第 30 卷第 3 期
2013 年 3 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 30 No. 3
Mar. 2013

DOI: 10.7641/CTA.2013.20835

利用条件概率和Gibbs抽样技术为分布估计算法构造通用概率模型

张　放†, 鲁华祥

(中国科学院 半导体研究所 神经网络实验室, 北京 100083)

摘要: 本文针对传统分布估计算法在建立概率模型时面临的各种困难, 提出一种基于条件概率和Gibbs抽样的概率模型, 能有效改进分布估计算法的通用性. 使用该模型的分布估计算法利用进化过程中有前途的优秀个体构造出多个监督学习样本集, 并对每个样本集估计出对应分量的条件概率, 再使用这一组条件概率进行Gibbs抽样产生新的个体替代种群中的劣等个体. 通过仿真实验表明, 改进后的算法能够求解出可加性降解函数的全局最优解, 表现出较强的全局优化能力.

关键词: 分布估计算法; Gibbs抽样; 分类; 监督学习

中图分类号: TP181　　**文献标识码**: A

General stochastic model for algorithm of distribution estimation with conditional probabilities and Gibbs sampling

ZHANG Fang†, LU Hua-xiang

(Lab of Neural Networks, Institute of Semiconductor, Chinese Academy of Science, Beijing 100083, China)

Abstract: A stochastic model based on conditional probability and Gibbs sampling is proposed to cope with the modeling problems occurred in traditional algorithms for distribution estimation, and extends the generality of the algorithm. The algorithm with this model takes promised individuals in the evolution process to form supervised training sets. For each of such sets, we estimate the conditional probability of a component given other components, and execute a Gibbs sampling procedure to generate new candidates for replacing inferior ones. The result of computer experiments shows that the improved algorithm can obtain the global optimum of additively decomposed functions, demonstrating a strong ability in global optimization.

Key words: estimation of distribution algorithm; Gibbs sampling; classification; supervised learning

1　引言(Introduction)

最近几年, 在进化计算(evolutionary computation, evolutionary algorithm)领域兴起了一类新型的优化算法, 称为分布估计算法(estimation of distribution algorithms, EDAs)[1−2], 并迅速成为进化计算领域的研究热点和解决工程问题的有效方法[3−6].

进化算法虽然无法保证获取全局最优, 但可以用有限的代价获取足够好的解, 或称准最优解(quasi-optimum solution). 进化算法更重要的特性在于, 对于任何优化目标, 理论上该算法都能以“黑盒”的方式进行求解, 也就是说这是一种纯粹的“通用算法”, 能够求解传统方法难以解决的复杂优化问题或者黑盒问题. 然而, 作为进化算法主要代表的遗传算法(genetic algorithm, GA)却很难达到这个目标, 其求解效果经常受到目标函数形式的限制. 分布估计算法是在遗传算法的基础上提出的, 与遗传算法类似, 其基本思想也是用种群来表示最优化问题的一组候选解, 然后将待优化的目标函数作为评价函数, 来指导解种群的进化, 从而启发式的探索解空间, 最终以有限的代价获取准最优解. 分布估计算法与遗传算法最重要的不同之处在于种群的更新方式: 遗传算法对种群中的优秀个体使用“交叉”、“变异”等遗传算子来产生新的个体, 从而替代劣等个体; 分布估计算法则对种群中的优秀个体建立分布模型, 并从模型中随机抽样来产生新的个体并替代劣等个体. 在构建新个体的过程中, 由于使用了优秀个体集合的全局信息而非仅仅来自父个体的局部信息, 分布估计算法不像遗传算法那样轻易的破坏建

收稿日期: 2012−07−30; 收修改稿日期: 2012−10−25.

†通信作者. Tel.: +86 15120094198.

基金项目: 国家自然科学基金资助项目(61076014); 江苏省高校自然科学基金资助项目(10KJA510042); 中科院战略性先导科技专项资金资助项目(XDA06020700).

筑块(building blocks, BBs)[7], 从而能够更为高效的探索解空间.

为了表征优秀个体的分布, 分布估计算法提出了一个抽象的“概率模型”, 这个模型能够通过学习过程由样本集来生成, 也可以通过采样过程来产生新的样本集, 如图1所示.

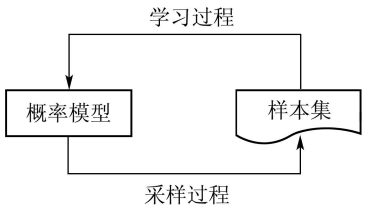

图 1 概率模型

Fig. 1 Stochastic model

分布估计算法的核心在于如何建立合适的概率模型, 并给出相应学习算法和采样算法. 从最早的被认为是EDAs雏形的基于种群的增量学习算法(population-based incremental learning, PBIL)[8]的出现到今天, 已经有大量的概率模型被提出以用于分布估计算法, 按照处理变量间相互作用的能力, 这些模型可以分为: 一阶模型, 包括PBIL[8]和一元边缘分布算法(univariate marginal distribution algorithm, UMDA)[9–10]等, 这些模型假设各分量之间互相独立, 所以联合分布函数可以表示为各分量的边缘分布函数的乘积, 从而可以对每个分量分别进行学习与采样; 二阶模型, 包括基于互信息最大化的输入聚类算法(mutual-information-maximizing input clustering, MIMIC)[11]和二元边缘分布算法(bivariate marginal distribution algorithm, BMDA)[12–13]等, 这类模型假设各分量之间的依赖关系可以表示为一个或多个树形结构, 学习过程分为两个步骤, 首先根据样本集寻找最优依赖树, 然后通过依赖树和样本集计算概率和条件概率, 采样过程则是根据依赖树依次进行采样; 高阶模型, 包括降解分布算法(factorized distribution algorithm, FDA)[14]和贝叶斯优化算法(Bayesian optimization algorithm, BOA)[15]等, 这一类模型通过某种结构来表示变量之间的一般性的依赖关系, 学习过程也分为寻求最优结构和计算条件概率这两个步骤, 采样过程也是根据依赖关系依次采样.

