

用 Householder 变换的递推 AR 模型化与谱估计算法*

余辉里

(航空航天工业部 634 研究所,北京)

摘要 Householder 变换用于上三角化是基于线性预测误差方程的数据阵。可以证明,由上三角阵的主对角元素便可得到各阶 AR 模型的残差平方和。因此用逐列处理的方法可以构成 AR 模型化与谱估计的递推算法。在大多数情况下,本文的算法不仅给出与协方差算法或修正协方差算法相同的计算结果;而且当计算中存在严重的数值病态问题时,协方差法和修正协方差法无法获得好的 AR 谱估计,而本文的算法则仍然可以获得好的估计。文中给出了典型的计算例子。

关键词 AR 谱估计; Householder 变换; AR 参数;递推算法

一、引言

前向与后向预测误差方程可以分别表为^[1]

$$x(n) = - \sum_{i=1}^p a(i)x(n-i) + e_f(n) \tag{1}$$

$$x(n-p) = - \sum_{i=1}^p a^*(i)x(n+i-p) + e_b(i) \tag{2}$$

式中, $\{x(n), n=1, \dots, N\}$ 是数据序列, $\{a(i), i=1, \dots, p\}$ 是预测滤波器系数, $e_f(n)$ 和 $e_b(n)$ 分别为前向预测误差和后向预测误差, p 是线性预测滤波器的阶次, 上标 '*' 表示复共轭。如果 $e_f(n)$ 是方差为 1 的零均白噪声序列, 则 (1) 式是 p 阶 AR 模型, 而 $\{a(i), i=1, \dots, p\}$ 是 AR 参数。

(1) 和 (2) 式可以表为矩阵形式

$$\mathbf{X}_f \mathbf{A}(p+1) = \mathbf{E}_f(p+1) \tag{3}$$

$$\mathbf{X}_b \mathbf{A}(p+1) = \mathbf{E}_b(p+1) \tag{4}$$

$$\mathbf{X}_f^H \mathbf{X}_f \mathbf{A}(p+1) = \mathbf{X}_f^H \mathbf{E}_f(p+1) \tag{5}$$

$$\mathbf{X}_b^H \mathbf{X}_b \mathbf{A}(p+1) = \mathbf{X}_b^H \mathbf{E}_b(p+1) \tag{6}$$

$$\mathbf{X}_f = \begin{bmatrix} x(p+1) & \cdots & x(1) \\ \vdots & & \vdots \\ x(N) & \cdots & x(N-p) \end{bmatrix}; \quad \mathbf{X}_b = \begin{bmatrix} x^*(1) & \cdots & x^*(p+1) \\ \vdots & & \vdots \\ x^*(N-p) & \cdots & x^*(N) \end{bmatrix}$$

* 1989年4月17日收到, 1990年4月26日修改定稿。

$$\begin{aligned} \mathbf{A}(p+1) &= [1 \ a^*(1) \cdots a^*(p)]^h \\ \mathbf{E}_f(p+1) &= [e_f^*(p+1) \cdots e_f^*(N)]^h \\ \mathbf{E}_b(p+1) &= [e_b^*(p+1) \cdots e_b^*(N)]^h \end{aligned}$$

式中上标 'h' 表示复共轭转置。(5)和(6)式被称为协方差正则方程^[1-3]。Morf^[4]等人首先给出了求解这类方程的快速算法,而后 Marple^[1,5]给出了进一步减少计算量的快速算法。

(3)式和(4)式显然可以组合为

$$\begin{aligned} \begin{bmatrix} \mathbf{X}_f \\ \vdots \\ \mathbf{X}_b \end{bmatrix} \mathbf{A}(p+1) &= \begin{bmatrix} \mathbf{E}_f(p+1) \\ \vdots \\ \mathbf{E}_b(p+1) \end{bmatrix} & (7) \\ (\mathbf{X}_f^h \mathbf{X}_f + \mathbf{X}_b^h \mathbf{X}_b) \mathbf{A}(p+1) &= \mathbf{X}_f^h \mathbf{E}_f(p+1) + \mathbf{X}_b^h \mathbf{E}_b(p+1) & (8) \end{aligned}$$

