摘要

随着人们的生活水平逐步上升，健康问题越来越受到人们的重视。在科学高度发展的今天，防癌治癌成为人们关注的热点话题。目前，结直肠癌的发病率及死亡率有明显的升高趋势，防治结直肠癌成为当代人们关切的问题。本文在基于人体生理学的背景角度出发，使用数学思想及计算方法，借助可实现数据挖掘软件及模型，构建能够在一定程度判断直肠癌、预测病人生存时长的数学模型。对此，我们展开了以下工作：

首先，对于直肠癌影像诊断过程，我们将本题拆解成三步：读取数据并筛选、分割图像；建立模型预测病人的生存时长；对分割结果进行检验。

针对问题一，我们依据题目提供文献，综合赛制期间收集的文献及材料发现，卷积神经网络（CNN）可在读取数据的基础上同时实现图像特征的筛选和图像的分割。而除卷积神经网络外，还可实现该目标功效的有随机森林（Random Forest）。我们考虑到现有的软件及知识储备情况，选择采用matlab软件及其神经网络工具箱实现随机森林模型（Random Forest）的构建，实现病人CT影像中的直肠癌筛选与图像分割。

针对问题二，我们依据题目提供文献了解到对于病人的生存时长可通过logistic 回归、支持向量机（SVM）、生存预测函数（Cox）对病人的生存时长进行测算。

针对问题三，我们借助DICE系数对于我们预测的CT图像上的直肠癌状况与已知的掩盖图像进行比对，借助Python实现预测结果与实际情况的比对，验证该模型对于临床的价值。

最后，我们回顾上述过程，分析模型构建过程中的优缺点，对模型中可能出现的误差进行了原因分析于阐释，反思模型创造的社会意义。此外，我们对于模型提出研究建议与设计改进建议，为发现更好的模型优化方案贡献绵薄之力。

关键词：直肠癌；CT影响识别；Matlab；Python；随机森林模型（Random Forest）；DICOM处理；DICE系数

1.问题重述

1.1背景

1.2问题

2.问题分析

2.1 问题一：读取数据并筛选、分割图像，进行特征提取

2.2问题二：建立模型预测病人的生存时长

2.3 问题三：对分割结果进行检验

3.基本假设与符号说明

3.1基本假设

3.2符号说明

4.模型的建立与求解

4.1模型准备

关于DICOM格式的图像：

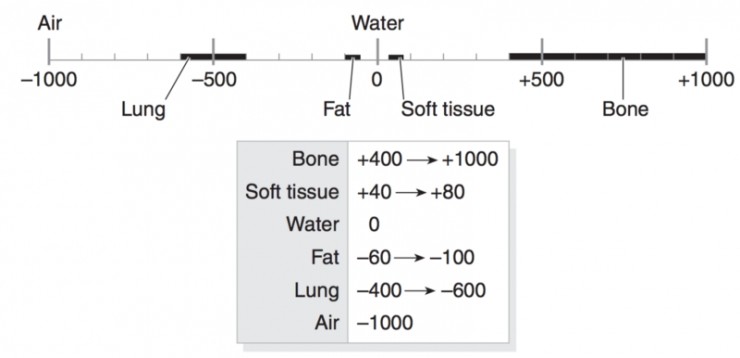
因为不同物质吸收X射线能力不同，所以便有了CT图像，它根据X光射线来吸收剖面的重构图像，X射线可用于诊断。CT图像被用来做大量的医学影像分析，可供医生窥视到人体内的结构，判断是否出现症状，便于早期诊断筛查。CT图像特异性及敏感性相较于MR图像，要差得多。

CT成像中的伪影包括：部分容积效应、条形伪影、运动伪影、束硬化伪影、环状伪影、金属伪影等。由于这些伪影的存在给CT图像分割带来了一定的难度，不同组织部位分割精度也不一样。

为了研究CT图像中的伪影及还原患者相关数据，便出现了DICOM通讯协议。

CT扫描会测量放射性密度，存在名叫“胡斯菲尔德单元”（HU）的测量的单元。以下是关于CT扫描测量的详细内容：

在voxel内通过计算所有衰减值可以得到平均值，该CT图像中每个像素点都会被分配到这个平均值。而“胡斯菲尔德单元”（HU）是以任意单元的规模显示的，也就是这个与水的衰减值相比较得出的平均值。