前两类模型由于对联合概率分布加入了额外的限制条件, 所以并不能解决一般性的优化问题; 第三类模型理论上可以解决一般性的优化问题, 但实际的求解过程又受到各方面的限制, 例如FDA的分解模型需要领域知识专家人为的指定[14], 而BOA依赖的贝叶斯网络的学习则是一个非确定性多项式难度(non-deterministic polynomial hard, NP-Hard)问题[16], 实际的求解只能通过各种非常技巧化的手段来进行[15,17]. 所以说, EDAs经过这么长时间的发展, 对于一般化优化问题的求解, 仍然没有较为理想的通用概率模型.

本文提出了一种基于条件概率和Gibbs抽样[18]的概率模型, 简称CGS(conditional probability and Gibbs sampling)模型, 通过对条件概率分布而非联合概率分布进行建模, 克服了传统概率模型面临的各种困难, 从而能够更容易的对一般性的分布进行学习和采样. 后面的内容将组织如下: 第2节将介绍CGS模型的原理及使用该模型的EDAs的工作流程, 第3节将构造一个实际的CGS模型, 第4节将使用该模型与几种传统的分布估计算法进行对比, 考察这些优化算法对于不同阶数的目标函数的优化效果.

2 CGS模型(CGS model)

对优秀个体的分布建立概率模型, 最大的困难在以下两个方面:

1) 受问题的维度限制. 维度越高, 意味着变量之间的耦合关系越错综复杂. 对于高耦合的分布, 不仅难以建立精确的概率模型, 而且即使建立了模型, 学习过程和采样过程也非常难于进行;

2) 学习的无监督性. 根据样本集求分布是一个典型的无监督学习; 而在机器学习领域, 有监督学习的丰富程度、操作难度、结果精确性等方面都远胜于无监督学习.

为了克服以上两个困难, 本文提出了一种基于条件概率和Gibbs抽样的概率模型, 其基本思想是, 对于高维随机变量$\boldsymbol{X}$的联合概率分布$f_{\boldsymbol{X}}(\boldsymbol{x})$, 直接建模是非常困难的, 所以退而求其次, 对条件概率分布$f_{X_i|\boldsymbol{X}^{(i)}}(x_i|\boldsymbol{x}^{(i)})$(即其他分量$\boldsymbol{X}^{(i)}=\boldsymbol{x}^{(i)}$的条件下$X_i=x_i$的概率)进行建模, 如图2所示.

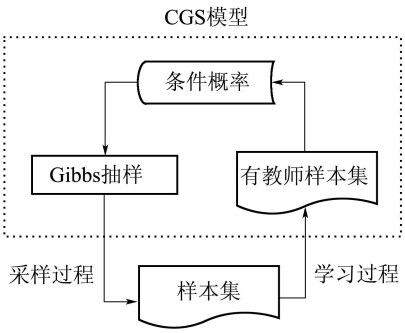

图 2 CGS模型

Fig. 2 CGS model

对于学习过程, 可以通过两个子步骤来实现:

1) 监督化样本集: 将其中一个分量作为期望输出, 从而将非监督样本集$\mathscr{D}$解释为监督学习样本

集$\mathscr{T}$;

2) 估计条件概率: 使用监督学习样本集来估计条件概率$f_{X_i|\boldsymbol{X}^{(i)}}(x_i|\boldsymbol{x}^{(i)})$的值.

对于抽样过程, 则可以通过一种叫做Gibbs抽样(Gibbs sampling)[18]的技术, 利用条件概率$f_{X_i|\boldsymbol{X}^{(i)}}(x_i|\boldsymbol{x}^{(i)})$来依次更新各个分量, 从而构造出一个马尔科夫链, 使得链上状态的分布逐渐的逼近联合概率分布$f_{\boldsymbol{X}}(\boldsymbol{x})$. 再通过对马尔科夫链上状态适当的选择, 便可实现抽样过程.

本文中使用的各种符号的意义约定如表1所示, 其中小写表示随机变量的具体取值.

表 1 符号约定

Table 1 Symbols convention

符号	意义
D	样本向量的维度
N	样本集的大小
$\boldsymbol{S}$	样本空间
$\boldsymbol{S}_i$	样本第i个分量构成的子空间
$\boldsymbol{S}^{(i)}$	样本第i个分量构成的子空间的补空间
$\boldsymbol{X}/\boldsymbol{x}$	样本向量
X_i/x_i	样本向量的第i个分量
$\boldsymbol{X}^{(i)}/\boldsymbol{x}^{(i)}$	样本除第i个分量外的其他分量构成的向量
$\boldsymbol{x}(k)$	样本集中的第k个样本

2.1 抽样过程(Sampling process)

Gibbs抽样是一种从多个随机变量的联合分布中抽取样本的算法, 该算法是马尔科夫链蒙特卡洛(Markov chain Monte Carlo, MCMC)算法[19]的一个实例, 非常适合于以下抽样场景: 多个随机变量的联合分布很难精确表达或很难直接从中抽样, 但是每个变量对其他变量的条件概率已知而且容易对该变量单独进行抽样[18].

Gibbs抽样算法轮流为每个变量随机产生一个新的实例, 产生的依据是该变量的条件概率和其他变量当前的值. 可以证明, 通过这种方法产生的样本序列构成了一个马尔科夫链, 其稳态分布就是这些随机变量的联合概率分布[20]. 也就是说对于任意统计量$\phi(\boldsymbol{X})$, 可以用时间平均来替代统计平均:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\phi(\boldsymbol{X}(t))=\mathrm{E}\{\phi(\boldsymbol{X})\}. \tag{1}$$

这表示可以从生成的时间序列中适当的选择一些样本, 使得构成的样本集与直接从联合分布中采样得到的样本集的性质完全相同.

由于生成的样本集将在下一轮的学习过程中使用, 为了避免引入额外的相关性, 不能采纳在马尔科夫链上相距太近的状态作为样本集, 这是因为相距时间Δt的两个状态最多只有Δt个变量是不同的, 所以对于较小的Δt, 这两个状态存在大量共同的模式, 其中绝大部分都对评价值并没有贡献. 在本文中取$\Delta t=D$, 也就是说只有每一个分量都进行了一次新的抽样(相对于上一个采纳的状态), 才采纳当前状态作为样本.