(8)式称为修正协方差正则方程,可以用文献[1]中的快速算法求解。这种快速算法是基于极小化误差函数 $J(fb) = 0.5(\mathbf{E}_f^h \mathbf{E}_f + \mathbf{E}_b^h \mathbf{E}_b)$, 而不是极小化 $J(f) = \mathbf{E}_f^h \mathbf{E}_f$ 或 $J(b) = \mathbf{E}_b^h \mathbf{E}_b$ 。

高斯消元法、QR 分解法、Householder 算法等都可以用来求解(3)、(4)或(7)式,从而得到 AR 参数估计。但问题是这些算法需要有递推形式以便在模型阶次 p 未知时,可以递推估计 AR 模型阶次及参数。近年来,奇异值分解法(SVD)被用来求解基于线性预测误差方程的 AR 参数估计^[6]。尽管使用 SVD 可以由主奇异值估计出 AR 阶次,但是当 p 未知而用 $L > p$ 代替 p 时,则解向量的维数是 L 而不是 p ,即是 p 阶 AR 参数的一个高维逼近。除非在用 SVD 估计出 AR 阶次 p 后,重新构造数据阵再做一次 SVD 才能得到 AR 参数的 p 维估计。

本文给出一种用 Householder 变换的递推算法,这一算法可以由(3)、(4)和(7)式中的任何一式求得 AR 参数估计。在大多数情况下,可以获得与协方差法(用(3)或(4)式)或修正协方差法(用(7)式)相同的计算结果。但在计算中遇到严重的数值病态问题时,协方差法或修正协方差法会得到负的 $J(f)$ 、 $J(b)$ 值或负的 $J(fb)$ 值,或者算法中使用的递推系数超出 0—1 范围,以致无法得到 AR 谱估计。然而使用本文的算法却仍然可以得到好的估计,这是因为 Householder 变换具有良好的数值稳定性。

二、Householder 变换和最小二乘解

设 \mathbf{X} 是 $n \times m (n \geq m)$ 维复矩阵, \mathbf{A} 是 m 维复向量, \mathbf{B} 和 \mathbf{E} 是 n 维复向量,则矩阵方程

$$\mathbf{X}\mathbf{A} - \mathbf{B} = \mathbf{E} \quad (9)$$

的最小二乘解可以使用 Householder 变换^[7]表为

$$\|\mathbf{E}\|^2 = \|\mathbf{H}\mathbf{E}\|^2 = \|\mathbf{H}(\mathbf{X}\mathbf{A} - \mathbf{B})\|^2$$

式中 'H' 表示 Householder 变换。最小二乘解为

$$\mathbf{R}(m)\mathbf{A} = \mathbf{b}(m) \quad (10)$$

式中 $R(m)$ 是由矩阵 HX 的前 m 行构成的上三角阵, $b(m)$ 是由向量 HB 的前 m 个元素构成的 m 维向量。向量 A 通常称为参数向量, $\|E\|^2$ 是 m 阶模型的误差平方和。通过回代算法由 (10) 式可求出向量 A 。

可以证明^[6], 如定义向量 A :

$$A = [A^h(k):1:0]^h, \quad 0 < k \leq m \quad (11)$$

$$A(k) = [a^*(k), \dots, a^*(1)]^h \quad (12)$$

并且对 X 阵做 Householder 变换, 则上三角阵中第 $(k+1)$ 个对角元的模平方就是 k 阶误差平方和。即如果用 $J(k)$ 表示 k 阶误差平方和, 则

$$J(k) = |d(k+1)|^2, \quad 0 \leq k < m \quad (13)$$

式中 $d(k+1)$ 是 $(k+1)$ 行上的对角元素。显然如果 $J(k)$ 用于判定阶次 $k=p < m$, 则解向量 $A(k)$ 与 X 阵的后 $(m-k-1)$ 列无关, 因此可以省略对这些列的 H 变换, 而使用逐列 H 变换法加上判阶来进一步减小计算量。这就是本文提出的 Householder 变换递推算法。

