复杂性与智能化：从 Church-Turning Thesis 到 AlphaGo Thesis 及其展望 (2)

王飞跃 [1,2]

摘 要 软件围棋机器人 AlphaGo 为解决复杂性与智能化问题带来新的思路, 对未来智能军事和智慧战争的指挥与控制问题可能具有深刻影响, 并且从技术方面证实了虚实互动的平行理念与方法是建立新军事体系的有效途径.

关键词 AlphaGo, 复杂性, 智能化, 智能军事, 智慧战争, 平行军事, 平行指控

引用格式 王飞跃. 复杂性与智能化：从 Church-Turning Thesis 到 AlphaGo Thesis 及其展望 (2) [J]. 指挥与控制学报, 2016, 2(2): 89−92

DOI 10.3969/j.issn.2096-0204.2016.02.0089

Complexity and Intelligence: From Church-Turning Thesis to AlphaGo Thesis and Beyonds (2)

WANG Fei-Yue [1,2]

Abstract AlphaGo's victory over a top human professional player offers a new way of thinking on engineering solution for complexity and intelligence, called "The Extended AlphaGo Thesis" here. This might have a significant impact on the command and control of future intelligent military and smart wars, and has verified technically the soundness of concept and method of parallel systems based on virtual-real interaction as an effective approach for a new military system of systems.

Key words AlphaGo, complexity, intelligence, intelligent military, smart wars, parallel military, parallel command and control

Citation Wang Fei-Yue. Complexity and intelligence: from Church-Turning Thesis to AlphaGo Thesis and beyonds (2)[J]. Journal of Command and Control, 2016, 2(2): 89−92

2 基于信息的复杂性及其分析

何谓复杂性? Kolmogorow 复杂性或算法熵是关于现代复杂性研究的开始. 如何在智能系统研究中引入复杂性分析, 是我 30 年所面临的问题. 我在 McNaughton 和 Saridis 的帮助下, 曾写过 3 份研究报告 [18−20], 其中第 1 份基本被否定, 但第 3 份引入基于信息的复杂性概念, 主要受 S. F. Traub 等的工作影响 [21−26], 自己还算满足. 后由于其他工作的影响, 这项研究没有再深入, 所以一直没有发表, 许多年后, 连研究工作报告也不知在何处. 万幸的是, 我的博士生在我从美国带回的材料中, 竟然找到了这份近 30 年前写的研究报告 [20]. 在此, 我摘录其中的 1 节于此, 让大家了解自己当时从事复杂性研究的思路, 或许对关于智能与复杂性的更进一步研究有所帮助.

Complexity Analysis in Intelligent Machines

In this section we address the issue of complexity analysis in Intelligent Machines. The motivation of performing complexity analysis for an Intelligent Machine is to determine the intrinsic limit and ability on which tasks can be accomplished by that Machine.

The performance of task processing of Intelligent Machines will be measured by precision or reliability. To include the reliability as the measure of performance is based on the consideration that for many tasks, especially the higher level tasks, in an Intelligent Machine, their solutions (i.e., ways

收稿日期 2016-06-10
Manuscript received June 10, 2016

1. 国防科技大学军事计算实验与平行系统技术研究中心 湖南 长沙 410073 2. 中国科学院自动化研究所复杂系统管理与控制国家重点实验室 北京 100190
1. Research Center of Military Computational Experiments and Parallel Systems Technology, The National University of Defense Technology, Changsha Hunan 410073, China 2. The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

to accomplish tasks) are not unique and are not defined in the conventional normed spaces (e.g., the normed linear space G in information-based complexity), therefore the terms "error" and then "precision" are not well defined for those tasks. Instead, the performance of those tasks are usually described by the degree of satisfaction of certain specifications (McInroy and Saridis, 1989). Reliability can then be used as a probabilistic measure of assurance of performance. Like the precision of a task processing, its reliability can be computed off-line or estimated by on-line observation.

The two central questions of complexity analysis for an Intelligent Machine are,

- What is the minimal cost to accomplish a task by the Machine for a given level of precision or reliability?
- What is the maximal precision or reliability accomplished for a task by the Machine with a specified level of cost?

It is very important to point out that here we discuss the complexity within the context of a specific Machine, not with respect to all Machines. In other words, we talk about the complexity of a task processing by an arbitrary single Machine, which is a machine-dependent quantity, not the complexity of a task processing by all Machines, which is a machine-independent quantity. The reason for this is simple: we want to find a way to measure the ability and limit of a specific Intelligent Machine, not that of all Intelligent Machines. The later is too complicated to be accomplished at the current stage.

Following the formulation of information-based complexity, we will formalize the problem of complexity analysis for Intelligent Machines. Since we just investigate the complexity of task processing of a specific Machine, unlike in the information-based complexity where almost all possible information operators and algorithms are assumed to be available, we have to assume that the specific Machine has only finite number of information operators and algorithms for its task processing.