水的衰减值被这个任意单元的规模定为零。CT数字的范围平均是2000HU。每个数值都代表了一种灰色阴影，在光谱两端有+1000白色和-1000黑色。有些扫描仪是有柱状扫描边界的，但是其输出的图像确实方形的。在扫描边界之外的像素将被赋予-2000的定值。

医学图像的自动分割是一项困难的工作，因为医学图像非常复杂，缺乏简单的线性特征，此外，分割结果的准确性还受到体积效应、灰度不均匀性、伪影、不同软组织间灰度接近度等因素的影响。

CT和MR图像分割主要涉及三个相关问题：噪声变化、像素灰度分类的不确定性和灰度的非均衡性。

一般而言，图像中单个组织的灰度级逐渐变化，其概率密度受特定分布函数的影响。 组织的相应图像区域包含有限数量的像素并满足部分体积平均。

但是，该区域中单个像素的灰度级与任何类别的灰度级不同，并且通常被视为混合组织类。

目前，常用的CT和MR图像分割方法包括基于阈值的分割，基于区域的分割，基于变形模型的分割，模糊分割和基于神经网络的分割。

传统的图像处理技术主要包括两部分：特征提取和分类器。 讨论了特征提取算法的设计复杂性，应用限制和稳定性。 特定特征提取算法与特定分类器组合的多样性限制了图像处理技术的发展。 神经网络的出现使得端到端图像处理成为可能。

当网络的隐藏层发展到多层次时，它被称为深度学习，但同时，它还需要逐层初始化技术来解决深度网络训练的问题。 从那时起，深入学习成为时代的主角。 卷积神经网络（cnn）是由深度学习和图像处理技术相结合产生的经典模型。 实现该模型的网络示例对于处理特定图像问题是有效的。

在对于医学CT图像进行分析时，首先我们要确定直肠癌的特征，确定特征划分标准及提取特征算法；随后，我们对于特征变化进行预测，并检验特征变化预测值；最后我们对于模型弊端进行改良。

在开始预测直肠癌的特征变化时，我们首先将CT图像数据导入matlab中，为图像的成像与分割及特征提取做准备。基于图像数据导入matlab后，我们借助matlab对图像进行了成像与分割。在进行图像成像时，我们先对图像进行灰度归一化处理以加快神经网络的收敛性和求解问题的速度，即使其像素的灰度值分布在0~255之间，避免图像对比度不足（图像像素亮度分布不平衡）从而对后续处理带来干扰。

我们基于图像灰度线性归一化对图像实现归一化：

我们借助matlab，直接调用matlab函数实现图像灰度归一化。

*% Another way to normalazation, which only calls MATLAB function*

img3 = mat2gray(oriImage);

figure;

imshow(img3);

