于一种不确定性的状态。模拟退火算法则通过 Metropolis 准则使算法系统以概率的形式接受恶化解,从而扩大对解空间的搜索范围。理论上我们已经证明模拟退火算法是严格的,前提是算法要有足够多的模型扰动及迭代次数,并且需要配以严密的退火计划。足够多的模型扰动及迭代次数和严密的退火计划表明如果对解空间进行更为精确的搜索,从而降低算法的不确定性程度,则算法是完全有能力进行全局搜索,直到求得最优解的。但此时算法实际上就退化为传统的搜索算法,而不具备处理 NP 问题的智能。

因此我们认为,自然算法的并行性产生的根源是算法所构造的不确定性搜索系统,实现对解空间的并行搜索,但同时付出的代价是无法确保求得问题的最优解。

自然计算中已知具备最强并行性能的量子计算机通过量子迭加态原理,在  $n$  个量子比特上可以迭加  $2^n$  个量子态,所有的量子态均以概率的形式存在于迭加态中。这一迭加系统具备  $2^n$  的并行能力,但也处于一种非常不确定性状态。在对这一系统进行测量时,迭加态将以概率的形式随机坍缩到其中的一个状态。

量子计算机上的算法与普通自然算法具有本质的不同。量子计算机是一种新的计算机工作体系,而普通自然算法只是运行在经典计算机的一类程序,但它们也可以看作是对量子计算机的一种经典仿真,从而实现在不确定性体系下的智能计算。

### 3.2 算法并行性与不确定性的关联

不确定性是自然界的一个普遍规律。Heisenberg 的不确定性原理已对这一规律进行了严格的证明。不确定性所带来的并行性为人工智能的发展提供了一个重要的方法。计算机算法作为解决物质世界问题的一种工具,也应该符合不确定性规律。借鉴 Heisenberg 的不确定性原理定义算法的不确定性原理。

**定理** 经典计算机中算法的不确定性定理 (Uncertainty Principle of Algorithm):

$$P \times A = C \quad (5)$$

其中,  $P$  为算法的并行数据处理能力,  $A$  为处理问题的精确度。问题处理精度越高,  $A$  值越大。  $C$  为常数。

这一定理定性地表明算法的并行能力和处理问题的精确度的乘积为常数,提高算法的并行数据搜索能力的同时,系统处理问题的精度就要降低,即系统就必须是一个不确定性的系统。对于人工智能领域的大多数问题,需要搜索的解空间都十分巨大,因此许多算法都采用牺牲问题求解精度的办法来达到对解空间并行搜索的目的。

图 1 中  $16 \times 16$  的方格可以抽象为解空间,每一个方格代表问题的一个可能解,黑色方格为解空间中的最优解。为了求得这个最优解,采用遍历所有解的方法,那么共需要 256 次运算。这对于解空间巨大的问题,如 NP 问题是不合适的,这类问题可以认为是在高维解空间网格中,如  $16^n$  的问题。如果降低问题的求解精度,通过随机生成  $2 \times 2$  搜索模板中的任意位置对解空间进行遍历,那所需要的计算次数就降低到 64 次,但获得正确解的可能性降低到了  $1/4$ ,这时  $P \times A = 1$ 。如

果采用  $4 \times 4$  的搜索模板,这时算法的并行性  $P = 16$ ,算法的不确定性  $A = 1/16$ ,  $P \times A = 1$ ;如果采用  $8 \times 8$  的搜索模板,这时算法的并行性  $P = 64$ ,算法的不确定性  $A = 1/64$ ,  $P \times A = 1$ ,只需进行 4 次计算就可实现最优解的定位,但真正得到最优解的概率只有  $1/64$ 。在不同的情况下,算法的并行度和精确度的乘积一直是常数 1。