设 $x_{i,j}$ 是原 X 阵的第 i 行、第 j 列元素, $x(k)_{i,j}$ 是第 k 次 H 变换后的 X 阵的第 i 行、第 j 列元素。则由 H 变换的特点可以得到如下递推算法:

$$\left. \begin{aligned} x(0)_{i,j} &= x_{i,j} \\ d(k) &= x(k)_{k,k} = y(k-1)_{k,k} \left| \sum_{i=k}^n x(k-1)_{i,k} x^*(k-1)_{i,k} \right| \\ y(k-1)_{k,k} &= \begin{cases} -x(k-1)_{k,k} / |x(k-1)_{k,k}|, & x(k-1)_{k,k} \neq 0 \\ 1, & x(k-1)_{k,k} = 0 \end{cases} \\ D(k) &= d(k) - x(k-1)_{k,k} \\ |F(k)|^2 &= D(k)d^*(k) \\ G_j(k) &= \sum_{i=k+1}^n x^*(k-1)_{i,k} x(k-1)_{i,j} - D^*(k)x(k-1)_{k,j} \\ C_j(k) &= -G_j(k) / |F(k)|^2 \\ x(k)_{k,i} &= x(k-1)_{k,i} + D(k)C_j(k) \\ x(k)_{i,j} &= x(k-1)_{i,j} - C_j(k)x(k-1)_{i,k} \\ i &= 2, \dots, m; \quad k = 1, \dots, j-1; \quad i = k+1, \dots, n \end{aligned} \right\} \quad (14)$$

因此 (14) 式用于逐列上三角化 X 阵, 再加上阶次判定就可以构成 Householder 变换的递推算法。

三、AR 模型化和谱估计

在大多数应用中, AR 模型的阶次 p 是未知的, 要用一个数 $L \geq p$ 代替 (1) 和 (2) 式中的 p 。令 $n = L+1, \dots, N$, (3), (4) 和 (7) 式重写为

$$X_j(L+1)A(L+1) = E_j(L+1) \quad (15)$$

$$X_b(L+1)A(L+1) = E_b(L+1) \quad (16)$$

$$\begin{bmatrix} \mathbf{X}_f(L+1) \\ \vdots \\ \mathbf{X}_b(L+1) \end{bmatrix} \mathbf{A}(L+1) = \begin{bmatrix} \mathbf{E}_f(L+1) \\ \vdots \\ \mathbf{E}_b(L+1) \end{bmatrix} \quad (17)$$

$$\mathbf{X}_f(L+1) = \begin{bmatrix} x(1) \cdots x(L+1) \\ \vdots \\ x(N-L) \cdots x(N) \end{bmatrix}; \quad \mathbf{X}_b(L+1) = \begin{bmatrix} x^*(L+1) \cdots x^*(1) \\ \vdots \\ x^*(N) \cdots x^*(N-L) \end{bmatrix}$$

$$\mathbf{A}(L+1) = [\mathbf{A}^h(k); 1; 0]^h, \quad \mathbf{A}(k) = [a^*(k) \cdots a^*(1)]^h$$

$$\mathbf{E}_f(L+1) = [e_f^*(L+1) \cdots e_f^*(N)]^h, \quad \mathbf{E}_b(L+1) = [e_b(L+1) \cdots e_b(N)]^h$$

式中 $0 < k \leq L$, 取决于 AR 模型的阶次。显然递推算法(14)式用于(15)–(17)式并结合 AR 阶次选择就可以构成 AR 模型化和谱估计的递推算法。下面讨论三个相关问题。

1. AR 阶次判定

很多判阶准则例如 FPE、AIC、CIT 和 MDL^[1] 均可根据误差平方和的信息(又称线性预测方差^[1])来判定 AR 阶次。但是这些准则都需要使用各阶方差(从阶次 1 到阶次 L)来判定合适的 AR 阶次。在这种情况下, 本文的递推算法失去了使用价值。

本文根据误差函数(误差平方和)的特性^[9]使用另一种简单而有效的判阶方法。文献[9]证明当数据长度 N 趋于无穷时, 误差平方和的下降将在 k 小于真实阶次 p 时显著停滞。在实际应用中, 尽管数据长度是有限的, 由本文算法所得的误差平方和也将出现明显的停滞状态。因此本文使用如下判阶方法:

$$(1) \text{ 计算 } D1(k) = |d(k+1)|/|d(k+2)|; \quad D2(k) = D1(k-1) * D1(k)$$

$$(2) \text{ 比较是否 } FH < D1(k-1) < FL; \quad FH < D1(k) < FL; \quad FH < D2(k) < FL$$

否: 令 $k = k + 1$, 回到步骤(1)