另外, 马尔科夫链需要一定的时间才能达到稳态分布, 越靠前的状态受初始状态的影响越大, 所以必须舍弃初始K_0轮的状态. 在本文中取$K_0=D^2$, 即舍弃前D^3个状态以保证马尔科夫链达到稳态分布.

综上, 本文中使用的Gibbs抽样算法($\mathscr{P}=\mathrm{Sample}(N,\{\mathrm{P}(X_i|\boldsymbol{X}^{(i)}\})$)如下:

1) 初始化:

a) $\mathscr{P}=\phi$;

b) $\boldsymbol{x}(0)\sim\mathrm{Uniform}(\boldsymbol{S})$.

2) 迭代: for $t=1$ to K_0+N:

a) 抽样: for $i=1$ to D:

$$\begin{aligned}x_i(t)\sim\mathrm{P}(X_i|&x_1(t),x_2(t),\cdots,\\&x_{i-1}(t),x_{i+1}(t-1),\cdots,\\&x_D(t-1));\end{aligned}$$

b) 输出: if $t>K_0$ then $\mathscr{P}=\mathscr{P}\cup\{\boldsymbol{x}(t)\}$.

整个抽样算法的时间复杂度为$\mathrm{O}((D^2+N)\cdot D)$.

2.2 学习过程(Learning process)

学习过程就是通过有监督样本集

$$\mathscr{T}_i=\{(\boldsymbol{x}^{(i)},x_i)|\boldsymbol{x}\in\mathscr{P}\} \tag{2}$$

来估计条件概率$f_{X_i|\boldsymbol{X}^{(i)}}(x_i|\boldsymbol{x}^{(i)})$的过程. 其中($\boldsymbol{x}^{(i)}$, x_i)表示样本$\boldsymbol{x}^{(i)}$的期望输出为x_i. 为了简化讨论, 本文假设$X_i\in\{-1,+1\}$.

为了估计条件概率$f_{X_i|\boldsymbol{X}^{(i)}}(x_i|\boldsymbol{x}^{(i)})$, 可以分两步进行:

1) 使用一些已有的分类算法和$\mathscr{T}_i$训练得到分类器;

2) 根据分类器来估计条件概率.

针对一类典型的分类算法$\mathscr{A}$, 这里给出一种通用的估计方法. $\mathscr{A}$中分类算法的特点是, 它们通过(显式的或隐式的)求解一个最优化, 计算出分类界面的显式表达式, 即

$$g=\arg\min_{g\in\mathscr{G}}\{\sum_{\mathscr{T}}L(y,g(\boldsymbol{x}))\}, \tag{3}$$

其中: $\mathscr{T}=\{(\boldsymbol{x},y)\}$为一组有监督样本集, $\mathscr{G}$为某个预先指定的候选函数集, 而$L(y,g)$称之为损失函数, 用来表征对不理想分类结果的惩罚. 分类界面$\mathscr{S}$由方程$g(\boldsymbol{x})=0$来表达.

对于$\mathscr{A}$中的分类算法, 可以使用如下方式估计条件概率:

$$\hat{\mathrm{P}}(Y|\boldsymbol{X})\sim\exp[-L(Y,g(\boldsymbol{X}))]. \tag{4}$$

这种估计仅依赖于分类算法的损失函数. 使用这种形式的条件概率能保证通过分类算法计算得到的分

类界面函数$g(\cdot)$能够最大化似然函数, 即式(3)等价于

$$g = \arg\max_{g\in\mathscr{G}}\{\prod_{(\boldsymbol{x},y)\in\mathscr{T}}\hat{\mathrm{P}}(y|\boldsymbol{x})\}. \tag{5}$$

通常各种分类算法不会直接求解式(3)来获取$g(\cdot)$, 第3节将给出一个实际求解的例子.

2.3 CGS–EDA算法流程(Routine of CGS–EDA)

使用CGS模型的EDAs算法(本文称之为CGS–EDA), 其算法$((\boldsymbol{x}_{\mathrm{opt}}, \mathrm{fit}_{\mathrm{opt}}) = \mathrm{CGS-EDA(eval)})$伪代码如下所示, 其中eval表示评价函数, 也就是待优化的函数:

1) 初始化:
   a) $\mathscr{P} = \{\boldsymbol{x}(i)|i = 1, 2, \cdots, N\}$, $\boldsymbol{x}(i) \sim \mathrm{Uniform}(\boldsymbol{S})$;
   b) $\mathscr{P}_{\mathrm{opt}} = \phi$.
2) 迭代: for $ep = 1$ to Epoch:
   a) 择优: for $i = 1$ to N_{o}:
      i) $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathscr{P}}\{\mathrm{eval}(\boldsymbol{x})\}$;
      ii) $\mathscr{P}_{\mathrm{opt}} = \mathscr{P}_{\mathrm{opt}} \cup \{\boldsymbol{x}^*\}$;
      iii) $\mathscr{P} = \mathscr{P}\backslash\{\boldsymbol{x}^*\}$.
   b) 学习: for $i = 1$ to D:
      i) $\mathscr{T}_i = \{(\boldsymbol{x}^{(i)}, x_i)|\boldsymbol{x} \in \mathscr{P}_{\mathrm{opt}}\}$;
      ii) $g_i = \arg\min_{g\in G}\{\sum_{\mathscr{T}_i} L(y, g(\boldsymbol{x}))\}$;
      iii) $\hat{\mathrm{P}}(X_i|\boldsymbol{X}^{(i)}) \sim \mathrm{e}^{-L(X_i, g(\boldsymbol{X}^{(i)}))}$.
   c) 采样:

$$\mathscr{P}_{\mathrm{new}} = \mathrm{Sample}(N - N_{\mathrm{o}}, \{\hat{\mathrm{P}}(X_i|\boldsymbol{X}^{(i)})\}).$$

   d) 更新: $\mathscr{P} = \mathscr{P}_{\mathrm{opt}} \cup \mathscr{P}_{\mathrm{new}}$, $\mathscr{P}_{\mathrm{opt}} = \phi$.
3) 输出:
   a) $\boldsymbol{x}_{\mathrm{opt}} = \arg\max_{\boldsymbol{x}\in\mathscr{P}}\{\mathrm{eval}(\boldsymbol{x})\}$;
   b) $\mathrm{fit}_{\mathrm{opt}} = \mathrm{eval}(\boldsymbol{x}_{\mathrm{opt}})$.