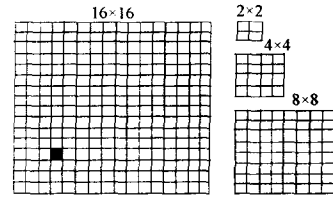


图 1 并行性与不确定性的关系

## 4 自然计算的不确定性智能模型

从前面的分析可以看到,不确定性是许多智能算法的基本特性,这一特性揭示了智能算法的不确定性本质。1948 年,Shannon 提出了“信息熵”的概念,采用不确定性解决了对信息的量化度量问题:

$$H(x) = - \sum_x P(x) \log_2 P(x) \quad (6)$$

一条信息或一个系统信息量的大小和它的不确定性有直接的关系。不确定性越大,熵也就越大。所以,从这个角度可以认为,信息量的度量就等于不确定性的多少。为了实现自然算法对解空间的高速搜索,我们需要算法构造一个处于高不确定状态的搜索系统,即算法系统处于高“信息熵”状态。通常在算法设计时采用随机数的形式来提高算法系统的信息熵水平。

从自然计算的不确定性出发,可以建立自然计算的不确定性智能模型 UIM (Uncertainty Intelligence Model),为自然算法建立一种统一的描述方法。

这一智能模型的工作机制如图 2 所示。根据这一模型我们认为自然智能的核心在于系统处于高信息熵的不确定状态,即智能系统有高的信息含量。这一智能模型由两个重要的部分组成。

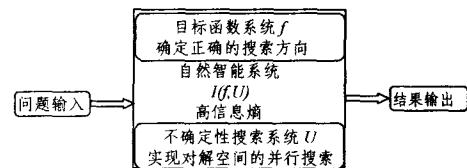


图 2 不确定性智能模型

#### (1) 目标函数系统

这一部分主要是保证智能系统能向正确的方向搜索。根据不同的问题构造合理的目标函数、启发式规则,使智能系统能向正确的方向在解空间进行搜索,从而得到问题的解。这部分信息主要来自先验知识。

#### (2) 不确定性搜索系统

通常智能算法需要解决的问题都具有十分巨大的解空间,甚至有很多问题是 NP 类问题。如果算法不具备对解空间的并行搜索能力,则算法将退化为普通的强力搜索算法。在不确定性智能模型中,我们认为智能系统具备智能的根源在于其算法具有的不确定性,这与人的思维方式是相同的。人的智能就是一种不确定性的智能,这种智能虽然不能保证得到的结果是绝对最优,但却能保证在大多数时候得到的是

一个较优的可行解。

在不确定性智能模型中,目标函数系统和不确定性搜索系统共同形成了具备高信息熵的自然智能系统,保证了自然智能系统能向正确的方向对解空间进行高速并行搜索,从而形成系统对问题的智能处理能力。按照 Shannon 的信息熵理论,在具有相同先验信息的前提下,这类具有不确定性的系统含有的信息量比确定性系统多。从自然界规律来看,含有较多信息量的系统必然就是一个智能系统,如人的大脑。

根据 UIM 模型可把一个智能系统形式化地描述为  $I(f, U)$ ,即智能系统  $I$  本质上是一个信息系统,  $f$  为系统中目标函数信息、启发式信息等先验知识,  $U$  为系统的不确定性水平。  $f$  值越高,系统的知识信息越多;  $U$  越高,系统的不确定性越高,按照 Shannon 的理论信息量也越多。通过这一表述可以得到智能系统设计方法的统一描述:最大可能地提高算法系统的  $f$  和  $U$  值,使智能系统  $I(f, U)$  具备尽可能多的信息量,从而获得高的智能水平。

$f$  和  $U$  的作用虽然都是为了提高系统的信息含量,但对自然智能算法所起的作用却不同。  $f$  的作用是引导算法的前进方向,而  $U$  的作用是提高算法的内在并行能力。如我们向专家系统输入领域知识,就是为了提高智能系统中的  $f$  值。而在遗传算法中采用随机方式对种群进行选择,则是为了提高系统的  $U$  值。但  $U$  值的提高会牺牲算法精确性。这一模型事实上可能推广到其他任意的智能系统。

