

控制系统生存核的逼近算法*

陈征^{1,†} 高岩²

摘要 生存核的计算是控制理论中的一个重要研究方向. 给出了一种计算一般离散控制系统生存核的新算法. 基于机器学习的方法, 给出了逼近生存核的算法. 并在一定条件下, 证明了此算法的收敛性. 此算法在一定程度上避免了计算量随控制空间的维数增长而指数增长的问题. 最后, 给出具体的实际例子来说明算法的有效性.

关键词 离散系统, 生存核, 机器学习

中图分类号 O231.2

2010 数学分类号 93C10, 33F05

Approximating viability kernel for control systems*

CHEN Zheng^{1,†} GAO Yan²

Abstract The computation of the viability kernel is an important topic in control theory community. In this paper, we propose a new algorithm that computes the viability kernel of a discrete-time system. Based on the theory of machine learning, the algorithm of approximating viability kernel is presented. We give some conditions that guarantee the convergence of the approximations towards the actual viable kernel. This method avoids the exponential growth of the computing time with the dimension of the control space. Finally, examples are given to illustrate this result.

Keywords discrete-time systems, viability kernel, machine learning

Chinese Library Classification O231.2

2010 Mathematics Subject Classification 93C10, 33F05

0 引言

所谓系统的生存性指的是对于某一区域内的任何初始值, 系统的运动都不离开此区域^[1]. 生存性是目前控制理论中的一个重要研究内容, 它在系统的稳定性分析和镇定研究以及混杂系统安全性设计中都有广泛的应用^[2-4]. 生存理论还广泛应用于各类学科的控制系统中, 如经济、管理和生态学等^[5,6]. 目前, 对于系统在某一特定区域内可生存的判断, 人们给出了一些方法^[7,8]. 当系统在某一区域内不可生存时, 希望能找出包含在此区域的一个最大子集, 使得对于子集中的任意一点作为初始值, 系统的轨迹都不离开此

收稿日期: 2012年12月10日

* 基金项目: 国家自然科学基金(Nos. 11171221, 40901241), 上海市科委与地方院校能力建设(10550500800), 上海市一流学科(XTKX2012)

1. 宁波工程学院理学院, 浙江宁波 315016; School of Sciences, Ningbo University of Technology, Ningbo 315016, Zhengjiang, China

2. 上海理工大学管理学院, 上海 200093; Management School, University of Shanghai for Science and Technology, Shanghai 200093, China

† 通讯作者 Corresponding author, Email: chenzhengnb@tom.com

区域. 这样的集合称为系统的生存核^[1], 其计算是生存理论中的一个重要方向, 并在生态学、语言学和食品加工等领域有广泛的应用^[9-11]. 文献[12]给出了两类二阶线性系统的生存核表示方法, 并猜想此结论在系统高阶时亦成立. 文献[13]考虑了单输入二阶非线性系统的生存核计算问题, 给出了生存核的具体表示. 但目前对一般系统的生存核计算缺少通用的方法, 而且生存核的计算随问题的维数增加, 其计算量呈现指数增长, 缺少高效的计算方法^[14,15].

本文将基于机器学习的方法, 给出一种不确定离散系统的生存核计算方法. 此方法的优点是避免了前面提到的维数灾难而且计算量小, 在一定条件下收敛到真实的生存核. 另外, 可以采用不同的分类方法, 来比较结果的好坏.

1 控制系统的生存核

考虑一般的离散控制系统

$$x^+ = f(x, u, w), \quad (1.1)$$

其中 x 为状态变量, x^+ 为 x 的下一个状态, $u \in U$ 为控制输入, $w \in W$ 是干扰. 这里的干扰往往是有限但不知道分布.

下面给出一些关于生存性和生存核的概念(见[16,17]).

定义 1.1 设 $K \subseteq \mathbb{R}^n$, 对于离散系统(1.1), 如果存在 $u \in U$, 对于所有的 $x \in K$ 和 $w \in W$, 都有 $f(x, u, w) \in K$, 那么称系统(1.1)在集合 K 上是鲁棒可生存的.

由此可以得到如下的关于生存的性质.

性质 1.1 系统(1.1)在闭集 K 上鲁棒可生存的充要条件是:

$$\exists u \in U, \exists f(x, u, w) \subseteq K, \forall x \in K, w \in W.$$

如果是某一点作为初始点, 系统(1.1)后的轨迹都不离开闭集, 那么称这一点是鲁棒可生存的点.