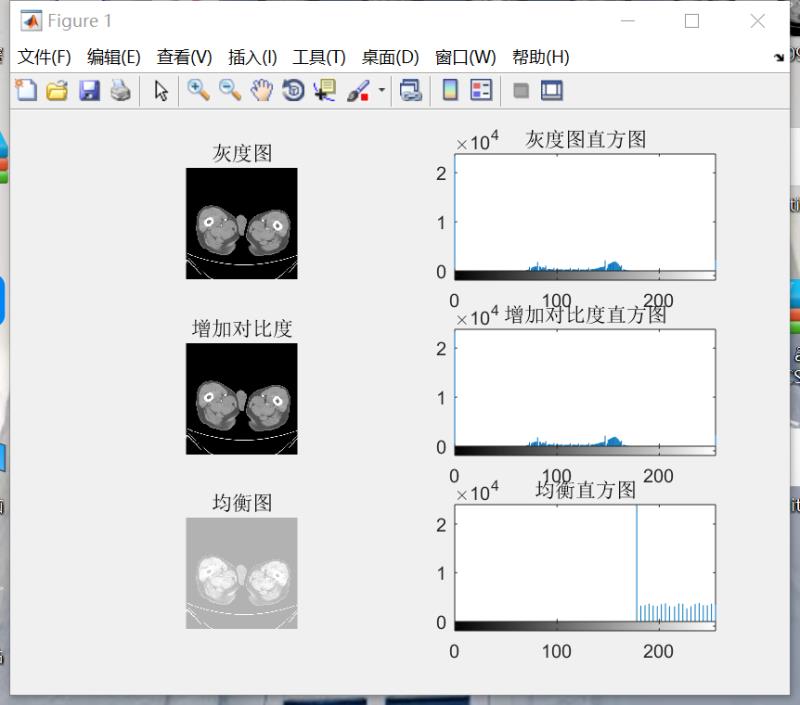
4.2求解

4.2.1 问题一的求解

在实现图像的归一化处理后，我们着手进行图像分割及图像特征提取。在图像特征方面，我们关注图像表现的强度、形状、纹理、小波特征。在进行图像特征提取时，我们基于数学方法定义图像特征，经分别计算后放入分类器进行分类。

首先，我们采用直方图将一幅图内的各像素分布情况描绘出来，基于

以像素作为横坐标，对应频率为纵坐标，通过matlab代码得到灰度图像的直方图。



vox\_val\_hist = zeros(num\_img\_values,1);

for this\_vox\_value = 1:num\_img\_values

vox\_val\_hist(this\_vox\_value) = length(find((img\_vol\_subvol == this\_vox\_value) & (mask\_vol\_subvol == 1) ));

end

*% Compute the relative probabilities from the histogram:*

vox\_val\_probs = vox\_val\_hist / num\_ROI\_voxels;

*% Compute the histogram\_based metrics:*

texture\_metrics(1:6) = compute\_histogram\_metrics(vox\_val\_probs,num\_img\_values);

随后，我们基于直方图的PET/CT纹理特征，进行特征提取。在特征值提取过程中，我们对6个特征代表值进行测算，这6个特征代表值分别有：均值、方差、偏态、峰态、能量、熵，并使用matlab代码实现特征获取。

val\_indices = (1:num\_img\_values)';

indice = find(prob);

vector(1) = sum(val\_indices .\* prob);

vector(2) = sum( ((val\_indices - vector(1)).^2) .\* prob );

if vector(2) > 0

vector(3) = sum( ((val\_indices - vector(1)).^3) .\* prob ) / (vector(2)^(3/2));

vector(4) = sum( ((val\_indices - vector(1)).^4) .\* prob ) / (vector(2)^2);

vector(4) = vector(4) - 3;

else

vector(3) = 0;

vector(4) = 0;

end

vector(5) = sum( prob .^2 );

vector(6) = -sum( prob(indice) .\* log(prob(indice)) );

我们提取到直肠癌的特征后，进行边缘及轮廓计算。我们调用matlab中的sobel函数，求取边缘的同时求取梯度方向特征，

void sobel\_te(IplImage \*img,IplImage \* sobel8u){

IplImage\* sobel=cvCreateImage( cvGetSize(img), IPL\_DEPTH\_16S, 1 );

IplImage\* sobel\_y=cvCreateImage( cvGetSize(img), IPL\_DEPTH\_16S, 1 );

IplImage\* sobel\_x=cvCreateImage( cvGetSize(img), IPL\_DEPTH\_16S, 1 );

cvSobel(img,sobel\_y,1,0,3); //提取Y方向

cvSobel(img,sobel\_x,1,0,3); //提取X方向

cvOr(sobel\_x,sobel\_y,sobel,NULL);

cvConvertScaleAbs(sobel,sobel8u,1,0);//把sobel计算后的值由16位转为8位供正常显示

sobel8u->origin = img->origin;

}

至此，我们得到了直肠癌CT图像的边缘与轮廓特征，实现直肠癌的筛选和提取。

4.2.2 问题二的求解

在对于病人的生存时长方面，我们采用随机森林模型进行预测。

首先，我们将已知的每张病人CT图像作为训练集，每次随机并有序放回训练集中的N个样本，实现迭代抽样，作为树的训练集，并开始对于行和列进行采样。

clear all;

right\_node=cell(3,1);*%3\*1的单元数组*

*% rchild\_value=cell(3,1);%3\*1的单元数组*

*% rchild\_node\_num=cell(3,1);%3\*1的单元数组*

sn=300; *%随机可重复的抽取sn个样本*

tn=10; *%森林中决策树的数目*

load('abc.mat');

n = size(r,1);