这里使用“截断”的方式进行更新, 即评价值最小的那一部分个体被直接淘汰, 并用概率模型抽样产生的新样本代替. $(N - N_{\mathrm{o}})/N$称之为更新过程的淘汰比率.

3 配置CGS模型(Configure CGS model)

一个CGS模型可以配置的部分包括两个: 分类器和分类算法. 其中分类器表征了CGS模型的结构复杂度, 而分类算法则对应于式(3)的求解方式. 本节给出一个模型实例, 使用的分类器为线性分类器, 分类算法为AdaBoost[21].

线性分类器指如下参数模型:

$$g(\boldsymbol{x}; \boldsymbol{\beta}, \theta) = \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x} + \theta. \tag{6}$$

为了方便表达偏置θ, 增加一个恒定为+1的分量x_D, 从而将$\boldsymbol{x}$由$D-1$维扩展到D维, 则线性分类器可以更简洁的表示为

$$g(\boldsymbol{x}; \boldsymbol{\beta}) = \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}, \tag{7}$$

其中β_D即表示偏置θ.

使用的分类算法为AdaBoost, 这种算法为每个样本赋予一个权重, 并用指定的弱分类器进行分类, 再根据分类结果提高那些被错误分类的样本的权重, 然后进行下一轮迭代, 最后对获得的每一个弱分类器根据分类效果进行线性加权, 得到最终的分类器. 可以证明[22], 通过这种方式获得的分类器, 等价于使用贪心算法对式(3)进行求解, 其中损失函数为指数函数

$$L(y, g) = \exp(-yg). \tag{8}$$

在这里, 为了最终获得线性分类器, 本文相对应的AdaBoost算法$(g(\boldsymbol{x}) = \mathrm{AdaBoost}(\mathscr{T}))$如下:

1) 权重化:

$$\mathscr{T}^* := \{(\boldsymbol{x}(i), y(i), w_i) \mid (\boldsymbol{x}(i), y(i)) \in \mathscr{T}\}.$$

2) 初始化: $w_i = 1/N$, $i = 1, 2, \cdots, N$.
3) 迭代: for $m = 1$ to M:
   a) $(k_m, s_m) = \arg\min_{\substack{k\in\{1,2,\cdots,D\}\\ s\in\{-1,+1\}}}\{\sum_{(\boldsymbol{x},y,w)\in\mathscr{T}^*} s\cdot x_k y\cdot w\}$;
   b) $\mathrm{err}_m = \sum_{i=1}^{N} w_i I(y_i \neq s_m x_{k_m}) / \sum_{i=1}^{N} w_i$;
   c) $\alpha_m = \log((1 - \mathrm{err}_m)/\mathrm{err}_m)$;
   d) for $i = 1$ to N:
      $w_i \leftarrow w_i \cdot \exp[\alpha_m I(y_i \neq s_m x_{k_m})]$.
4) 输出: $g(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m s_m x_{k_m}$.

算法中使用的弱分类器为单分量分类器, 即

$$b(\boldsymbol{x}; k, s) = s \cdot x_k, \tag{9}$$

其求解过程

$$(k, s) = \arg\min_{\substack{k\in\{1,2,\cdots,D\}\\ s\in\{-1,+1\}}}\{\sum_{(\boldsymbol{x},y,w)\in\mathscr{T}^*} s\cdot x_k y\cdot w\} \tag{10}$$

的意义为: 求出与y相关性(正相关或负相关的绝对值)最高的一个分量x_i, 并乘以符号系数s($s \in \{-1, +1\}$, 表示x_i与y是正相关还是负相关)作为分类指标.

至于AdaBoost算法中的迭代次数M, 本文选择为D, 一方面保证每个分量都至少有一次可能的参与分类器作用的机会, 另一方面限制迭代数量过多, 因为这一迭代过程嵌套在外层EDAs框架的迭代过

程之中. 为了加快训练算法的寻优过程, 当$\alpha_m D<1$时, 即认为当前的弱分类器的作用已经可以忽略不计, 并退出迭代过程, 直接输出分类器. 每轮迭代的时间复杂度为$\mathrm{O}(D^2N)$.

4　计算机实验(Computer experiments)

为了测试CGS模型的效果, 本节使用2.5节中给出的CGS–EDA算法以及第3节中给出的模型配置, 与几个典型的分布估计算法进行对比, 包括一阶模型UMDA[9–10]、二阶模型BMDA[12–13]以及高阶模型BOA[15]. 作为测试的目标函数为可加性降解函数[10], 即

$$F(\boldsymbol{x})=\sum_{i=1}^{L} f_i(\boldsymbol{z}_i), \tag{11}$$

其中$\boldsymbol{z}_i$是$\boldsymbol{x}$的所有分量$\{x_i|i=1,2,\cdots,D\}$的一个子集组成的向量. 不同的$\boldsymbol{z}_i$可以选择重叠的, 即同时包含某个分量x_i. 这一类函数可以降解为更简单的子问题, 每个子问题对应着原问题的一个建筑块. 对于精心构造的具有欺骗性的函数$f_i(\cdot)$, 低阶的概率模型将无法有效的捕捉到这些建筑块; 而随着$\boldsymbol{z}_i$维数的增加, 即使高阶的概率模型也将面临极大的困难.

对于子函数$f_i(\cdot)$, 下文将给出几种典型的函数, 对每种函数来说, 其自变量的维数D_{s}都不相同. 本文按顺序选择前D_{s}个未使用的x_i来构造子串$\boldsymbol{z}_i$. $\boldsymbol{x}$的维度D则选择为60. 为了方便描述$f_i(\cdot)$以及和其他几种算法保持一致, 在本节中, 令$\boldsymbol{S}=\{0,1\}$, 所以在使用CGS–EDA进行优化之前, 需要进行一个简单的变换将自变量变换至空间$\{-1,+1\}$:

$$X_i'=2X_i-1. \tag{12}$$

下面将使用相同参数的UMDA, BMDA, BOA以及CGS–EDA分别优化目标函数, 并重复100次, 比较输出的准最优值的分布情况.