当系统无法满足生存条件时, 希望找到如下的生存核 $Viab(K)$.

定义 1.2 闭集 K 关于控制系统(1.1)的生存核为如下的集合:

$$Viab(K) = \{x_0 \in K | \exists u \in U, \text{系统(1.1)的轨迹}x\text{始终包含在}K\text{中, 对任意的}w \in W\}.$$

可见生存核是最大的鲁棒可生存状态的集合.

下面给出一些定义.

定义 1.3 对于系统(1.1), 定义如下集合

$$Pre(K) = \{x | \exists u \in U, \forall w \in W, \text{有}f(x, u, w) \in K\}.$$

显然, 上面的集合是由所有通过系统(1.1)一步能到集合 K 的状态所组成的集合.

定义 1.4 对于函数 f , 用如下的记号来表示函数的复合

$$f^{[2]}(x) = f(f(x)),$$

$$f^{[m+1]} = f(f^{[m]}(x)), m \geq 2.$$

这里, 设系统(1.1)能写成如下的差分包含

$$x^+ \in F(x),$$

其中 $F(x) = \{f(x, u, w) | u \in U, w \in W\}$, 这里 $F(x)$ 是闭集.

定义 1.5 映射 $F: X \rightarrow 2^X$ 是 μ -Lipschitz 的当且仅当存在常数 $\mu > 0$ (被称为Lipschitz常数)使得对所有的 $x, x' \in X$, 有

$$F(x) \subseteq F(x') + \mu \|x - x'\| B(0, 1).$$

2 机器学习过程

所谓机器学习是研究计算机怎样模拟或实现人类的学习行为, 以获取新的知识和技能, 重新组织已有的知识结构使之不断改善自身的性能^[18,19].

机器学习的标准过程如下^[18]:

考虑一组数据 $(X_1, Y_1), \dots, (X_n, Y_n)$, 称为训练样本. 其中 $X_i = (X_{i1}, \dots, X_{id}) \in \chi \subseteq \mathbb{R}^n, i = 1, \dots, n$ 是 d 维向量, X_i 称为特征. Y_i 取值于有限集合 $\gamma = \{-1, +1\}$. 这里提到的 $-1, +1$ 表示不同的两种属性. 然后再根据训练样本, 得到一个分类机器 $h: \chi \rightarrow \gamma$. 最后如果有新的 X 出现, 可以根据机器 $h(X)$ 等于 -1 或 $+1$ 来给出 X 的分类. 常见的分类机器有: 线性回归、最近邻、支持向量机、决策树等. 可见机器学习的关键是根据训练样本来获得一个好的分类机器, 便于当有新的数据输入时, 分类机器给判断是 -1 或 $+1$.

3 生存核的逼近算法

首先给出一个紧集 K , 给出一个 K 的网格 $K_h \subseteq K$, K_h 是一个如图1的有限集合, 即图1中的黑点所形成的集合.

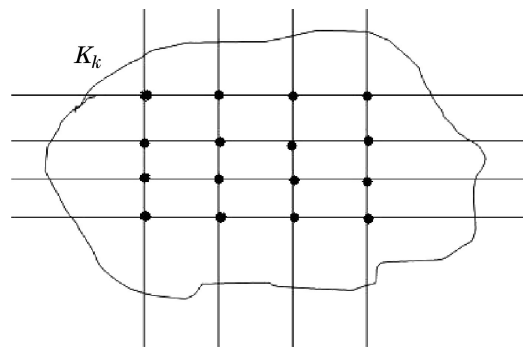


图1 网格示意图

并满足 $\forall x \in K, \exists x_h \in K_h, \exists \|x - x_h\| \leq \beta(h)$, 其中 $\beta(h) \rightarrow 0$, 当 $h \rightarrow 0$ 时.

接下来, 将讨论如何求 k 的关于系统(1.1)的生存核, 首先要由 K_h 来构造求生存核的训练样本. 下面给出具体步骤.