是: $p = k + 1$

式中 FH 和 FL 是判定 $D1(k-1)$, $D1(k)$ 和 $D2(k)$ 近似相等的容限。本文后面部分的仿真计算中将给出相应的判阶结果。

2. AR 参数和谱估计

对于 N 点数据, 如果阶次判定为 p , 则数据阵中只有前 $(p+1)$ 列被用于计算(参看(15)–(17)式)。这等价于只使用 $(N-L+p)$ 点数据。本文后面的仿真计算部分将举例说明所丢失的数据并没有对参数估计的精度造成严重的影响。一旦数据阵((15)–(17)式中的任一个)是上三角阵并判定了 AR 阶次, 便可以通过回代法解出 AR 参数估计。AR 谱估计为

$$P(f) = \frac{Q(p)T}{\left| 1 + \sum_{i=1}^p a(i) \exp(-j2\pi f iT) \right|^2} \quad (18)$$

式中 T 为采样间隔; $\{a(i), i = 1, \dots, p\}$ 可以是(15)–(17)式中任何一式的解; $Q(p)$ 是白噪声方差的估计,

$$\text{对(15)和(16)式, } Q(p) = J(p)/(N-L+p) \quad (19)$$

或

$$\text{对 (17) 式, } Q(p) = J(p)/[2(N - L + p)] \quad (20)$$

3. 与协方差法和修正协方差法的关系

设 (15)—(17) 式中 $L = p$, 则 (3)、(4) 和 (7) 式分别等于 (15)—(17) 式, 只是参数向量和数据阵的表达方式略有区别。这意味着在 AR 阶次已知而且数值病态问题不严重的情况下, 本文的算法会得到与协方差法或修正协方差法相同的计算结果。

如果 AR 阶次未知并且数值病态问题不严重, 则由于 $p < L$ 而只用了 $(N - L + p)$ 点数据, 本文的算法虽不同于协方差法或修正协方差法, 但当 N 大于 L 很多时, 计算结果是相当接近的。没有被使用的 $(L - p + 1)$ 点数据还可以在阶次 p 选定后补入数据阵, 然后用 Householder 变换的实时递推算法^[10] 获得 AR 参数估计。这样获得的计算结果与协方差法或修正协方差法相同。

在数值病态问题严重的情况下, 使用协方差法或修正协方差法获得的白噪声方差估计会变为负值且其绝对值远大于 1, 或者递推系数超过 0—1 范围。在这种情况下, 这两种算法都无法得到 AR 谱估计, 而本文的算法却仍然具有好的估计性能。

四、仿真计算

前面已指出, 在数值病态问题不严重的情况下, 本文的算法可以获得与协方差法或修正协方差法相同的计算结果。因此本文省略掉一些有关 AR 谱估计的著名实例, 如谱分裂和谱线位移、四阶 AR 谱(如下所示的 AR(4)I 模型) 以及文献 [1] 中所使用的 64 点复数据的 AR 谱。应该指出, 即使用 (15)、(16) 式做 AR 谱估计也没有发现显著的谱分裂和谱位移现象。

本文所举的计算实例都是基于前向协方差方程 (15) 式, 计算结果和结论同样适用于以 (16) 式为基础的计算, 因为两式得到的计算结果是相当等同的。而用 (17) 式得到的计算结果, 在大多数情况下, 优于 (15) 或 (16) 式的计算结果。

例 1 选择 AR 阶次。考虑如下 AR 模型:

$$\text{AR(1): } x(n) = 0.5x(n-1) + e(n)$$

$$\text{AR(2): } x(n) = 1.8x(n-1) - 0.97x(n-2) + e(n)$$

$$\text{AR(3): } x(n) = 2.8602x(n-1) - 2.8416x(n-2) + 0.98x(n-3) + e(n)$$

$$\text{AR(4)I: } x(n) = 2.7607x(n-1) - 3.8106x(n-2) \\ + 2.6535x(n-3) - 0.9238x(n-4) + e(n)$$

$$\text{AR(4)II: } x(n) = 3.6741x(n-1) - 5.3457x(n-2) \\ + 3.6226x(n-3) - 0.9723x(n-4) + e(n)$$

$$\text{AR(5): } x(n) = 2.8602x(n-1) - 2.03155x(n-2) - 1.3367x(n-3) \\ + 2.30166x(n-4) - 0.794x(n-5) + e(n)$$