## 5 一个简单的不确定性智能系统模型实例及分析

Monte-Carlo 方法是一类采用随机数对问题进行求解的方法,典型的算法是对  $\pi$  值的计算问题。该算法给出了计算  $\pi$  值的一种幽雅的算法,从本质上讲就是一种智能算法。该算法的计算方法如图 3 所示。



图 3 Monte-Carlo 方法计算  $\pi$  值

假设图中圆半径为 0.5,则正方形的边长为 1,算法不断生成 0 到 1 之间的随机数,组成二维坐标  $(x, y)$  点,从而确保所有点都落入正方形之内。对落入正方形和落入圆形内的随机点进行计数,分别为  $m$  和  $n$ 。按概率论关系,落入正方形和落入圆形内的点的数目之比在随机点数目趋近于无穷时等于两者面积之比,可以写成

$$\frac{m}{n} = \frac{(2R)^2}{\pi R^2} \quad (7)$$

$$\text{则 } \pi = \frac{4n}{m}。$$

这样只要得到  $m$  和  $n$  的计数就可算出  $\pi$  值。

这么一个计算  $\pi$  值的算法,如果用不确定性智能模型对它进行描述的话,可以形式化地写为:

$$I(f(\pi = \frac{4n}{m}), U(H)) \quad (8)$$

表述表明,智能系统  $I$  由先验知识  $\pi = \frac{4n}{m}$  和不确定度即信息熵  $H$  所构成。很明显,这个计算  $\pi$  值的简单智能系统如果不具备先验知识  $\pi = \frac{4n}{m}$ ,是无法实现计算的,因为算法无法

向正确的方向搜索;另一方面,这一系统如果只有先验知识而没有使用随机数所形成的不确定性,也是无法实现  $\pi$  值计算的。

根据实际的计算,当生成 1 亿个随机点时算出  $\pi = 3.1411$ ;当生成 100 万个随机点时算出  $\pi = 3.1339$ 。随机点的数目越大,计算机精度越高。我们可以用不确定性智能模型对这一结果做出定性的解释:在这一模型中,先验信息  $f$  部分是固定的,不论有多少个随机点,  $f$  部分所包含的信息量都一样,算法的信息量多少由  $U$  来决定。因为本算法要获得精确解则需要生成无限多个随机点,为讨论方便把无限多个随机点数目定义为一个相当大的数值  $K, K \rightarrow \infty$ ,算法停止时已产生的随机点数为  $L$ ,从不确定性上讲已产生的随机点是算法系统已确定的部分,算法停止时未生成的随机点为系统的不确定部分,未生成的随机点数为  $K-L$  个。根据 Shannon 所定义的信息熵,算法在取  $L$  个随机点计算  $\pi$  值后,  $U$  所包含的信息量  $H$  可以近似定性地写为:

$$H = -\sum_{x=1}^K P(x) \log_2 P(x) = -\sum_{i=0}^{K-L-1} \frac{1}{K-L-i} \log_2 \frac{1}{K-L-i} \quad (9)$$

其中,  $\frac{1}{K-L-i}$  为生成第  $L+i$  个随机点时未生成的随机点被选中的概率。可以认为,随着随机点的不断生成,系统的不确定性不断降低。我们可以以此来度量算法在不同的  $L$  值停止时系统信息熵的大小。

对比式(9)和实际计算  $\pi$  值的结果,可以得到如表 1 所列的关系。表 1 中的结果与所提出的不确定性智能模型所描述的结果完全符合。当算法的  $L$  值为 100 万时,算法系统的不确定性大,熵值高,内在并行性强,从而计算  $\pi$  值所需时间短,计算精度低;当算法的  $L$  值为 1 亿时,算法系统的不确定性下降,熵值降低,内在并行性减弱,从而计算  $\pi$  值所需时间加长,计算精度提高。采用 UIM 模型的分析方法事实上可以对许多智能算法的特性做出类似解释。