方法一

步骤1 初始化 $m = 0$, 令 $K_h^m = K_h, K^m = K$;

步骤2 $K_h^{m+1} = K_h^m \cap Pre(K^m)$;

步骤3 如果 $K_h^{m+1} \neq K_h^m$, 令 $K^{m+1} = K^m \cap Pre(K^m)$. $m := m + 1$, 并转步骤2;

步骤4 如果 $K_h^{m+1} = K_h^m$, 得到如下的训练样本:

如果 $x_h \in K_h^m$, 则标上+1; 如果 $x_h \notin K_h^m$, 则标上-1.

引理 3.1 如上的训练样本产生法必在有限步后停止.

证明 在每一步, 由方法一有 $K_h^{n+1} \subseteq K_h^n$. 由于 K_h 一但取定就是有限个的, 所以经过有限步后算法必然停止.

特别地, 如果控制系统是如下的自治系统:

$$x^+ = f(x). \quad (3.1)$$

训练样本产生方法可以简化为如下的方法.

方法二

步骤1 初始化 $m = 0, K_h^m = K_h$;

步骤2 $K_h^{m+1} = \{x_h | x_h \in K_h^m, f^{[m+1]}(x_h) \in K\}$;

步骤3 如果 $K_h^{m+1} \neq K_h^m$, $m := m + 1$, 并转步骤2;

步骤4 如果 $K_h^{m+1} = K_h^m$, 得到如下的训练样本:

如果 $x_h \in K_h^m$, 则标上+1; 如果 $x_h \notin K_h^m$, 则标上-1.

可见, 系统(3.1)的生存核的计算大大地简化了. 只要计算有限个函数值就可以了.

训练样本生成后, 需要选择合适的分类机器来判断集合 K 中哪些是属于生存核的. 为了生成较有效的分类机器, 可以用文[18]中提到的AdaBoost算法. 其优点是可以由若干个弱分类机器生成一个最终的强分类机器. 而强分类机器分类的错误率会极大地降低.

设生成的分类机器为 h , 则集合 K 的近似生存核表示如下:

$$V = \{x | x \in K, h(x) = +1\}.$$

显然, 近似生存核 V 的好坏取决于训练样本的取法和分类机器的选择, 可以进一步用数值方法来检验结果的好坏.

下面将讨论算法的收敛性. 由引理3.1, 假设算法停止时有 $K_h^{p+1} = K_h^p$.

定理 3.1 假设 $F(x)$ 关于 x 是 μ -Lipschitz的. 存在 $\lambda > 0$, V 满足如下的条件

$$\forall x \in V, \exists x_h \in K_h^p, \exists \|x - x_h\| \leq \lambda\beta(h), \quad (3.2)$$

$$\forall x \in K \setminus V, \exists x_h \in K_h \setminus K_h^p, \exists \|x - x_h\| \leq \beta(h), \quad (3.3)$$

$$V \subseteq K^p. \quad (3.4)$$

那么当 $h \rightarrow 0$, V 逼近真实的生存核 $Viab(K)$.

证明 首先证明 $Viab(K) \subseteq K$.

假设 $x \notin V$.

(a) 如果 $x \notin K^p$, 由于 $Viab(K) \subseteq K^p$, 那么 $x \notin Viab(K)$.

(b) 如果 $x \in K^p$, 由条件(3.3), 有

$$\exists x_h \in k_h \setminus k_h^p, \exists \|x - x_h\| \leq \beta(h).$$

又由方法一的构造有

$$x_h \in k_h \setminus k_h^p, F(x_h) \cap (K \setminus K^{p-1}) \neq \phi.$$

由于 $K^p \subseteq K^{p-1}$, 所以 $F(x_h) \cap K \setminus K^p$. 于是有 $d(F(x_h), K^p) > \mu\beta(h)$, 又由于 $F(x)$ 是 μ -Lipschitz 的, 所以

$$F(x) \subseteq F(x_h) + \mu\beta(h)B(0, 1).$$

故有

$$F(x) \subseteq (K \setminus K^p).$$

由(a), $F(x)$ 中的点都是不可生存的, x 也是不可生存的, 也就是 $x \notin Viab(K)$. 综合(a)、(b), 有 $x \notin V \Rightarrow x \notin Viab(K)$, 即 $Viab(K) \subseteq V$.

其次, 假设如下的集值映

$$x^+ \in F_h(x) = F(x) + \mu(1 + \lambda)\beta(h)B(0, 1), \quad (3.5)$$

将证明 V 包含于集合 K 关于(3.5)的生存核 $Viab_F(K)$.

