基于遗传算法的多阶马氏链组合预测方法[1]

程乾生[2] 王守章 武连文
(北京大学数学科学学院金融数学系)

【摘要】给出遗传算法的一种改进形式:增加了"引进算子",采用了"混合编码",并提出多阶马氏链组合预测方法.通过二者的结合,给出了一种股市行情数据预测的方法.

关键词:预测,遗传算法,高阶多重马氏链,证券市场
分类号:F832.5

0 引 言

遗传算法[1]是由美国Michigan大学的John Holland创建的,1975年他出版了*Adaptation in Natural and Artificial Systems*一书阐述了遗传算法的基本理论.达尔文进化理论认为,自然界中的生物都是由低等的原始生物通过进化、自然选择得到的.现代生物学研究表明,自然界中形形色色、千差万别的各种形态的高等生物都是从原始的单细胞生物进化而来的,其不同的外部表现是由其内在的遗传物质的不同决定的.现代遗传学认为,生物的遗传物质存在于细胞核内部的染色体上,染色体由代表遗传性质的基本片段——基因组成,基因品种及其排列顺序的不同决定了生物体的不同表现.这种不同具体地表现为生物体适应自然环境的能力的不同.并且现代高级生命体的产生,相对于生命体在地球上出现的漫长岁月,是非常短暂的.另外,在自然进化过程中,因为有性繁殖保证了遗传物质在其后代中的混合和重组,比起那些仅有单个父本的无性繁殖仅靠变异来产生新的后代,进化要快得多.

自然界有性繁殖生物的进化过程具有以下特征:

1) 进化过程发生在染色体上.

2) 适应性好的个体的染色体经常比差的个体的染色体有更多的繁殖机会.

3) 繁殖过程是进化发生的那一刻.

4) 生物进化没有记忆.

有性繁殖生命体的进化过程,仅由简单的复制、杂交和变异就能够快速地演化出对自然界具有高度适应能力的生命体,而并不需要任何其他信息,这足以说明进化方法的高效性和稳健性(robustness).

遗传算法其实是一种概率搜索算法.它是利用某种编码技术作用于称为染色体的二进制数串,其基本思想是模拟由这些串组成的群体的进化过程.遗传算法通过有组织地然而是随机地信息交换来重新组合那些适应性好的串,并保留到下一代(选择).在每一代中,利用上一代串结构中适应性好的位和段来生成一个新的串的群体(杂交,从上一代中以某一概率 p_c 选取个体进行杂交);作为额外的添加,偶尔也要在串结构中尝试用新的位和段来代替原来的部分(变异,在新的一代中的个体以 p_m 的概率进行变异,变异的位置是等概率的).遗传算法是一类随机算法,但它不是简单的随机走动,它可以有效地利用已有的信息(染色体代表的个体所具有的适应值)来搜索那些有希望改善解的质量的串(染色体).类似

① 国家自然科学基金资助项目(69872003).
② 程乾生:教授,博士生导师.通讯地址:北京大学数学科学学院.邮编:100871.电话:010-62752752. Fax:010-62751801
E-mail:qcheng@.pku.edu.cn.

于自然界的进化，遗传算法通过作用于染色体上的基因，寻求"好"的基因的组合（串、染色体）来求解优化问题. 与自然界相似，遗传算法对求解问题的本身并不需要任何信息（它不受搜索空间的有限制性的假设约束，不必要求诸如对解的连续、光滑及单峰等的假设），它所需要的仅是对算法所产生的每个串（染色体）进行评价——给出适应值（Fitness），然后基于串的适应值来"选择"染色体，使适应性好的染色体比适应性差的染色体有更多的繁殖机会（在下一代中出现）. 遗传算法能够快速有效地搜索高度复杂、高度非线性的多维空间.

遗传算法是从模拟自然界生物进化机制的研究中产生出来的，它依据优胜劣汰的生物进化规则，能够在较大的参数空间中较快地搜索到问题的最优解. 它是一种自适应启发式群体型迭代式全局搜索的算法，在解决非线性问题时具有全局收敛性. 遗传算法的收敛性由下面的定理保证：

定理（收敛性定理）[3].

如果在群体的演化过程中，遗传算法总是保留每代中最好的解，并且算法以杂交和变异作为随机化算子，则对于一个全局优化问题，随着演化代数趋于无穷，遗传算法将以概率 1 找到全局最优解.

遗传算法实现的关键在于问题编码的表示方法，即确定表示方案，然后确定适应值的度量，再就是要确定一些遗传算法的参数和变量及给出结果的表示方法和搜索停止的准则. 在完成上述准备工作以后，可以执行下面的基本遗传算法：

1）随机产生一个由确定长度的特征串组成的初始群体.