A. Complexity of Machine Precision: The formulation of *complexity of machine precision* is a triple

$$\mathrm{CMP} = (\mathrm{TS_p},\ \mathrm{DB},\ \mathrm{MI})$$

where

1. $\mathrm{TS_p}$ =(T, D, E, ε) is *task specification*:

   - T: a finite set of *tasks*. An element f from T is called a task.
   - D: T→ DB×MI is the *decision mapping*. D(f)=($\mathrm{N_D}$, ϕ_D) indicates a decision to processing task f with information operator $\mathrm{N_D}$ and algorithm ϕ_D.
   - E: D(T)→ $\mathrm{R_+}$ is the *error function*. E(D(f)) defines the error of the decision D at task f. $\mathrm{R_+} = [0, +\infty)$.
   - $\varepsilon \in \mathrm{R_+}$ is the specification of *precision*: D is called an *ε-approximate decision* at f iff E(D(f))$\leq \varepsilon$. The error of a decision D can be defined in different settings (e.g., the worst, average, and probabilistic cases). For simplicity, we consider only worst case setting, therefore, the *worst case error* of D is defined by

     $$\mathrm{e(D)} = \max\{\mathrm{E(D(f))} : \mathrm{f} \in \mathrm{T}\}.$$

     D is called an *ε-approximate decision* iff e(D)$\leq \varepsilon$.

2. DB=(H, Λ, $\mathrm{N_1}$, ..., $\mathrm{N_s}$) is *database specification*:

   - H: a set of measurements of the database.
   - Λ: a finite set of primitive *information operations* of the database. An element L from Λ is a mapping L: T→ H. L(f) represents the information about f through the computation of the form L(f). The cost of each information operation is assumed to be a constant c_{DB} (c_{DB} primitive operations).
   - $\mathrm{N_i}$: T→ K is an *information operator*. $\mathrm{N_i}$(f) specifies the information about task f obtained by $\mathrm{N_i}$. K= {[$\mathrm{L_1}$(f), ..., $\mathrm{L_m}$(f)]: f∈T, $\mathrm{L_i} \in \Lambda$, m $>$ 0} is the set of all finite sequences of information operations.

3. MI=(Ω, ϕ_1, ..., ϕ_t) is *algorithm specification*:

- Ω: a set of primitive *computational operations*. The cost of each computation operation is assumed to be a constant c_{MI} (c_{MI} primitive operations).

- ϕ_{i}: $\mathrm{N_1(T)\cup...\cup N_s(T)\to G}$ is an *algorithm*, where G is the set of task solutions. An algorithm combines the known information N(f) and produces an approximate solution for a task, using computational operations in Ω.

Now we can define the *ε-complexity of machine precision*. The total cost of decision $\mathrm{D(f)=(N_D,\ \phi_D)}$ at task f is given by

$$\mathrm{cost(D,\ f) = cost(N_D,\ f) + cost(\phi_D,\ N_D(f))} \quad (1)$$

where $\mathrm{cost(N_D,\ f)}$ is the *database cost* of computing $\mathrm{N_D(f)}$ and $\mathrm{cost(\phi_D,\ N_D(f))}$ is the *decision cost* of computing $\phi\mathrm{(N_D(f))}$. The cost here is defined in terms of the number of *primitive operations* performed. The primitive operations include the addition, multiplication, etc. The cost of computation of decision D in the worst case setting is given by

$$\mathrm{cost(D) = \max\{cost(D,\ f) : f \in T\}};$$

The *ε-complexity of machine precision* is then defined as the minimal cost among all decisions with error at most ε,

$$\mathrm{comp_{MP}}(\varepsilon) = \min\{\mathrm{cost(D) :\ D\ such\ that\ e(D)} \leq \varepsilon\}; \quad (2)$$

$\mathrm{comp_{MP}}(\varepsilon)$ is the measure of difficulty of processing tasks with the given precision ε by an Intelligent Machine. A task cannot be accomplished with precision ε if the operational resource assigned is less than $\mathrm{comp_{MP}}(\varepsilon)$. Tasks that cannot be achieved because limitations dictate that the requisite operational resources cannot be granted by the Machine are said to be *intractable* by the Machine.

C-precision is defined as the minimal error among all decisions which accomplish the tasks with cost at most C,

$$\mathrm{pres}(C) = \min\{\mathrm{e(D)\!:\ D\ such\ that\ cost(D)} \leq C\}; \quad (3)$$

$\mathrm{pres}(C)$ is the upper bound of precision of processing tasks with the given operational resource C. Precision higher than $\mathrm{pres}(C)$ cannot be achieved with the resource bounded by C.

B. Complexity of Machine Reliability: The formulation of *complexity of machine reliability* is a triple

$$\mathrm{CMR = (TS_R,\ DB,\ MI)}$$

where
1. $\mathrm{TS_R}$ =(T, M, S, D, δ) is *task specification*:

- T: a finite set of *tasks*. An element f from T is called a task.