设 $x \in V$. 由(3.2), 得

$$\exists x_h \in K_h^p, \exists \|x - x_h\| \leq \lambda\beta(h).$$

由方法一,

$$x_h \in K_h^{p+1} \Rightarrow d(F(x_h), K^p) \leq \mu\beta(h). \quad (3.6)$$

又 $F(x)$ 是 μ -lipschitz 的, 所以

$$F(x) \subseteq F(x_h) + \lambda\mu\beta(h)B(0, 1). \quad (3.7)$$

考虑(3.6)和(3.7)以及三角不等式, 有

$$d(F(x), K^p) \leq \mu(1 + \lambda)\beta(h).$$

这说明 $F_h(x) \cap K^p \neq \phi$. 于是任意的 $x \in K^p$, 存在轨迹始终在 K^p 内, 所以 $K^p \subseteq Viab_F(K)$.

最后, 因为 $F(x)$ 是 μ -lipschitz 的, 集合 K 是紧的. 所以对于 $\forall \varepsilon > 0$, $\exists \eta > 0$ 满足 $h < \eta$, 有

$$Viab_F(K) \subseteq (Viab(K) + \varepsilon B(0, 1)).$$

又由上面的证明, 并考虑(3.4)有

$$Viab(K) \subseteq V \subseteq Viab_F(K).$$

因此, 当 $h \rightarrow 0$, V 逼近真实的生存核 $Viab(K)$.

注3.1 条件(3.4)是非常合理的, 如果 V 有部分没在 K^p 中, 则这一部分必不属于生存核. 因此可以去掉.

4 数值仿真

下面给出具体的仿真例子.

例1 汽车在高速公路上的速度变化方程为:

$$x^+ = 2x + u,$$

其中 x 表示车的速度, u 表示用发动机或刹车可以控制的加速.

这里假设集合 $K = [0, 2]$, $U = [-1, 1]$, 这里不考虑干扰. 容易计算其真实的生存核是 $[0, 1]$, 见图2.

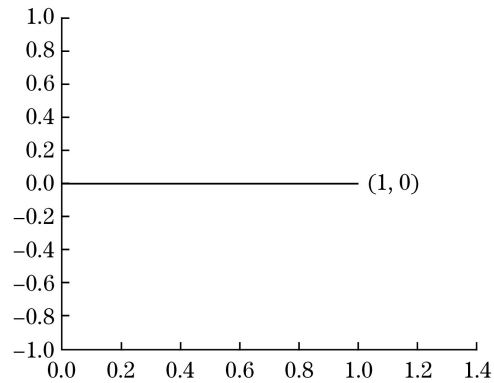


图2 例1中的真实生存域

下面用机器学习的方法来求生存核.

首先将 K 等分成100个区间(形成网格):

$$[0, 0.02], [0.02, 0.04], \dots, [1.98, 2]$$

于是考虑点 $0, 0.02, \dots, 1.98, 2$ 这101个点来生成训练样本. 由方法一得如下的训练样本:

属于生存核的训练样本: $(0, +1), (0.02, +1), \dots, (1, +1)$ 共51个, 并记为 $(x_i, y_i), i = 1, \dots, 51$.

不属于生存核的训练样本: $(1.02, -1), (1.04, -1), \dots, (2, -1)$ 共50个, 并记为 $(x_i, y_i), i = 52, \dots, 101$.

下面就是选择分类机器的问题了. 这里机器用线性回归进行分类, 即对 $(x_i, y_i), i = 1, \dots, 101$ 进行线性回归. 假设

$$Y = \beta_0 + \beta_1 X.$$

由数据 (x_i, y_i) , 用最小二乘法得出系数 β_0, β_1 . 于是有

$$Y = 1.495 - 1.485X.$$

于是得到分类机器:

$$h(X) = \begin{cases} +1, & 1.495 - 1.485X > 0, \\ -1, & 1.495 - 1.485X < 0. \end{cases}$$

于是分界点就是 $1.495 - 1.485X = 0$, 即 $X = 1.007$. 所以近似生存核是 $[0, 1.007)$, 与真实生存核相比多出了 0.7% , 逼近程度很好, 见图3. 用R软件计算, 其计算量很小.