2）对串群体迭代地执行下面的 i、ii 两步，直到满足停止准则：

i　计算群体中每个个体的适应值；

ii　应用复制、杂交和变异算子产生下一个代.

3）把最后一代中出现的最"好"的个体指定为遗传算法的执行结果.

基本遗传算法（simple genetic algorithms，SGA）的编码是基于二进制数的，如下式所示：

$$a_1 a_2 \cdots a_n \tag{1}$$

式（1）中 a_i 的取值为 0 或 1 表示组成染色体的基因，并称上式为一个染色体，n 表示染色体的长度.

首先要构造一个初始群体 $\{A_1, A_2, \cdots, A_m\}$，$m$ 为群体的规模，在整个计算过程中不变. 每个个体（染色体）A_i 都是形如式（1）的二进制串. 一般初始群体可以凭经验选取，也可以随机得到. 当然无论如何选取，都必须保证群体的个体对于实际问题来说要有意义.

SGA 包括复制、杂交和变异 3 种算子：

SGA 的复制算子使得上一代中的优秀（适应值较大的）个体按一定的概率保留到下一代中去，这样就能够使得群体素质不断改进，一般采用适应值较大的个体复制的概率越大的原则.

SGA 的杂交算子是按一定的概率（由个体的适应值大小所决定），选择两个较好的串，并在某一位置上进行片段交换（一点杂交）.

如染色体 $A = a_1 a_2 a_i a_{i+1} \cdots a_n$ 和染色体 $B = b_1 b_2 b_i b_{i+1} \cdots b_n$ 在 i（$1 < i$ 且 $i < n$）处进行一点杂交

$$a_1 a_2 \cdots a_i a_{i+1} \cdots a_n, b_1 b_2 \cdots b_i b_{i+1} \cdots b_n$$

得到

$$a_1 a_2 \cdots a_i b_{i+1} \cdots b_n, \ b_1 b_2 \cdots b_i a_{i+1} \cdots a_n$$

杂交算子试图从优秀个体的组合中得到更好的新个体，因为可以认为优秀的个体是由优秀的基因片段组成的，通过优秀染色体的重组有可能把优秀的基因片段组合到一块，进而产生更加优秀的个体.

SGA 的变异算子即在染色体上的某个基因取反，即 0 变成 1，1 变成 0. 杂交算子能够产生新的个体.

算法停止的依据一般以得到的较优个体的适应值为准（也就是说，如果得到了满意的个体就停止搜索），也可以限制进化的代数.

1 改进的遗传算法

在SGA 的基础上，提出“引进算子”和“混合编码”.

1.1 引进算子

在SGA 的搜索过程中由于缺乏对问题本身的有关知识的应用，一般来说速度比较慢. 并且由于对问题本身理解的不够全面、彻底，在选择初始群体时基本上是盲目的，这种盲目性会直接导致初始群体中个体仅具有局部性，给搜索全局最优解带来困难 —— 至少要等到变异产生出有代表性的、新的个体，这必然会使搜索时间大幅度延长. 据此，在原算法的基础上，提出下述增补的遗传算法.

希望群体在进化（复制、杂交和变异）的同时，每代中都“引进”一部分新的个体（正如现实生活中优良品种的引进一样，当然这里只是随机地引进，因而引进的未必是优良“品种”），也就是说，在保留前一代进化得到的好的个体的条件下，并不排除新个体的加入. 这样，由于对问题理解不足而决定群体的规模较小、或者选择的初始群体比较片面时，在群体的进化过程中也不会因为选取的群体缺乏代表性，进而引起仅在局部空间搜索，使得最优解短期内不能达到的问题. 因为在每一代中都要补充新的个体，这就使得群体在进化过程中更能表现出全局搜索的能力.

在改进的算法中，对于每一代群体，都要对适应值最差的那部分个体进行替换，也就是说，用“引进”的新的个体来代替. 至于引进遗传算法的收敛性问题，可以把“引进”的新个体看成是由原个体变异的极端情况，因而并不破坏遗传算法的收敛性. 实践证明，引进遗传算法将增进收敛的速度.

1.2 混合编码

传统的简单遗传算法的编码是基于二进制的编码，这种编码容易产生和操作，并且在理论上也比较容易处理. 但在实际的应用中，有时由于计算时精度或编码方便方面的考虑，二进制编码可能并不适用，因此考虑在遗传算法的编码中采用浮点数表示法[1].

$$x_i^{(l)} = (v_1^{(l)}, v_2^{(l)}, \cdots, v_k^{(l)}, \cdots, v_r^{(l)}) \quad i = 1,2,\cdots,N \quad l = 1,2,\cdots,GEN \tag{2}$$

如式(2) 所示，把每个基因用一个浮点数 $v_k^{(l)}$ 表示，用一个向量 $x_i^{(l)}$ 来表示一个染色体. 在式(2) 中 l 表示 $x_i^{(l)}$ 所在群体的代数，i 表示 $x_i^{(l)}$ 是群体中的 i 个个体，k 表示个体 $x_i^{(l)}$ 的第 k 个基因，N 表示群体的规模，GEN 表示遗传算法将要进化的最大代数.