- M: a finite set of *specifications*.

- S: T→M is the *specification mapping*. S(f) represents the specifications to be met by task f.

- D: T→DB×MI is the *decision mapping*. $\mathrm{D(f)=(N_D,\ \phi_D)}$ indicates a decision to processing task f with information operator $\mathrm{N_D}$ and algorithm ϕ_D.

- $\delta \in [0, 1]$ is the specification of *reliability*: D is called an *δ-reliable decision* at f iff Prob[S(f) is true | D(f)]≥ δ. The *worst case reliability* of a decision D is defined by

  $$\mathrm{r(D) = \max\{Prob[S(f)\ is\ true\ |\ D(f)]\!:\ f \in T\}};$$

  D is called an *δ-reliable decision* iff r(D)≥ δ.

2. DB=(H, Λ, $\mathrm{N_1}$, ..., $\mathrm{N_s}$) is the same database specification as in the formulation of complexity of machine precision.
3. MI=(Ω, ϕ_1, ..., ϕ_t) is the same algorithm specification as in the formulation of complexity of machine precision.

Now we can define the *δ-complexity of machine reliability*. As for complexity of machine precision, the total cost of decision $\mathrm{D(f)=(N_D,\ \phi_D)}$ at task f is given by

$$\mathrm{cost(D,\ f) = cost(N_D,\ f) + cost(\phi_D,\ N_D(f))},$$

and the cost of computation of decision D in the worst case setting is given by

$$\mathrm{cost(D) = \max\{cost(D,\ f)\!:\ f \in T\}}.$$

The δ-*complexity of machine reliability* is then defined as the minimal cost among all decisions with reliability at least δ,

$$\mathrm{comp_{MR}}(\delta) = \min\{\mathrm{cost(D)\colon D\ such\ that\ r(D)} \geq \delta\}; \tag{4}$$

$\mathrm{comp_{MR}}(\delta)$ is the measure of difficulty of processing tasks with the given reliability δ by an Intelligent Machine. A task with the specified reliability δ is intractable if the operational resource assigned is less than $\mathrm{comp_{MR}}(\delta)$.

C-reliability is defined as the maximum reliability among all decisions which accomplish the tasks with cost at most C,

$$\mathrm{rely}(C) = \max\{\mathrm{r(D)\colon D\ such\ that\ cost(D)} \leq C\}; \tag{5}$$

$\mathrm{rely}(C)$ is the upper bound of reliability of processing tasks with the given operational resource C. Reliability higher than $\mathrm{rely}(C)$ cannot be achieved with the resource bounded by C.

The main difference between the information-based complexity and the machine complexities is that, in information-based complexity, the complexity is computed on the convention that the same information operator and algorithm will be used for all the problems, however, in both machine precision complexity and machine reliability complexity, the complexity is calculated on the assumption that the different information operator and algorithm are allowed to be used for different tasks. The decision to find suitable information operator and algorithm for a task is specified by the decision mapping D. This is because the facts that only a finite number of information operators and algorithms is available for any kind of task processing by any specific Intelligent Machine and that a Machine is capable of obtaining different information and applying different controls for different tasks.

The complexity formulation of machine precision or reliability can be applied in complexity analysis of any task processing units in an Intelligent Machine. Depending on the connection configuration of task processing units is sequential or parallel, the resulting complexity will be the maximum of complexities of task processing units or the sum of complexities of task processing units. For example, in Petri net model of the Coordination Level of an Intelligent Machine (Wang and Saridis, 1990), each of the transitions in a Petri net transducer can be considered as a task processing unit, and therefore we can use the above complexity formulations in their complexity analysis. When complexities of information operators and algorithms ($\mathrm{cost(N_D,\ f)}$ and $\mathrm{cost}(\phi_\mathrm{D},\ \mathrm{N_D(f)})$) are unknown and uncertainty but can be observed from the task execution, they can be used as the performance indices in the on-line learning algorithm for decision making in Petri net transducer.

By the same arguments made for information-based complexity, equation (5) indicates that to processing tasks by a task unit of Machine with the given operational resource and the specified precision (or reliability), higher intelligent algorithms have to be used for small size databases and lower intelligent algorithms have to be used for large size databases. This observation verifies the principle of IPDI of Intelligent Machines for any specific task processing unit. Since complexity is additive, we can reach the conclusion that the principle holds for any subsystem of an Intelligent Machine.

（未完待续）

王飞跃 (1961−), 男, 博士, 研究员, 国防科学技术大学军事计算实验与平行系统技术研究中心教授和主任, 中国科学院自动化研究所复杂系统管理与控制国家重点实验室研究员和主任. 国防领域“国家特聘专家”. 主要研究方向为智能系统和复杂系统的建模, 分析与控制.
E-mail: wangfeiyue@nudt.edu.cn