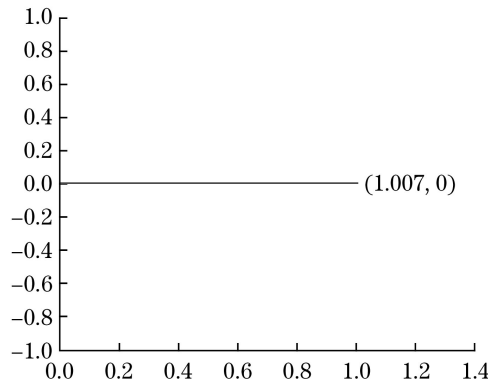


图3 例1中的近似生存域

例2 考虑如下的二维系统:

$$\begin{pmatrix} x_1^+ \\ x_2^+ \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix},$$

这里 $U = [-40, 40] \times [-20, 20]$, $K = [0, 120] \times [0, 120]$. 这里真实的生存核是 $[0, 80] \times [0, 40]$.

下面将 $x_1 \in [0, 120]$ 等分成120份, 又将 $x_2 \in [0, 120]$ 等分成120份. 取121条直线 $x_1 = 0, x_1 = 1, \dots, x_1 = 120$ 和121条直线 $x_2 = 0, x_2 = 1, \dots, x_2 = 120$, 如此生成了集合 K 的一个网格, 共有 121×121 个点.

根据方法一, 得到如下的样本点:

属于生存核的训练样本:

$$(i, j, +1), i = 0, 1, \dots, 80, j = 0, 1, \dots, 40,$$

共 81×41 个点.

不属于生存核的训练样本:

$$(i, j, -1),$$

其中 (i, j) 属于集合 $(x_1, x_2)^T \in (K \setminus G_1)$, 且 $x_1, x_2 \in \mathbb{Z}$, 其中 $\{G_1 = \{(x_1, x_2)^T | 0 \leq x_1 \leq 80, 0 \leq x_2 \leq 40\}\}$, 共11 320个点.

此时如果采取线性回归进行分类, 会得到如下的分类机器:

$$h(X) = \begin{cases} +1, 0.792\ 319 - 0.007\ 537x_1 - 0.014\ 874x_2 > 0, \\ -1, 0.792\ 319 - 0.007\ 537x_1 - 0.014\ 874x_2 < 0. \end{cases}$$

其中 X 是所有 K 中的点.

由分类机器, 可以得到分界线为

$$0.792\ 319 - 0.007\ 537x_1 - 0.014\ 874x_2 = 0,$$

比较真实的生存核, 得其错误率是0.0728. 发生错误的地方在图4中的三个用斜线标出的三角形区域.

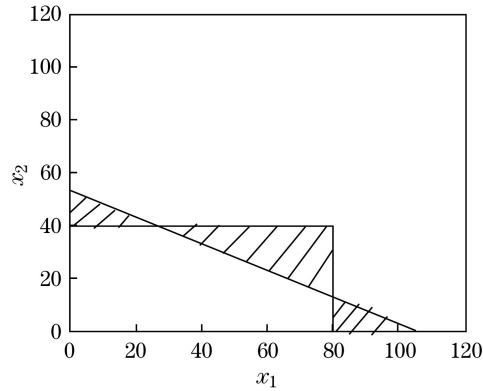


图4 例2中的错误区域

若想提高准确率, 可以根据训练样本的特点, 采用朴素Bayes分类方法. 其过程如下: 考虑训练样本中标+1的特征所形成的向量, 可以认为其在区域 $G_1 = \{(x_1, x_2)^T | 0 \leq x_1 \leq 80, 0 \leq x_2 \leq 40\}$ 上是均匀分布. 其概率密度可以写成如下的式子

$$f_1(X) = \begin{cases} \frac{1}{3200}, & X \in G_1; \\ 0, & X \notin G_1. \end{cases}$$

又标+1的训练样本个数占总样本各数的比例为 $\pi_1 = 0.23$.

再考虑训练样本中标-1的特征所形成的向量, 可以认为其在区域 $G_2 = (K \setminus G_1) \cup \{x_1 = 80, 0 \leq x_2 \leq 40\} \cup \{x_1 = 40, 0 \leq x_2 \leq 80\}$ 上是均匀分布.