*%% 样本训练采用随机森林和ID3算法构建决策森林*

arr = [];

for j=1:tn

sample=randi([1,n],1,sn);*%从1至1000内随机抽取sn个样本*

SData=r(sample,:);

[tree,arr]= train\_C4\_5(SData, 5, 10, arr);

right\_node{j,1}=tree;

end

load('abc.mat');

T = r;

TData = roundn(T,-1);

*%统计函数，对输入的测试向量进行投票，然后统计出选票最高的标签类型输出*

result = statistics(tn, right\_node, TData, arr);

gd = T(:,end);

len = length(gd);

count = sum(result==gd);

fprintf('共有%d个样本','判断正确的有%d\n',len,count);

之后，我们对于训练集中的树进行迭代抽样，展开训练。

function [tree,discrete\_dim] = train\_C4\_5(S, inc\_node, Nu, discrete\_dim)

*% Classify using Quinlan's C4.5 algorithm*

*% Inputs:*

*% train\_ptn - Train patterns 训练样本 每一列代表一个样本 每一行代表一个特征*

*% train\_trg - Train targets 1×训练样本个数 每个训练样本对应的判别值*

*% test\_patterns - Test patterns 测试样本，每一列代表一个样本*

*% node - Percentage of incorrectly assigned samples at a node 一个节点上未正确分配的样本的百分比*

*% node为防止过拟合，表示样本数小于一定阈值结束递归，可设置为5-10*

*% 注意inc\_node设置太大的话会导致分类准确率下降，太小的话可能会导致过拟合*

*% Nu is to determine whether the variable is discrete or continuous (the value is always set to 10)*

*% Nu用于确定变量是离散还是连续（该值始终设置为10）*

*% 这里用10作为一个阈值，如果某个特征的无重复的特征值的数目比这个阈值还小，就认为这个特征是离散的*

*% Outputs*

*% test\_targets - Predicted targets 1×测试样本个数 得到每个测试样本对应的判别值*

*% 也就是输出所有测试样本最终的判别情况*

*%NOTE: In this implementation it is assumed that a pattern vector with fewer than 10 unique values (the parameter Nu)*

*%is discrete, and will be treated as such. Other vectors will be treated as continuous*

*% 在该实现中，假设具有少于10个无重复值的特征向量（参数Nu）是离散的。 其他向量将被视为连续的*

train\_ptn = S(:,1:end-1)';

train\_trg = S(:,end)';

[Ni, M] = size(train\_ptn); *%M是训练样本数，Ni是训练样本维数，即是特征数目*

node = node\*M/100; *% 5\*训练样本数目/100*

if isEmpty(dim)

*%Find which of the input patterns are discrete, and discretisize the corresponding dimension on the test patterns*

*%查找哪些输入模式(特征)是离散的，并离散测试模式上的相应维*

dim = zeros(1,Ni); *%用于记录每一个特征是否是离散特征，初始化都记为0，代表都是连续特征，*

*%如果后面更改，则意味着是离散特征，这个值会更改为这个离散特征的无重复特征值的数目*

for i = 1:Ni *%遍历每个特征*

Ub = unique(train\_ptn(i,:)); *%取每个特征的不重复的特征值构成的向量*

Nb = length(Ub); *%得到无重复的特征值的数目*

if (Nb <= Nu) *%如果这个特征的无重复的特征值的数目比这个阈值还小，就认为这个特征是离散的*

*%This is a discrete pattern*

dim(i) = Nb; *%得到训练样本中，这个特征的无重复的特征值的数目 存放在discrete\_dim(i)中，i表示第i个特征*

*% dist = abs(ones(Nb ,1)\*test\_patterns(i,:) - Ub'\*ones(1, size(test\_patterns,2)));*