优化器的参数设置为: 最大迭代次数100, 种群大小1024, 淘汰比率0.5. 由于实验中选取的子问题的维度最大不超过6, 即子问题的变量空间尺寸不超过64, 对于能有效进行降解的算法来说, 1024的种群大小足以在其他变量的干扰下完整的探索子问题的变量空间; 迭代次数的选择为100的原因在于, 本文假定有效的算法在每次迭代中平均分离出的建筑块数目的数量级不小于10^{-1}. 使用该组参数, 算法覆盖到的变量空间为$100\times1024\approx10^5$, 只占总变量空间的$1/10^{13}$, 这种比例对于暴力搜索来说太小了, 无法有效实现寻优.

下面分别给出5种不同的目标函数的优化结果和结果分析.

4.1　Onemax函数(Onemax function)

Onemax函数统计自变量中1的个数, 即

$$f_{\mathrm{onemax}}(\boldsymbol{x})=\sum_{i=1}^{D} x_i. \tag{13}$$

这个函数是典型的“分量无关”型函数, 即每个分量独立的为总的函数值做贡献. 该函数在所有变量取1时取到最大值D, 并且没有局部极小点. 4种优化器的优化结果如图3所示.

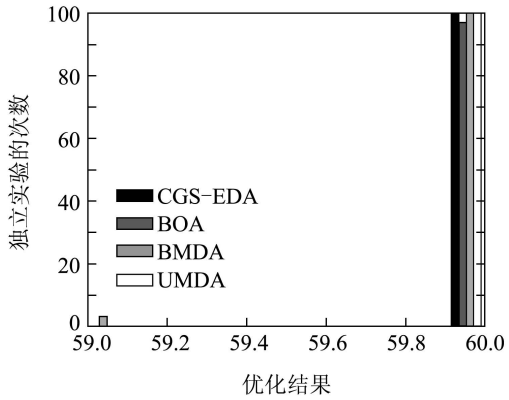

图 3　Onemax优化结果

Fig. 3　Optimization result on onemax

对于这种分量无关的函数, 所有优化算法的表现都很好, 几乎所有100次实验都得到了全局最优值60.

4.2　Quadratic函数(Quadratic function)

Quadratic函数的定义如下:

$$f_{\mathrm{quadratic}}(\boldsymbol{x})=\sum_{i=1}^{L} f_2(\boldsymbol{z}_i), \tag{14}$$

其中$f_2(\cdot)$是一个双变量的二次函数:

$$f_2(z_1,z_2)=0.9-0.9(z_1+z_2)+1.9z_1z_2, \tag{15}$$

$f_2(\boldsymbol{z})$的取值情况为如图4所示.

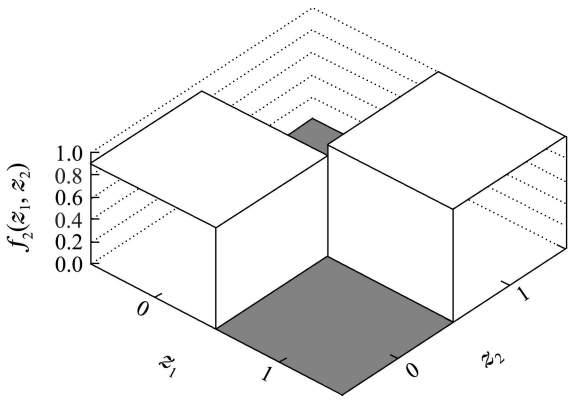

图 4　f_2取值分布情况

Fig. 4　Value distribution of f_2

f_2包含一个最优点和一个次优点, 且分布位置基本对称, 最优点的吸引域略大于次优点.

显然, $f_{\text{quadratic}}(\boldsymbol{x})$在所有变量取1时取到最大值$D/2$, 同时还含有一些局部极小点, 这些局部极小点对应着某些$\boldsymbol{z}_i = (0,0)$ 的情况. 几种优化算法的优化结果如图5所示.

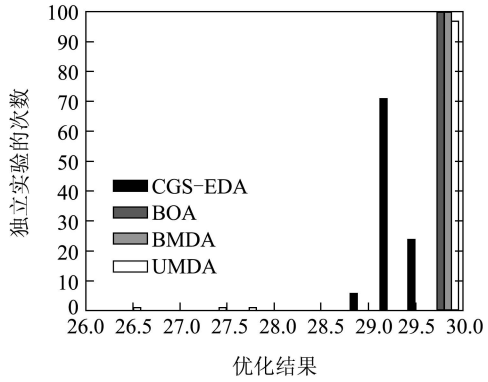

图 5 Quadratic优化结果

Fig. 5 Optimization result on quadratic

由优化结果可以看出, BMDA和BOA非常善于解决此类问题, 能够有效捕捉到全局最优; UMDA表现也非常好, 但是有一小部分实验的结果比较差, 靠近最差的局部极大值27, 甚至有一部分还低于27. 相比于这几个算法, CGS–EDA的表现中庸的多, 绝大部分实验都未能得到全局最优, 但是所有解都比较靠近全局最优, 没有太差的结果.

4.3 Deceptive–3函数(Deceptive-3 function)

Deceptive–3函数的定义如下:

$$f_{\text{deceptive}-3}(\boldsymbol{x}) = \sum_{i=1}^{L} f_3(\boldsymbol{z}_i), \tag{16}$$

其中$f_3(\boldsymbol{z})$只与$\boldsymbol{z}$中1的个数u有关:

$$f_3(\boldsymbol{z}) = \begin{cases} 0.9, & u = 0, \\ 0.8, & u = 1, \\ 0, & u = 2, \\ 1, & u = 3. \end{cases} \tag{17}$$

这里的$\boldsymbol{z}$为3维向量, 函数值分布如图6所示.