其概率密度可以写成如下的式子

$$f_{-1}(X) = \begin{cases} \frac{1}{11200}, & X \in G_2; \\ 0, & X \notin G_2. \end{cases}$$

又标-1的训练样本个数占总样本各数的比例为 $\pi_{-1} = 0.77$.

于是构造如下的分类机器:

$$h(X) = \operatorname{argmax}_{k \in \{-1, 1\}} \{\pi_k f_k(X)\}$$

其准确率可以达到100%. 可见采用不同的机器, 得到的系统近似生存核精度不同. 所以可以采用一些常见的机器对其进行分类, 比较结果并用最好的机器来得到近似生存核.

5 结论和展望

本文研究了一般离散系统生存核的逼近算法. 主要利用了机器学习的方法, 给出了近似的生存核, 证明了在一定条件下, 此方法得到的生存核是收敛于真实的生存核. 通过

数值例子, 说明此算法极大地简化了计算量. 此算法的另一优点是具体的分类机器可以根据具体题目来选择. 最后, 给出了两个具体的算例. 可以预见, 机器学习的方法将来在微分对策和系统可达性等控制领域将有较广泛的应用.

参考文献

- [1] Aubin J P. *Viability theory* [M]. Boston: Birkhauser, 1991.
- [2] Gao Y, Lygeros J, Quincampoix M, et al. On the control uncertain impulse system: approximate stabilization and controlled invariance [J]. *International Journal of Control*, 2004, **77**: 1393-1407.
- [3] Limon D, Alamo T, Camacho E F. Enlarging the domain of attraction of MPC controllers [J]. *Automation*, 2005, **41**: 629-635.
- [4] Aubin J P, Bayen A, Patrick S P. *Viability theory new directions* [M]. Princeton: Springer-Verlag, 2011.
- [5] 陈征, 高岩. 人口控制的混杂模型与应用 [J]. *系统管理学报*, 2009, **18**: 107-110.
- [6] Bene C, Doyen L, Gabay D. A viability analysis for a bio-economic model [J]. *Ecological economics*, 2001, **36**: 385-396.
- [7] 高岩. 一类非线性控制系统关于非光滑区域生存性的判别 [J]. *控制与决策*, 2006, **21**: 923-925.
- [8] 蒋卫华, 黄琳, 楚天广. 离散非线性时变凸多面体系统族的鲁棒正不变集 [J]. *自动化学报*, 2001, **27**: 631-636.
- [9] Rougé C, Mathias J D, Deffuant G. Extending the viability theory framework of resilience to uncertain dynamics, and application to lake eutrophication [J]. *Ecological Indicators*, 2013, **29**: 420-433.
- [10] Bernard C, Martin S. Building strategies to ensure language coexistence in presence of bilingualism [J]. *Applied Mathematics and Computation*, 2012, **218**: 8825-8841.
- [11] Sicard M, Perrot N, Reuillon R, et al. A viability approach to control food processes: Application to a Camembert cheese ripening process [J]. *Food Control*, 2012, **23**: 312-319.
- [12] Botkin N D, Ryazantseva E A. Structure of viability kernels for some linear differential games [J]. *Journal of optimization theory and applications*, 2010, **147**: 42-57.
- [13] Maggiore M, Rawn B, Lehn P. Invariance kernels of single-input planar nonlinear systems [J]. *SIAM Journal on Control and Optimization*, 2012, **50**: 1012-1037.
- [14] Bonneuil N. Computing the viability kernel in large state dimension [J]. *Journal of Mathematical Analysis and Applications*, 2006, **323**: 1444-1454.
- [15] Doyena L, Martinet V. Maximin, viability and sustainability [J]. *Journal of Economic Dynamics and Control*, 2012, **36**: 1414-1430.
- [16] Rakovic S V, Kerrigan E C, Mayne D Q, et al. Reachability analysis of discrete-time systems with disturbances [J]. *IEEE Transactions on Automatic Control*, 2006, **51**: 546-560.
- [17] 高岩. 非光滑优化 [M]. 北京: 科学出版社, 2008.
- [18] Hastie T, Tibshirani R, Friedman J. *The elements of statistical learning: data mining, Inference, and prediction* [M]. Princeton: Springer-Verlag, 2009.
- [19] Jin W, Zhang J Q, Zhang X. Face recognition method based on support vector machine and particle swarm optimization [J]. *Expert Systems with Applications*, 2011, **38**: 4390-4393.