其中 AR(1) 和 AR(2) 引用文献 [11] 中的例子, AR(4)II 引用文献 [12] 中的例子。 $e(n)$ 是具有单位方差的零均白噪声序列。对上述每一个 AR 模型, 分别生成 2000 个独立的 40 点和 80 点数据。对 40 点数据, 分别选 $L = 9$ 和 $L = 12$ (数据阵列数), 对 80 点

数据,分别选 $L = 10$ 和 $L = 15$ 。用本文的算法进行 AR 阶次判定,计算结果示于表 1。由表 1 可以看出,算法是简单而有效的, L 的选择对判阶结果没有明显的影响,尤其当 $N = 80$ 时。

表 1 AR 模型阶次的计算结果 ($PL = 0.85, FH = 1.15$)

2000 个	N = 40		N = 80		
	L = 9	L = 12	L = 10	L = 15	
AR(1)	1	1913	1867	2000	2000
	2	50	71	0	0
	3	32	53	0	0
	其余	5	9	0	0
AR(2)	2	1861	1821	1998	1999
	3	87	95	2	1
	4	46	76	0	0
	其余	6	8	0	0
AR(3)	2	11	8	0	0
	3	1841	1776	2000	1998
	4	86	114	0	1
	5	49	79	0	1
	其余	13	23	0	0
AR(4)I	2	1	2	0	0
	3	18	25	0	0
	4	1820	1743	2000	1995
	5	91	119	0	5
	6	0	85	0	0
	其余	56	26	0	0
AR(4)II	3	31	29	0	0
	4	1786	1699	1999	1997
	5	112	145	1	2
	6	0	90	0	1
	其余	73	35	0	0
AR(5)	3	117	147	6	10
	4	235	299	24	28
	5	1505	1343	1970	1961
	6	0	117	0	0
	7	143	70	0	1
	其余	0	24	0	0

例 2 AR 参数估计。考虑 AR(4)I 模型,生成 50 个独立的 40 点数据。选择 $L = 4, 8, 12, 16$, 分别用本文的算法做 AR 参数估计。50 个 AR 参数估计的均值与方差示于表 2。尽管 L 越接近于 4, 估计结果越好,但是并没有很明显的差别。

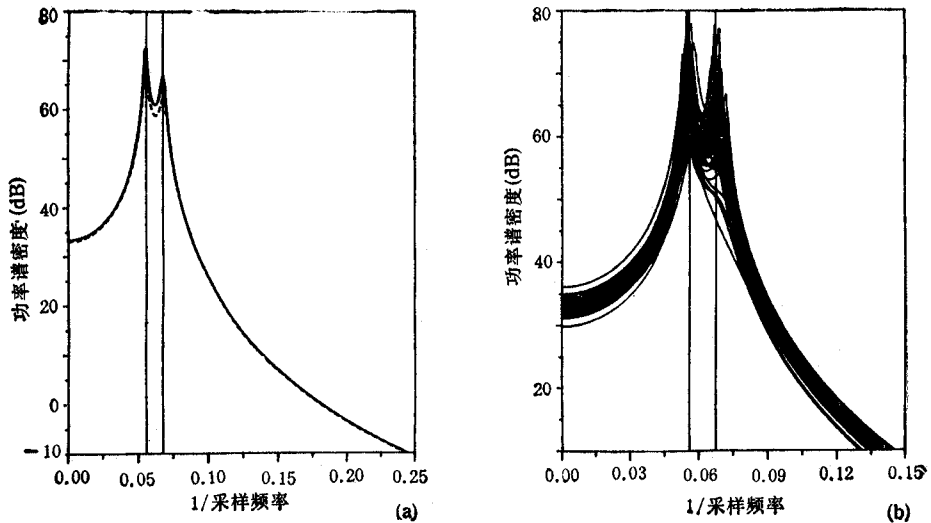
例 3 AR 谱估计。考虑 AR(4)II 模型。生成 50 个独立的 160 点数据。用本文的算法求解 AR 参数估计,均值和方差示于表 3。由于 N 远大于 L , 所以对于不同的 L 值,计算结果没有明显的差别。

表 2 AR(4)I 参数估计的均值和方差 ($\times 0.01$)