在传统简单遗传算法中的算子都是基于二进制的串的处理，新的编码方式要求遗传算法中的算子对浮点数组成的向量进行处理. 这时，并不排除简单遗传算法中的一点或多点杂交，因为这些杂交算子并不依赖二进制的编码表示，只是要求作用的对象是一种序列的形式，所以，它们同样可以用于浮点数表示的个体上，只是杂交的位置只能选择在各个分量之间处；变异算子仍为随机修改向量中的某一个分量（即基因变异）.

除了可以采用简单遗传算法中的一点或多点杂交算子外，还可以采用具有浮点数特征的杂交算子，比如：浮点数向量（染色体）$x_i^{(l)}$ 和 $x_j^{(l)}$ 杂交产生下一代中个体 $x_i^{(l+1)}$ 和 $x_j^{(l+1)}$ 时，取下面的杂交算子

$$x_i^{(l+1)} = a^* x_i^{(l)} + (1-a)^* x_j^{(l)} \qquad x_j^{(l+1)} = (1-a)^* x_i^{(l)} + a^* x_j^{(l)}$$

其中 $a \in (0,1)$，a 可以是常数（这时，称之为一致杂交算子），也可以是一个随着代数而变化的变数（这时，称之为非一致杂交算子），甚至，这种算子也可以仅仅作用在向量（染色体）的某些分量（基因）上来作为个体杂交计算的结果.

本文将采用所谓混合编码的技术，即对各个要搜索的参数根据其具体情况进行不同的编码方式，适合于二进制编码的就采用二进制编码，适合于浮点数编码的就采用浮点数编码. 相应地，在遗传算法中的各个算子发生作用时，要依据发生作用的基因的不同性质（是二进制数还是浮点数）的不同而采用不同类型的算子.

2　基于动态串长的高阶多重马氏链预测方法

称一个离散参数随机过程 $\{X(t),t=0,1,\cdots\}$ 或连续参数随机过程$\{X(t),t\geqslant 0\}$ 为马尔可夫过程,如果对于随机过程附标集中的任意 n 个时刻:$t_1<t_2<\cdots<t_n$,当给定 $X(t_1),X(t_2),\cdots X(t_{n-1})$ 时,$X(t_n)$ 的条件分布只依赖于最邻近的已知值 $X(t_{n-1})$,而与 $X(t_1),X(t_2),\cdots,X(t_{n-2})$ 的取值无关.更精确地说,对于任给的实数 $x_1,x_2,\cdots,x_n$,有

$$P[X(t_n)=x_n|X(t_1)=x_1,X(t_{n-1})=x_{n-1}]$$
$$=P[X(t_n)=x_n|X(t_{n-1})=x_{n-1}]$$

随机过程的所有可能值的集合称为该随机过程的状态空间.如果马尔可夫过程的状态空间是有限的或者可数的,则称该马氏过程为马尔可夫链.

称 $P(x,t_0;E,t)$ 为马尔可夫链从状态 $X(t_0)=x$ 到状态 $X(t)=E$ 的转移概率.称马尔可夫过程是具有平稳转移概率的或时齐的,如果 $P(x,t_0;E,t)$ 仅通过 $t-t_0$ 依赖于 t_0 和 t.

考虑一个字符序列 $a_1a_2\cdots a_n\cdots$,产生相连的长度为 r 的一系列字符串块 $a_1a_2\cdots a_r,a_2a_3\cdots a_{r+1},a_3a_4\cdots a_{r+2},\cdots$.一个有 n 个字符组成的序列可以产生 $n-r+1$ 块相连的 r 个字符组成的字符块.由此考虑 r 元组 $(a_1a_2\cdots a_r)$,其中每一个 a_i 仅可以取 γ 种状态,这样,每个字符串块可以取 γ^r 种不同的状态.如果相连的由 r 个字符组成的字符块序列形成一个马尔可夫链.即当给定任一上述字符块序列时,下一块特定状态出现的概率,等于仅知道这串字符块的最后一块的状态时下一块有这一特定状态的概率,这时就称原来的字符序列为 r 阶多重马尔可夫链[4,5].