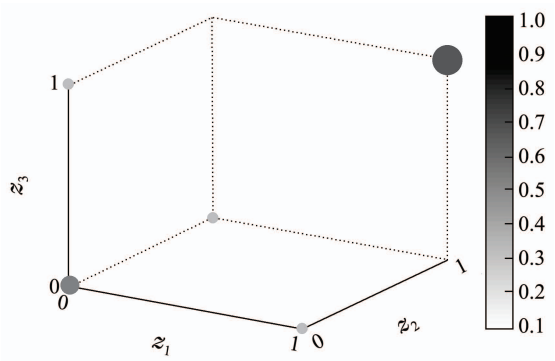

图 6 f_3取值分布情况

Fig. 6 Value distribution of f_3

当$\boldsymbol{z} = (1,1,1)$时, $f_3(\boldsymbol{z})$取到最大值1; 当$\boldsymbol{z} = (0,0,0)$时, $f_3(\boldsymbol{z})$取到次大值0.9. 次优点$(0,0,0)$的吸引域比最优点$(1,1,1)$的吸引域要大, 这是因为次优点的邻近点(含有1个1的点)的评价值比最优点的邻近点(含有2个1的点)更大, 所以次优点的邻近点更容易在进化过程中被保留从而引导种群向次优点发展, 而最优点的邻近点则被淘汰从而阻隔了向最优点探索的路径.

相应地, $f_{\text{deceptive}-3}(\boldsymbol{x})$在所有变量取1时取到全局最优值$D/3$, 另外包含许多局部极大点, 这些局部极大点均对应着某些$\boldsymbol{z}_i = (0,0,0)$的情况. 几种优化算法的优化结果如图7所示.

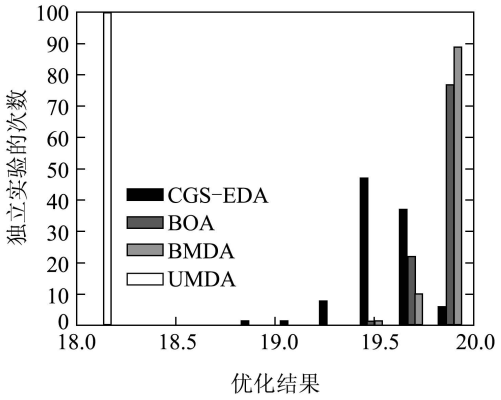

图 7 Deceptive–3优化结果

Fig. 7 Optimization result on deceptive–3

由优化结果可以看出, UMDA对于该问题已经完全无能为力, 全部100次实验的结果都得到最差的局部极大值18; BMDA和BOA的效果依然非常好, 但相比于quadratic函数的实验结果, 性能开始有些下降; 相比于这两个算法, CGS–EDA的表现依然比较中庸, 绝大部分实验都未能得到全局最优, 但是所有解都比较靠近全局最优, 没有太差的结果.

4.4 Trap–5函数(Trap–5 function)

Trap–5函数的定义如下:

$$f_{\text{Trap}-5}(\boldsymbol{x}) = \sum_{i=1}^{L} f_5(\boldsymbol{z}_i), \tag{18}$$

其中$f_5(\boldsymbol{z})$只与$\boldsymbol{z}$中1的个数u有关:

$$f_5(\boldsymbol{z}) = \begin{cases} 5-u, & u < 5, \\ 5, & \text{其他}. \end{cases} \tag{19}$$

这里的$\boldsymbol{z}$为5维向量, 其函数值分布情况与图6类似, 但是处于一个5维的超立方体中. 当$\boldsymbol{z}$的所有分量全为1时, $f_5(\boldsymbol{z})$取到最大值5; 当$\boldsymbol{z}$ 所有分量全为0时, $f_5(\boldsymbol{z})$取到次大值4. 与f_3类似, f_5的次优点的吸引域比最优点的吸引域要大, 因为次优点的邻近点(含有1个1)以及次邻近点(含有2个1)的评价值比最优点

的邻近点(含有4个1)和次邻近点(含有3个1)的评价值更大, 所以次优点的附近的值比最优点附近的值更容易在进化过程中被保留并吸引种群的进化方向.

相应地, $f_{\mathrm{Trap-5}}(\boldsymbol{x})$在所有变量取1时取到全局最优值$D$, 另外包含许多局部极小点, 这些局部极小点均对应着某些$\boldsymbol{z}_i$为全0向量的情况. 几种优化算法的优化结果如图8所示.

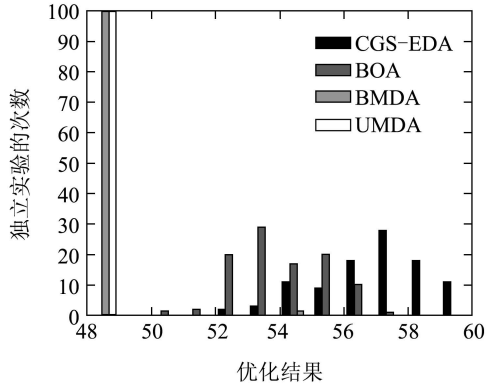

图 8 Trap–5优化结果

Fig. 8 Optimization result on trap–5

由优化结果可以看出, UMDA对于更高阶的问题仍旧无能为力, 全部100次实验的结果都得到最差的局部极大值48; BMDA和BOA的效果则出现大幅度的下降, 其中BMDA绝大部分实验都仅得到最差的局部极大值48, 而BOA的结果则分布在48和全局最优值60之间; 相比于这几个算法, CGS–EDA的表现则稳定得多, 所有结果仍然聚集在全局最优附近, 尽管取得全局最优的比例仍然很少, 但是作为对比的其他算法则一次都没有取得全局最优.

4.5 Trap–6 函数(Trap–6 function)

Trap–6函数的定义与Trap–5类似, 只是$\boldsymbol{z}_i$使用6维向量来代替. 函数性质方面也与trap–5类似, 只是变量间的耦合程度更高. 几种优化算法的优化结果如图9所示.