表 1  $L$  值与算法特性的关系

L 值	H(熵)	算法的不确定性	计算精度	计算时间	算法内在并行性
大	小	小	高	长	弱
小	大	大	低	短	强

**结束语** 通过分析自然算法中的不确定性和并行性的内在关联,认为正是由于算法中采用的不确定性方法提高了系统对解空间的并行搜索能力,使自然算法表现出较好的内在并行能力,成为处理 NP 问题的有力算法工具。由此,我们利用 Shannon 信息熵的观点建立了智能系统的不确定性模型。该模型指出,系统的不确定能提高算法系统的信息含量,从而提高算法的智能水平。该模型为一种基于不确定性原理的人工智能学的数学物理奠定了基础,为设计和构造新的人工智能算法,改进传统的人工智能算法提供了一个新的思考方向,并能有效地对许多智能算法的特性做出合理的解释。

## 参考文献

- [1] Tu X Y. Artificial Intelligence: Review & Prospect[M]. Beijing: Science Press, 2006 (in Chinese with English abstract)
- [2] Shannon C E. A mathematical theory of communication[J]. The Bell System Technical Journal, 1948, 27: 379-423, 623-656
- [3] Gong T, Cai Z X. Research development of natural computation [J]. Control Theory & Applications, 2006, 23 (1): 79-85 (in Chinese)

Chinese with English abstract)

- [4] Li D Y, Cheung D, Ng V, et al. Uncertainty reasoning based on cloud models in controllers[J]. Computers and Mathematics with Application Elsevier Science, 1998, 35(3): 99-123
- [5] Li D Y, Liu C Y, Du Y, et al. Artificial Intelligence with Uncertainty[J]. Journal of Software, 2004, 15(11): 1583-1594 (in Chinese with English abstract)
- [6] Li D Y. Uncertainty in Knowledge Representation[J]. Engineering Science, 2000, 2(10): 73-79 (in Chinese with English abstract)
- [7] Wang P, Li J P. Quantum Interpretation of Frequency Operator [C]//ICNC'07: Third International Conference on Natural Computation, China: IEEE Computer Society Press, 2007: 613-618
- [8] Wang P, Li J P. Quantum Interpretation of Signal's Uncertainty

Principle[J]. Journal of University of Electronic Science and Technology of China, 2008, 37(1): 14-16 (in Chinese with English abstract)

- [9] Feynman R P. Simulating physics with computers[J]. International Journal of Theoretical Physics, 1982, 26(21): 467-488
- [10] Bennett C H, Divincenzo D P. Quantum information and computation[J]. Nature, 2000, 404(3): 247-255
- [11] Holland J H. Outline for a logical theory of adaptive systems [J]. Journal of the Association for Computing Machinery, 1962, 3: 297-314
- [12] Holland J H. Genetic algorithm and the optimal allocations of trials[J]. SIAM Journal of Computing, 1973, 2: 88-105
- [13] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by Simulated Annealing[J]. Science, 1983, 220(4598): 671-680

(上接第 202 页)

CS 的参数  $\rho$  和  $\theta$  设定为使该算法能得到较好的泛化性能和较低的 ISVR, 具体数值在表 2 第 5 列给出。表 3、表 4 给出了两个算法在基准数据集上的识别率(包括训练精度和测试精度)、训练时间和 ISVR。由于训练时间过长, CS 在 german, image 和 splice 数据集上的结果没有报告。

表 3 SSH 和 CS 在基准数据集上的 ISVR 和训练时间

数据	ISVR (%)		训练时间(s)	
	CS	SSH	CS	SSH
banana	73.17	59.92	1227.90	41.27
german	—	49.67	Very long	14.74
image	—	62.59	Very long	702.78
ringnorm	99.17	50.42	1622.00	2.93
splice	—	56.37	Very long	21.77
thyroid	50.71	30.71	8.28	1.26
twonorm	72.92	35.75	920.76	2.97
waveform	57.25	30.83	2435.60	5.08