如果一个序列构成一马氏链,可以依据马氏链的转移概率给出下一状态的预测.进一步说,如果原序列构成一高阶多重马尔可夫链,但具体阶数难以确定,这时就对高阶多重马尔可夫链采用一定的融合的方法把它们组合起来用于预测.具体地可以采用阈值法、最大值法、线性加权法等融合方法,将在下一节的例子中给出各种融合方法的详细描述.

3　在股市行情预测中的应用

3.1　数据准备

现在考虑某股票第 i 时期的数据:

$$P_i=(O_i,H_i,L_i,C_i,N_i,M_i,I_i,Q_i)$$

其中 O_i、H_i、L_i、C_i、N_i、M_i、I_i、Q_i 分别表示相应时期该股票的开盘价、最高价、最低价、收盘价、成交量、成交金额、股市指数和股市成交量.

期望从已知的 $(P_1,P_2,\cdots,P_K)$ 来预测未知的 P_{K+1},并且为了实际可操作性,这里采用时期的单位为天,当然也可以根据需要以半年,月,周,小时,半小时,甚至以分钟为时期单位来实现预测(本文讨论的预测方法对时期的选取和股票品种的选取没有任何限制).

直接通过 $\{P_i\}(i=1,2,\cdots,K)$ 来得到 P_{K+1},当然是人们所感兴趣的,但是为了处理方便,首先考虑 $\{P_i\}$ 的变化百分率 $\{R_i\}$ 序列:

$$R_i=(P_{i+1}-P_i)/P_i*100\%\quad i=1,2,\cdots,K-1$$

然后进一步把 $\{R_i\}$ 离散化 —— 考虑基于状态的预测问题:可以把每天相对前一天的状态分为 3 种:下跌、持平、上涨(或者分为 5 种:大跌、小跌、持平、小涨、大涨),期望由今天和过去的状态得到明天的涨跌状态.

根据股市波动的具体状况,采用一定的百分率 η,如果涨跌的幅度在 $\eta\%$ 以内,则是持平的,记为状态 0;涨幅大于 $\eta\%$,认为是上涨,记为状态 1;跌幅大于 $\eta\%$,则认为是下跌,记为状态 -1(分 5 种状态时,

可以依据大多数人所能接受的百分率η_1、η_0、η_2，当股市涨幅不小于η_1%时，认为是大涨；当涨幅小于η_1%但不小于η_0%时，认为是小涨；当涨跌幅度小于η_0%时，认为是持平；跌幅小于η_0%但大于η_2%时，认为是小跌；跌幅不小于η_2%时，认为是大跌).

以此分别离散化$\{R_i\}$向量中的各个分量，得到预测的入口参数向量S_i

$$S_i = (o_i, h_i, l_i, c_i, n_i, m_i, I_i, q_i) \quad i = 1,2,\cdots,K-1$$

其中o_i、h_i、l_i、c_i、n_i、m_i、I_i、q_i分别表示某种股票第i天的开盘价、最高价、最低价、收市价、成交额、成交量和股市价格指数及其成交量的增长率的离散化. 现在，要讨论的问题是期望由$(S_1, S_2, \ldots, S_{K-1})$得到$S_K$的预测.

技术分析的目的就是从过去数据的变动中分析出数据波动的规律，并用于将来数据的预测和分析. 于是很自然地，希望从过去几天的波动，来预测第2天的波动状况，即预测明天股市相对今天的涨跌情况. 因为本文持有技术分析的基本观点：认为股票市场具有其本身的规律，在没有特别的外界干预的情况下，它将按自己固有的规律波动. 并且，股市的波动是和近期的股市行情密切相关的，而和较久远的行情关系不大，或者说，"近期的波动足以代表它的过去". 因此，可以假定第2天股市的涨跌与X天以前的股市状态基本无关. 也就是说，如果把每天的股市状态排列成一行，进而就可以把它看成是由X元组

$$S = (s_1, s_2, \cdots, s_X)$$

组成的一个马尔可夫链. 即近似地认为原股市状态序列构成一个X阶的高阶多重马尔可夫链. 并且假定由X元组S组成的马氏链还是时齐的，这样可以通过过去的数据到马氏链的转移频率(作为马氏链的转移概率)，基于这一思想对股市进行预测.

但是，X的确定没有任何准则，可以认为X在一定范围内，因而首先假定S构成一多阶多重马氏链，并采用融合的方法来处理这一问题.

为了叙述的方便，今后在具体实现时，仅以开盘价状态的预测为例，给出一些方法的预测结果(其余各分量的预测方法类似，下面以飞乐股份1997年1月1日至9月30日的预测为例).