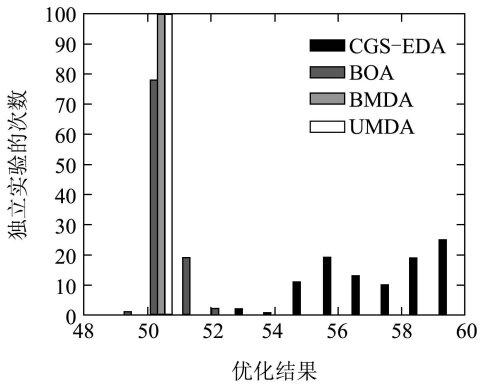

图 9 Trap–6优化结果

Fig. 9 Optimization result on Trap–6

由优化结果可以看出, CGS–EDA取得压倒性的结果, 其他3种优化算法已经几乎完全陷入最差的局部最优, 而CGS–EDA的结果仍旧十分稳定.

4.6 结果分析(Analysis on results)

表2给出了对于不同测试函数, 各种分布估计算法在100次实验中的最好结果和平均结果. 为了便于对比不同的测试函数, 这些值都进行了归一化处理, 归一化方式如下:

$$\mathrm{fit}_{\mathrm{norm}} = \frac{\mathrm{fit}_{\mathrm{pratical}} - \mathrm{fit}_{\mathrm{local-opt}}}{\mathrm{fit}_{\mathrm{global-opt}} - \mathrm{fit}_{\mathrm{local-opt}}}, \tag{20}$$

其中: $\mathrm{fit}_{\mathrm{pratical}}$为优化算法实际计算得到的结果, $\mathrm{fit}_{\mathrm{global-opt}}$和$\mathrm{fit}_{\mathrm{local-opt}}$分别表示全局最优点(即全1向量)和评价值最小的一个局部最优点(即全0向量)的评价值.

几种传统的分布估计算法在解决变量轻度耦合的问题时表现非常出色, 相对来说CGS–EDA的结果较为中庸, 获得全局最优的比例并不多, 但是所有实验的结果都在全局最优附近, 没有特别差的结果.

表 2 优化结果对比

Table 2 contrast on results of optimizations

测试函数	CGS–EDA		BOA		BMDA		UMDA	
	最好	平均	最好	平均	最好	平均	最好	平均
onemax	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
quadratic	0.867	0.749	1.000	0.999	1.000	1.000	1.000	0.955
deceptive–3	0.950	0.808	1.000	0.953	1.000	0.972	0.100	0.014
Trap–5	1.000	0.791	0.833	0.553	0.583	0.038	0.000	0.000
Trap–6	1.000	0.771	0.200	0.022	0.000	0.000	0.000	0.000

随着子函数自变量维度的增加, 几种传统的分布估计算法的效果逐渐降低以至于几乎完全陷入局部极小, 而CGS–EDA依然表现出稳定的性能, 如图10所示.

对于所有这些测试函数, CGS–EDA得到的解都有规律的分布在全局最优解附近, 说明相较于传统算法, 该算法在处理一般性的优化问题上性能更为优秀.

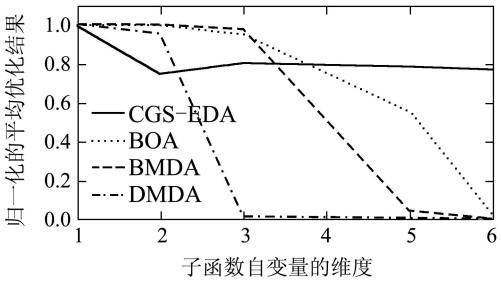

图 10 优化结果对比

Fig. 10 Contrast on results of optimizations

为了理解这一结果, 需要引入模型复杂度的概念[23]. 算法的性能很大程度上取决于抽样得到的新样本集的质量, 而这取决于模型的预测能力. 模型越简单, 就越难精确逼近真实的概率分布, 这导致预测出现较高的“偏倚”; 模型越复杂, 虽然逼近能力更强, 但带来了更高的结构风险, 这导致预测包含较高的“方差”. 而最终的预测误差是由偏倚和方差两部分共同决定的, 如图11所示.

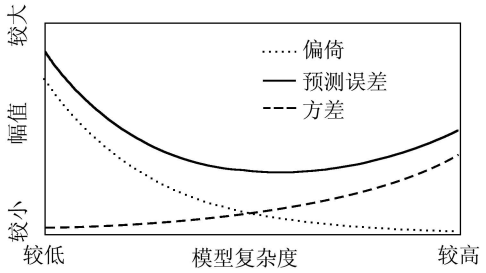

图 11 预测误差与模型复杂度

Fig. 11 Prediction error and model complexity

对于UMDA和BMDA来说, 由于模型自身的限制(仅对局部建模), 只能描述较轻的耦合关系, 随着子问题维度的增加, 这两种模型的偏倚越来越大, 导致最终的预测结果价值不大. BOA由于引入了贝叶斯网络来描述一般性的依赖关系, 理论上可以对任意分布建模, 但是模型复杂度太高, 导致数据集中的噪声(来自其他子问题的干扰)很容易被错误的耦合进模型, 导致预测结果不可靠.

CGS–EDA在这些问题上表现好的原因, 很大程度在于条件概率分布的引入. 一方面, 条件概率分布$f_{X_i|\boldsymbol{X}^{(i)}}(x_i|\boldsymbol{x}^{(i)})$在建模时使用了全局信息(即表达式的值取决于所有变量的值), 另一方面, 由于使用了D个条件概率分布, 整个联合概率分布$f_{\boldsymbol{X}}(\boldsymbol{x})$的复杂度分散到了每个条件概率分布之中, 导致这里可以使用较为简单的模型(例如本文中使用的线性模型)来对条件概率分布建模, 从而降低了模型的结构风险. 例如对于f_3, f_5, f_6, 都能用如下函数来近似描述其条件概率的分布:

$$\hat{f}_{X_i|\boldsymbol{X}^{(i)}}(x_i|\boldsymbol{x}^{(i)}) = \frac{1}{1+\mathrm{e}^{(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^{(i)}-D+0.5)(2x_i-1)}}. \tag{21}$$

当其他分量不全为1时, x_i倾向于取0; 当其他分量全为1时, x_i倾向于取1. 而该分布对应的分类界面就是一个线性函数

$$g(\boldsymbol{x}^{(i)}) = \mathbf{1}^{\mathrm{T}}\boldsymbol{x}^{(i)} - D + 0.5. \tag{22}$$

当然, 存在着更复杂的优化问题, 使得条件概率分布的形式更复杂, 以至于线性模型太粗糙而效果不佳. 这种情况下, 可以通过配置CGS模型, 使用更复杂的分类器(例如二次分类函数等)来降低偏倚, 从而在偏倚和方差之间形成新的折衷, 最大程度的限制算法性能的衰退; 而其他几种算法在这种情况下则无可避免的面临性能的急剧恶化.