理论值	L = 4		L = 8		L = 12		L = 16	
	平均值	方差	平均值	方差	平均值	方差	平均值	方差
2.7607	2.7227	0.92	2.7087	1.211	2.7107	1.516	2.7003	1.886
3.8106	3.7075	5.315	3.6731	7.149	3.6744	9.067	3.6617	10.89
2.6535	2.5482	5.532	2.5148	7.332	2.5155	9.124	2.5111	10.94
0.9238	0.8754	1.170	0.8605	1.527	0.8588	1.848	0.8629	2.143

表 3 AR(4)II 参数估计的均值和方差 ($\times 0.01$)

理论值	L = 4		L = 8		L = 12		L = 16	
	平均值	方差	平均值	方差	平均值	方差	平均值	方差
3.6741	3.6581	0.097	3.6577	0.099	3.6563	0.084	3.6546	0.128
5.3457	5.3020	0.760	5.3009	0.776	5.2969	0.840	5.2915	0.996
3.6226	3.5807	0.737	3.5796	0.754	3.5757	0.812	3.5698	0.966
0.9723	0.8754	0.089	0.9579	0.091	0.9565	0.098	0.9543	0.117

图 1 50 个实现的 AR 谱估计及其均值
——理论值RHLS 平均值

用本文算法所得到的 AR 谱估计 (50 个独立实现, $L = 4$) 及其平均谱示于图 1。两个谱峰之差仅为 0.0117。由于这个例子是病态的, 协方差法和修正协方差法在 50 次计算中有多次失败。

五、结 论

本文提出一种基于 Householder 变换的 AR 模型化和谱估计的递推算法。这一算法的特点在于利用了 Householder 变换的数值稳定性。本文给出的数值计算实例表明

了本文算法的实用性。

参 考 文 献

- [1] S. L. Marple, *Digital Spectral Analysis with Applications*, Prentice-Hall Inc., NJ (1987).
- [2] S. M. Kay, S. L. Marple, *Proc. IEEE*, **69**(1981), 1380—1419.
- [3] J. Makhoul, *Proc. IEEE*, **63**(1975), 561—580.
- [4] Morf., et al., *IEEE Trans. on ASSP*, **ASSP-25** (1977), 429—433.
- [5] S. L. Marple, *IEEE Trans. on ASSP*, **ASSP-29** (1981), 62—73.
- [6] J. A. Cadzow et al., *Proc. Inst. Elec. Eng.*, **130**(1983), Part F, 202—210.
- [7] C. L. Lawson, R. J. Hanson, *Solving Least Squares Problems*, Englewood Cliffs, Prentice-Hall, NJ (1974).
- [8] Huang Junqin, Huili Yu, A Recursive Householder Algorithm for Identification and MA/AR Spectral Estimation, Seventh IFAC/IFORS Symposium on Iden. and System Parameter Esti., Vol. 2, pp 1461—1466, July, (1985).
- [9] A. J. W. Van Den Boom, A. W. M. van Den Enden, *Automatica*, **10**(1974), 245—256.
- [10] 余辉里, AR 谱估计的一种自适应算法, 电子科学学刊, **11**(1989)4, 428—433.
- [11] M. Wax, *IEEE Trans. on ASSP*, **ASSP-36** (1988), 581—588.
- [12] S. H. Leung, W. Y. Horng, Superresolution Autoregressive Spectral Estimation Technique Using Multiple Step Prediction, *Proc. IEEE Int. Cont. ASSP*, Vol. 1, pp 101—104, Mar. (1985).

A RECURSIVE ALGORITHM FOR AR MODELING AND SPECTRAL ESTIMATION USING HOUSEHOLDER TRANSFORM

Yu Huili

(Research Institute No. 634, Ministry of Aviation Industry, Beijing)

Abstract Householder transform is used to triangularize the data matrix which is based on the linear prediction error equation. It is proved that the sum of squared residuals for each AR order can be obtained by the main diagonal elements of upper triangular matrix, so the column by column procedure can be used to develop a recursive algorithm for AR modeling and spectral estimation. In the most cases, the presented algorithm yield the same results as the covariance method or modified covariance method does. But in some special cases where the numerical ill-conditioned problems are so serious that the covariance method and modified covariance method fail to estimate AR spectrum, the presented algorithm still tends to keep good performance. The typical computational results are given finally.

Key words AR spectral estimation; Householder transform; AR parameter; Recursive algorithm