3.2 组合预测方法

3.2.1 阈值法

给定一个阈值$\varepsilon > 0$，对于当前的状态，取链长为$L = L_0$的状态块(即考虑基于L_0阶以内的高阶多重马氏链为预测依据)，从过去的数据中得到当前串出现后各个状态出现的频率，并取出最大的频率δ_l，若$\varepsilon < \delta_l$则由该种串长的链给出对第2天的状态的预测，否则L减1，并重复以上的过程，直到满足上述条件或者L为0为止. 若L为0，也就是说，没有一种串长能够决定的频率大于ε，这时，拒绝给出预测. 取阈值$\varepsilon = 0.40$，可以得到准确率为59.78%预测.

3.2.2 最大值法

假若取链长为L天以内，以L天内出现频率最大的状态的串作为预测的依据(也就是说，考虑1至L阶的高阶多重马氏链，并以这$L*3$个转移概率中最大的为预测依据)(注：$L*3$表示状态分为3种时的情形，若分为5种状态则为$L*5$)，有准确率为72.22%的预测结果.

3.2.3 加权法

正如前面所论述的一样，认为近期的数据决定现在的数据，并且较近的数据对现在的影响较大，而较久远的数据对现在的影响较小，依据这一观点，可以得到较好的预测结果. 假若取链长为L天以内，以L天内每天各个状态出现频率加权和的最大值的串作为预测的依据(即是把这L种预测方法进行线性融合：以这L个马氏链的各种状态转移概率的线性组合为依据，取转移概率最大的状态为预测状态)：

任选一权数：$R = (1/55, 2/55, 3/55/, 4/55, 5/55, 6/55, 7/55, 8/55, 9/55, 10/55)$，$L$取10，预测准确率为78.65%.

3.2.4 基于遗传算法的权数选择方法

在上节中采用的加权法可以得到较好的预测效果，但是对权值的选取没有给出有效的方法，只能凭

预测者的主观猜测. 现在，期望对于前一节中所论述的预测问题，通过遗传算法搜索那些权数(10 维)的最优值.

1) 传统遗传算法搜索

除了采用浮点数编码，采用的方法和 SGA 相同，杂交算子本文采用基因片段交换的方法. 对于其它的杂交算子，方法基本类似，本文就不讨论了. 搜索出的结果见表 1.