表 4 SSH 和 CS 在基准数据集上的识别率

数据	训练精度(%)		测试精度(%)	
	CS	SSH	CS	SSH
banana	89.92	84.92	89.25	84.25
german	—	82.43	—	82.43
image	—	90.62	—	90.38
ringnorm	99.83	99.83	99.25	99.25
splice	—	90.80	—	91.10
thyroid	98.33	98.33	98.57	98.57
twonorm	97.58	98.00	96.75	96.50
waveform	93.42	94.17	90.25	90.25

(在表 3、表 4 中, 由于 CS 算法的一些结果没有在 4 个小时内得到, 我们不得不停止了计算。)

由表 3、表 4 可以看出, SSH 的识别率与 CS 相当, 但是 SSH 的训练时间远少于 CS。即使加上参数值的寻找时间, SSH 总的时间仍然远小于 CS。很明显, 我们的算法在计算效率上具有异常突出的优势。SSH 的 ISVR 也明显低于 CS。这表明我们的算法在内存空间和 CPU 时间的占用上比 CS 更经济。这得益于 SSH 本身的高效率节省了大量的时间, 同时其较低的 ISVR 降低了训练代价。

**结束语** 本文提出了一种渐进支持向量获取的新方法, 即支撑分离面法。其主要思想基于下面事实: 支持向量位于间隔边界上, 如果支撑分离面会变成间隔边界或与其比较接近, 那么位于其上的样本将极其可能成为支持向量, 进而成为 ISVM 的渐进支持向量。该方法根据样本点的支撑分离率来估计该样本成为渐进支持向量的概率, 进而选取概率值较高的样本作为渐进支持向量去训练和更新 ISVM。实验结果表

明该方法能够高效地获取渐进支持向量, 能极大地提高 ISVM 的训练速度却不牺牲其泛化能力。

### 参考文献

- [1] Jia Hongbin, Lu Yi, Murphey, et al. Identifying knowledge domain and incremental new class learning in SVM[C]//JCNN '05. 2005: 2742-2747
- [2] Wan Sheng, Banta L E. Parameter Incremental Learning Algorithm for Neural Networks[J]. IEEE Trans. Neural Networks, 2006, 17(6): 1424-1438
- [3] Vapnik V. The Nature of Statistical Learning Theory[M]. London, UK: Springer-Verlag, 1995
- [4] Burges C J C. A Tutorial on Support Vector Machines for Pattern Recognition[J]. Data Mining and Knowledge Discovery, 1998, 2(2): 1-47
- [5] Cristianini N, Shawe-Taylor J. An introduction to Support Vector Machines and other kernel-based learning methods[M]. Cambridge, UK: Cambridge University Press, 2000
- [6] Mitra P, Murthy, et al. Data condensation in large databases by incremental learning with support vector machines[C]//Proc. Internat. Conf. on Pattern Recognition. 2000: 2708-2711
- [7] Domeniconi C, Gunopulos D. Incremental Support Vector Machine Construction[C]//Proc. ICDM 2001. 2001: 589-592
- [8] Xiao Xianbo, Hu Guangshu. An Incremental Support Vector Machine based Speech Activity Detection Algorithm[C]//Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 2005: 4224-4226
- [9] An Jin-long, Zhen Gou-wang, et al. An Incremental Learning Algorithm for Support Vector Machine[C]//Proceedings of the Second International Conference on Machine Learning and Cybernetics. Xi'an, 2003: 1153-1156
- [10] Katagiri S, Abe S. Selecting support vector candidates for incremental training[C]//IEEE International Conference on Systems, Man and Cybernetics. 2005(2): 1258-1263
- [11] Katagiri S, Abe S. Incremental training of support vector machines using hyperspheres[J]. Pattern Recognition Letters, 2006(27): 1495-1507
- [12] Shawe-Taylor J, Cristianini N. Kernel Methods for Pattern Analysis[M]. UK: Cambridge University Press, 2004
- [13] Joachims T. Making large-scale SVM learning practical[M]//Schölkopf B, Burges C, Smola A, eds. Advances in Kernel Methods-Support Vector Machines. Cambridge, MA: MIT Press, 1999