5 结论(Conclusions)

计算机实验结果表明, 在求解可加性降解函数的最优化问题时, 使用本文提出的基于CGS模型的分布估计算法在有限轮迭代的过程中能够得到较好的准最优解, 并且随着子问题耦合程度的增加, 算法的表现依然稳定, 这说明CGS模型显著提高了分布估计算法的通用程度.

参考文献(References):

[1] LARRANAGA P, LOZANO J A. *Estimation of Distribution Algorithms: a New Tool for Evolutionary Computation* [M]. Boston: Kluwer Academic Publishers, 2002.

[2] 周树德, 孙增圻. 分布估计算法综述 [J]. 自动化学报, 2007, 33(2): 113 – 124.
(ZHOU Shude, SUN Zengqi. A survey on estimation of distribution algorithms [J]. *Acta Automatica Sinica*, 2007, 33(2): 113 – 124.)

[3] SHIM V A, TAN K C, CHEONG C Y. A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem [J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2012, 42(5): 682 – 691.

[4] ARMANANZAS R, SAEYS Y, INZA I, et al. Peakbin selection in mass spectrometry data using a consensus approach with estimation of distribution algorithms [J]. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2011, 8(3): 760 – 774.

[5] 窦丽华, 王高鹏, 陈杰, 等. 求解弹头散布均匀度的分布估计算法 [J]. 控制理论与应用, 2009, 26(6): 624 – 628.

(DOU Lihua, WANG Gaopeng, CHEN Jie, et al. A hybrid algorithm for computing cannonball dispersion evenness [J]. *Control Theory & Applications*, 2009, 26(6): 624 – 628.)

[6] 王圣尧, 王凌, 许烨, 等. 求解混合流水车间调度问题的分布估计算法 [J]. 自动化学报, 2012, 38(3): 437 – 443.
(WANG Shengyao, WANG Ling, XU Ye, et al. An estimation of distribution algorithm for solving hybrid flow-shop scheduling problem [J]. *Acta Automatica Sinica*, 2012, 38(3): 437 – 443.)

[7] GOLDBERG D E, KORB B, DEB K. Messy genetic algorithms: Motivation, analysis, and first results [J]. *Complex System*, 1989, 3(5): 493 – 530.

[8] BALUJA S. *Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning, technical report CMU–CS–94–163* [R]. Pittsburgh, PA: Carnegie Mellon University, 1994.

[9] MUHLENBEIN H, PAASS G. From recombination of genes to the estimation of distributions, I: binary parameters [C] //*Parallel Problem Solving from Nature–PPSN IV*. Heidelberg: Springer, 1996: 178 – 187.

[10] MUHLENBEIN H. The equation for response to selection and its use for prediction [J]. *Evolutionary Computation*, 1997, 5(3): 303 – 346.

[11] DEBONET J S, ISBELL C L, VIOLA P. MIMIC: finding optima by estimating probability densities [M] //*Advances in Neural Information Processing Systems*. Cambridge: MIT Press, 1997, 9: 424 – 430

[12] PELIKAN M, MÜHLENBEIN H. The bivariate marginal distribution algorithm [M] //*Advances in Soft Computing – Engineering Desgin and Manufacturing*. London: Springer-Verlag, 1993: 521 – 535.

[13] PELIKAN M, MUEHLENBEIN H. Marginal distributions in evolutionary algorithms [C] //*International Conference on Genetic Algorithms*. Brno, Czech Republic: Technical University of Brno, 1998: 90 – 95.

[14] MUEHLENBEIN H, MAHNIG T. FDA: a scalable evolutionary algorithm for the optimization of additively decomposed functions [J]. *Evolutionary Computation*, 1999, 7(4): 353 – 376.

[15] PELIKAN M, GOLDBERG D E, CANTU-PAZ E. BOA: the Bayesian optimization algorithm [C] //*Proceedings of the Genetic and Evolutionary Computation Conference*. Orlando, Florida: Morgan Kaufmann, 1999: 525 – 532

[16] CHICKERING D M, GEIGER D, HECKERMAN D. *Learning Bayesian networks is NP-hard, Technical Report MSR–TR–94–17* [R]. Redmond, Washington: Microsoft Research, Microsoft Corporation, 1994.

[17] CHENG J, GREINER R, KELLY J, et al. Learning Bayesian networks from data: an information-theory based approach [J]. *Artificial Intelligence*, 2002, 137(1/2): 43 – 90.

[18] GEMAN S, GEMAN D. Stochastic relaxation, Gibbs Distributions, and the Bayesian restoration of images [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984, 6(6): 721 – 741.

[19] BERG B A. *Markov Chain Monte Carlo Simulations and Their Statistical Analysis* [M]. Singapore: World Scientific, 2004.

[20] GELMAN A, CARLIN J B, STERN H S, et al. *Bayesian Data Analysis* [M]. London: Chapman and Hall, 1995.

[21] FREUND Y, SCHAPIRE R E. A decision-theoretic generalization of on−line learning and an application to boosting [J]. *Journal of Computer and System Sciences*, 1997, 55(1): 119 – 139.

[22] SCHAPIRE R E, FREUND Y, BARTLETT P, et al. Boosting the margin: a new explanation for the effectiveness of voting methods [J]. *Annals of Statistics*, 1998, 26(5): 1651 – 1686

[23] VAPNIK V N. *The Nature of Statistical Learning Theory* [M]. 2nd edition. New York: Springer, 1999: 80 – 82.

作者简介:

张　放　(1985−), 男, 博士研究生, 目前研究方向为机器学习, E-mail: zhangfang08@mails.gucas.ac.cn;

鲁华祥　(1965−), 男, 研究员, 目前研究方向为智能信息处理和人工智能, E-mail: luhx@semi.ac.cn.