表 1　搜索结果

Generation 0		max_fitness: 79.268								
79.268	0.119	0.153	0.094	0.085	0.097	0.080	0.050	0.045	0.119	0.158
79.268	0.131	0.124	0.111	0.147	0.082	0.15	0.043	0.046	0.013	0.153
78.659	0.055	0.073	0.193	0.012	0.171	0.04	0.044	0.08	0.189	0.144
78.659	0.081	0.017	0.157	0.172	0.026	0.085	0.149	0.061	0.124	0.128
78.659	0.067	0.114	0.041	0.164	0.059	0.171	0.176	0.028	0.116	0.064
78.659	0.165	0.194	0.026	0.044	0.143	0.000	0.022	0.103	0.138	0.164
78.659	0.108	0.016	0.185	0.028	0.093	0.119	0.149	0.05	0.16	0.093
78.659	0.087	0.112	0.017	0.184	0.062	0.146	0.122	0.031	0.142	0.096
78.659	0.071	0.107	0.022	0.059	0.104	0.043	0.191	0.04	0.209	0.154
78.660	0.002	0.07	0.061	0.03	0.069	0.154	0.16	0.115	0.162	0.179
Generation 1		max_fitness: 79.268								
79.268	0.119	0.153	0.094	0.085	0.097	0.08	0.05	0.045	0.119	0.158
79.268	0.131	0.124	0.111	0.147	0.082	0.15	0.043	0.046	0.013	0.153
79.268	0.069	0.109	0.217	0.096	0.069	0.051	0.056	0.035	0.13	0.168
79.268	0.105	0.014	0.018	0.144	0.08	0.105	0.182	0.04	0.155	0.157
79.268	0.163	0.154	0.001	0.013	0.134	0.107	0.07	0.045	0.114	0.2
79.268	0.06	0.29	0.101	0.131	0.102	0.036	0.052	0.057	0.008	0.162
79.268	0.119	0.167	0.102	0.094	0.078	0.087	0.05	0.049	0.07	0.185
79.268	0.109	0.105	0.138	0.034	0.156	0.137	0.002	0.041	0.109	0.17
78.659	0.071	0.107	0.022	0.059	0.104	0.043	0.191	0.04	0.209	0.154
78.659	0.002	0.07	0.061	0.03	0.069	0.154	0.16	0.115	0.162	0.179
Generation 2		max_fitness: 79.268								
79.268	0.119	0.153	0.094	0.085	0.097	0.08	0.05	0.045	0.119	0.158
79.268	0.131	0.124	0.111	0.147	0.082	0.15	0.043	0.046	0.013	0.153
79.268	0.069	0.109	0.217	0.096	0.069	0.051	0.056	0.035	0.13	0.168
79.268	0.105	0.014	0.018	0.144	0.08	0.105	0.182	0.04	0.155	0.157
79.268	0.163	0.154	0.001	0.013	0.134	0.107	0.07	0.045	0.114	0.2
79.268	0.06	0.29	0.101	0.131	0.102	0.036	0.052	0.057	0.008	0.162
79.268	0.119	0.167	0.102	0.094	0.078	0.087	0.05	0.049	0.07	0.185
79.268	0.109	0.105	0.138	0.034	0.156	0.137	0.002	0.041	0.109	0.17
79.268	0.137	0.08	0.029	0.15	0.08	0.121	0.029	0.052	0.149	0.173
79.268	0.072	0.132	0.12	0.092	0.074	0.163	0.126	0.03	0.015	0.176
Generation 3		max_fitness: 79.268								
79.268	0.119	0.153	0.094	0.085	0.097	0.08	0.05	0.045	0.119	0.158
79.268	0.131	0.124	0.111	0.147	0.082	0.15	0.043	0.046	0.013	0.153
79.268	0.069	0.109	0.217	0.096	0.069	0.051	0.056	0.035	0.13	0.168
79.268	0.105	0.014	0.018	0.144	0.08	0.105	0.182	0.04	0.155	0.157
79.268	0.163	0.154	0.001	0.013	0.134	0.107	0.07	0.045	0.114	0.2
79.268	0.06	0.29	0.101	0.131	0.102	0.036	0.052	0.057	0.008	0.162
79.268	0.119	0.167	0.102	0.094	0.078	0.087	0.05	0.049	0.07	0.185
79.268	0.109	0.105	0.138	0.034	0.156	0.137	0.002	0.041	0.109	0.17
79.268	0.137	0.08	0.029	0.15	0.08	0.121	0.029	0.052	0.149	0.173
79.268	0.072	0.132	0.12	0.092	0.074	0.163	0.126	0.03	0.015	0.176
Generation 4		max_fitness: 79.268								
79.268	0.119	0.153	0.094	0.085	0.097	0.08	0.05	0.045	0.119	0.158
79.268	0.131	0.124	0.111	0.147	0.082	0.15	0.043	0.046	0.013	0.153
79.268	0.069	0.109	0.217	0.096	0.069	0.051	0.056	0.035	0.13	0.168
79.268	0.105	0.014	0.018	0.144	0.08	0.105	0.182	0.04	0.155	0.157
79.268	0.163	0.154	0.001	0.013	0.134	0.107	0.07	0.043	0.114	0.2
79.268	0.06	0.29	0.101	0.131	0.102	0.036	0.052	0.057	0.008	0.162
79.268	0.119	0.167	0.102	0.094	0.078	0.087	0.05	0.049	0.07	0.185
79.268	0.109	0.105	0.138	0.034	0.156	0.137	0.002	0.041	0.109	0.17
79.268	0.137	0.08	0.029	0.15	0.08	0.121	0.029	0.052	0.149	0.173
79.268	0.072	0.132	0.12	0.092	0.074	0.163	0.126	0.03	0.015	0.176
Generation 5		max_fitness: 79.268								
79.268	0.119	0.153	0.094	0.085	0.097	0.08	0.05	0.045	0.119	0.158
79.268	0.131	0.124	0.111	0.147	0.082	0.15	0.043	0.046	0.013	0.153
79.268	0.069	0.109	0.217	0.096	0.069	0.051	0.056	0.035	0.13	0.168
79.268	0.105	0.014	0.018	0.144	0.08	0.105	0.182	0.04	0.155	0.157
79.268	0.163	0.154	0.001	0.013	0.134	0.107	0.07	0.045	0.114	0.2

79.268	0.06	0.29	0.101	0.131	0.102	0.036	0.052	0.057	0.008	0.162
79.268	0.119	0.167	0.102	0.094	0.078	0.087	0.05	0.049	0.07	0.185
79.268	0.109	0.105	0.138	0.034	0.156	0.137	0.002	0.041	0.109	0.17
79.268	0.137	0.08	0.029	0.15	0.08	0.121	0.029	0.052	0.149	0.173
79.268	0.072	0.132	0.12	0.092	0.074	0.163	0.126	0.03	0.015	0.176

表中仅列出了前 6 代的数据，每代列出了 10 个个体(10 行)，每行代表一个个体(每行中右边的 10 个数字)，每行中最左边的一个数(即最左边的一列)表示该个体的适应值(在这里，也就是以该个体为权数时的预测准确率)，max_fitness 表示该代群体中个体适应值的最大值(即该代中列在第1行的个体的适应值，为了便于观察，对每代群体中的个体都按其适应值由大到小进行了排序)，后面表2的表示方法相同，不再重复.

