function [Err,Mu,YFT] = EvalTree(AugX ,Y,Tree,C)

YNodecx = AugX\*C;  **//矩阵相乘**

[mAugX,nAugX]=size(AugX); **//找到AugX的行列数**

Mu(:,1) = ones(mAugX,1); **//生成AugX行数个全为1的一个列向量**

for i = 2 : size(Tree,1) **//循环次数为Tree的行数-1，也就是样本数-1**

parent = find(Tree(:,1) == Tree(i,4)); **//将Tree第一列中等于Tree中第i行**

**//第4列中相等的元素行数记录到parent中**

AuxiliaryMu = EvalIms(YNodecx(:,parent),[Tree(i,5) Tree(i,6)]);**//调用EvalIms函数，生成AuxiliaryMu**

Mu(:,i) = AuxiliaryMu.\*Mu(:,parent); **//生成Mu的第i列**

end

Leaves = find(Tree(:,2)==0 & Tree(:,3)==0);  **//找到第二及第三列都为0的行数；**

if Leaves == 1,

YFT = YNodecx;

else

YFT = sum((YNodecx(:,Leaves).\*Mu(:,Leaves))')'./sum(Mu(:,Leaves)')';

end

Err = norm(Y-YFT)/sqrt(mAugX);

function [Err,Mu,YFT] = EvalTree(AugX ,Y,Tree,C) % Err为模型输出的均方根误差；YFT 为模型的输出；AugX 为输入样本的变形，输入样本加一列全为1的矩阵；Y 模型的输出样本；Tree,C为模型参数 YNodecx = AugX\*C; %两矩阵的乘积，得到一个节点的输出[mAugX,nAugX]=size(AugX); %求矩阵AugX 的行和列，mAugX 为行数，nAugX 为列数Mu(:,1) = ones(mAugX,1); % 定义一个mAugX 行1列的全为1的向量for i = 2 : size(Tree,1) parent = find(Tree(:,1) == Tree(i,4)); %找父节点的编号 AuxiliaryMu = EvalIms(YNodecx(:,parent),[Tree(i,5) Tree(i,6)]); %计算当前节点的辅助隶属度函数 Mu(:,i) = AuxiliaryMu.\*Mu(:,parent); %计算当前节点的隶属度函数endLeaves = find(Tree(:,2)==0 & Tree(:,3)==0); %找出叶节点的编号if Leaves == 1, YFT = YNodecx; %不用编写else YFT = sum((YNodecx(:,Leaves).\*Mu(:,Leaves))')'./sum(Mu(:,Leaves)')'; %模型（算法）的输出endErr = norm(Y-YFT)sqrt(mAugX); %模型输出的均方根误差

设输入变量数为m；

AugX矩阵维数为100\*11，100为预测的样本数量，有10个变量；

Y为100\*1；

~~Tree矩阵维数为x\*6~~；经验证，Tree的结构为[2m-1，6]；

~~C矩阵维数为11\*x~~；经验证， C的结构为：[m+1，2m-1]；

YNodecx矩阵维数为100\*x，x取决于建模样本的数量，不能确定，是个变值，只要建模的样本定了，x不会再变了；

Mu(:,1) 矩阵维数 为100\*1的全1矩阵；

Mu(:,i) 矩阵维数 这个矩阵是循环，行数为100，列数由1列逐渐增加，一直增到x列，最后的矩阵维数为100\*x；