2) 改进遗传算法搜索

现在采用增加了"引进算子"的遗传算法，可以得出如下表2的结果.

表2 搜索结果

Generation 0		max_fitness: 79.268								
79.268	0.119	0.153	0.094	0.085	0.097	0.080	0.050	0.045	0.119	0.158
79.268	0.131	0.124	0.111	0.147	0.082	0.150	0.043	0.046	0.013	0.153
78.659	0.055	0.073	0.193	0.012	0.171	0.040	0.044	0.080	0.189	0.144
78.659	0.081	0.017	0.157	0.172	0.026	0.085	0.149	0.061	0.124	0.128
78.659	0.067	0.114	0.041	0.164	0.059	0.171	0.176	0.028	0.116	0.064
78.659	0.165	0.194	0.026	0.044	0.143	0.000	0.022	0.103	0.138	0.164
78.659	0.108	0.016	0.185	0.028	0.093	0.119	0.149	0.050	0.160	0.093
78.659	0.087	0.112	0.017	0.184	0.062	0.146	0.122	0.031	0.142	0.096
78.659	0.071	0.107	0.022	0.059	0.104	0.043	0.191	0.040	0.209	0.154
78.659	0.002	0.070	0.061	0.030	0.069	0.154	0.160	0.115	0.162	0.179
Generation 1		max_fitness: 79.878								
79.878	0.081	0.104	0.178	0.050	0.093	0.194	0.025	0.043	0.056	0.175
79.268	0.131	0.124	0.111	0.147	0.082	0.150	0.043	0.046	0.013	0.153
79.268	0.116	0.110	0.055	0.121	0.109	0.095	0.120	0.038	0.093	0.142
79.268	0.119	0.153	0.094	0.085	0.097	0.080	0.050	0.045	0.119	0.158
79.268	0.171	0.158	0.017	0.061	0.148	0.155	0.026	0.032	0.014	0.218
79.268	0.108	0.166	0.106	0.039	0.035	0.139	0.111	0.041	0.105	0.150
79.268	0.071	0.158	0.047	0.097	0.112	0.110	0.049	0.052	0.133	0.172
79.268	0.048	0.096	0.159	0.123	0.060	0.144	0.036	0.024	0.179	0.132
79.268	0.311	0.281	0.077	0.075	0.060	0.102	0.000	0.003	0.055	0.036
79.268	0.001	0.020	0.098	0.059	0.190	0.192	0.116	0.009	0.162	0.152
Generation 2		max_fitness: 79.878								
79.878	0.081	0.104	0.178	0.050	0.093	0.194	0.025	0.043	0.056	0.175
79.878	0.240	0.085	0.043	0.093	0.120	0.140	0.086	0.015	0.043	0.135
79.268	0.116	0.110	0.055	0.121	0.109	0.095	0.120	0.038	0.093	0.142
79.268	0.119	0.153	0.094	0.085	0.097	0.080	0.050	0.045	0.119	0.158
79.268	0.171	0.158	0.017	0.061	0.148	0.155	0.026	0.032	0.014	0.218
79.268	0.108	0.166	0.106	0.039	0.035	0.139	0.111	0.041	0.105	0.150
79.268	0.071	0.158	0.047	0.097	0.112	0.110	0.049	0.052	0.133	0.172
79.268	0.048	0.096	0.159	0.123	0.060	0.144	0.036	0.024	0.179	0.132
79.268	0.311	0.281	0.077	0.075	0.060	0.102	0.000	0.003	0.055	0.036
79.268	0.001	0.020	0.098	0.059	0.190	0.192	0.116	0.009	0.162	0.152
Generation 3		max_fitness: 79.878								
79.878	0.081	0.104	0.178	0.050	0.093	0.194	0.025	0.043	0.056	0.175
79.878	0.240	0.085	0.043	0.093	0.120	0.140	0.086	0.015	0.043	0.135
79.878	0.094	0.198	0.015	0.107	0.140	0.068	0.000	0.063	0.063	0.252
79.878	0.081	0.112	0.122	0.059	0.060	0.179	0.000	0.052	0.057	0.277
79.878	0.220	0.078	0.163	0.046	0.035	0.087	0.110	0.050	0.052	0.160
79.268	0.108	0.166	0.106	0.039	0.035	0.139	0.111	0.041	0.105	0.150
79.268	0.071	0.158	0.047	0.097	0.112	0.110	0.049	0.052	0.133	0.172
79.268	0.048	0.096	0.159	0.123	0.060	0.144	0.036	0.024	0.179	0.132
79.268	0.311	0.281	0.077	0.075	0.060	0.102	0.000	0.003	0.055	0.036
79.268	0.001	0.020	0.098	0.059	0.190	0.192	0.116	0.009	0.162	0.152
Generation 4		max_fitness: 79.878								
79.878	0.081	0.104	0.178	0.050	0.093	0.194	0.025	0.043	0.056	0.175
79.878	0.240	0.085	0.043	0.093	0.120	0.140	0.086	0.015	0.043	0.135
79.878	0.094	0.198	0.015	0.107	0.140	0.068	0.000	0.063	0.063	0.252
79.878	0.081	0.112	0.122	0.059	0.060	0.179	0.000	0.052	0.057	0.277
79.878	0.220	0.078	0.163	0.046	0.035	0.087	0.110	0.050	0.052	0.160
79.878	0.192	0.154	0.058	0.091	0.052	0.096	0.000	0.055	0.059	0.243
79.878	0.090	0.158	0.170	0.047	0.035	0.183	0.034	0.015	0.049	0.219
79.878	0.092	0.096	0.045	0.105	0.128	0.098	0.124	0.049	0.064	0.198

79.268	0.311	0.281	0.077	0.075	0.060	0.102	0.000	0.003	0.055	0.036
79.268	0.001	0.020	0.098	0.059	0.190	0.192	0.116	0.009	0.162	0.152
Generation 5		max_fitness: 79.878								
79.878	0.081	0.104	0.178	0.050	0.093	0.194	0.025	0.043	0.056	0.175
79.878	0.240	0.085	0.043	0.093	0.120	0.140	0.086	0.015	0.043	0.135
79.878	0.094	0.198	0.015	0.107	0.140	0.068	0.000	0.063	0.063	0.252
79.878	0.051	0.112	0.122	0.059	0.060	0.179	0.000	0.052	0.057	0.277
79.878	0.220	0.078	0.163	0.046	0.035	0.087	0.110	0.050	0.052	0.160
79.878	0.192	0.154	0.058	0.091	0.052	0.096	0.000	0.055	0.059	0.243
79.878	0.090	0.158	0.170	0.047	0.035	0.183	0.034	0.015	0.049	0.219
79.878	0.092	0.096	0.045	0.105	0.128	0.098	0.124	0.049	0.064	0.198
79.878	0.080	0.164	0.097	0.049	0.126	0.137	0.110	0.041	0.049	0.148
79.878	0.078	0.269	0.045	0.089	0.078	0.186	0.000	0.050	0.042	0.165

比较表1和表2,可以发现从相同的初始群体出发,通过基本遗传算法进化6代得到最佳个体的适应值为79.268.即以最佳个体为权数可以得到的预测准确率约为79.268%,而通过增添了引进算子的改进型遗传算法仅仅通过一次进化就得到了最佳适应值为79.878的个体.并且从群体进化的角度可以发现,整个群体素质的提高,通过改进型的遗传算法要比传统的基本遗传算法进化的要快得多.

从表1和表2还可以发现,不同的个体可以拥有相同的适应值(在这里即是不同的权值可以得到相同的预测准确率).

取一适应值最大的个体:R=(0.081, 0.104, 0.178, 0.050, 0.093, 0.194, 0.025 0, 0.043 5, 0.056, 0.175),以此权数可以得到准确率约为79.878 % 的预测.

基于给出的预测算法仅与股票市场的波动数据有关,而并不针对某一种股票.因此可以对上市的其它股票进行试算,也可施用于大势(即股市交易指数)的分析,给出整个股市波动方向的预测.

参 考 文 献

1 Goldbery E. Genetic algorithms in search, optimization & machine learning. MA: Addison-Wesley, 1989

2 Eiben A E, et al. Global Convergence of Genetic Algorithms : a Markov Chain Analysis . Parrel Problem Solving from Nature . Dortmund: FRG . 1990

3 刘　勇等. 非数值并行算法—遗传算法. 北京:科学出版社,1997

4 Anderson T W, Goodman L A. Statistical inference about Markov chains. Ann. Math. Stat., 1957;28:89～109

5 Billingsley P. Statistical methods in Markov chains. Ann. Math. Stat. ,1961;32:12～40

The Combined Prediction Method Based on Genetic Algorithm and Multiple-order Markov Chains

Cheng Qiansheng, Wang Shouzhang, Wu Lianwen

Department of Financial Mathematics, School of Mathematical Science, Peking University

Abstract　In this paper, we discussed a kind of GA's improved form: giving a new operator—import-operator, and a coding method-mixed-coding. At the same time, we discussed Multiple-order Markov Chains in the combined prediction method. With the combining of above two, we presented a prediction method in stock-market.

Keywords:　prediction, Genetic Algorithm, higher-order multiple Markov chains, stock market