*% %前面是把测试样本中，这个特征的那一行复制成Nb行，Nb是训练样本的这个特征中，无重复的特征值的数目*

*% %后面是把这几个无重复的特征值构成的向量复制成测试样本个数列*

*% %求这两个矩阵相应位置差的绝对值*

*% [m, in] = min(dist); %找到每一列绝对差的最小值，构成m(1×样本数目) 并找到每一列绝对差最小值所在行的位置，构成in(1×样本数目)*

*% %其实，这个in的中每个值就是代表了每个测试样本的特征值等于无重复的特征值中的哪一个或者更接近于哪一个*

*% %如=3，就是指这个特征值等于无重复的特征值向量中的第3个或者更接近于无重复的特征值向量中的第3个*

*% test\_patterns(i,:) = Ub(in); %得到这个离散特征*

end

end

end

*%Build the tree recursively 递归地构造树*

*% disp('Building tree')*

flag = [];

T = make\_tree(train\_ptn, train\_trg, node, dim, max(dim), 0, flag);

function T = make\_tree(patterns, targets, node, dim, maxNbin, base, flag)

*%Build a T recursively 递归地构造树*

[N\_all, L] = size(patterns); *%%L为当前的样本总数，Ni为特征数目*

if isEmpty(flag)

nodeChoose = randi([1,N\_all],1,0.5\*sqrt(N\_all));*%从所有特征中随机选择m个*

Ni\_choose = length(nodeChoose);

flag.nodeChoose = nodeChoose;

flag.NiChoose = NiChoose;

else

nodeChoose = flag.nodeChoose;

NiChoose = flag.NiChoose;

end

Uc = unique(targets); *%训练样本对应的判别标签 无重复的取得这些标签 也就是得到判别标签的个数*

T.dim = 0; *%初始化树的分裂特征为第0个*

*%T.child(1:maxNbin) = zeros(1,maxNbin);*

T.split\_loc = inf; *%初始化分裂位置是inf*

if isEmpty(patterns)

return

end

*% inc\_node为防止过拟合，表示样本数小于一定阈值结束递归，可设置为5-10*

if ((node > L) | (L == 1) | (length(Uc) == 1)) *%如果剩余训练样本太小(小于inc\_node)，或只剩一个，或只剩一类标签，退出*

H = hist(targets, length(Uc)); *%统计样本的标签，分别属于每个标签的数目 H(1×标签数目)*

[m, largest] = max(H); *%得到包含样本数最多的那个标签的索引，记为largest 包含多少个样本，记为m*

T.Nf = [];

T.split\_loc = [];

T.child = Uc(largest);*%姑且直接返回其中包含样本数最多一类作为其标签*

return

end

for i = 1:length(Uc) *%遍历判别标签的数目*

Pnode(i) = length(find(targets == Uc(i))) / L; *%得到当前所有样本中 标签=第i个标签 的样本的数目 占样本总数的比例 存放在Pnode(i)中*

end

*% 计算当前的信息熵（分类期望信息）*

*% 例如，数据集D包含14个训练样本，9个属于类别“Yes”，5个属于类别“No”，Inode = -9/14 \* log2(9/14) - 5/14 \* log2(5/14) = 0.940*

Inode = -sum(Pnode.\*log(Pnode)/log(2));

*%For each dimension, compute the gain ratio impurity ％对于每维，计算杂质的增益比 对特征集中每个特征分别计算信息熵*

*%This is done separately for discrete and continuous patterns ％对于离散和连续特征，分开计算*

delta\_Ib = zeros(1, NiChoose); *%Ni是特征维数 用于记录每个特征的信息增益率*

split\_loc = ones(1, NiChoose)\*inf; *%初始化每个特征的分裂值是inf*

for i\_idx = 1:NiChoose*%遍历每个特征*

i = nodeChoose(i\_idx);

data = patterns(i,:); *%得到当前所有样本的这个特征的特征值*

Ud = unique(data); *%得到无重复的特征值构成向量Ud*

Nbins = length(Ud); *%得到无重复的特征值的数目*

if (dim(i)) *%如果这个特征是离散特征*

P = zeros(length(Uc), Nbins); *%Uc是判别标签的个数 判别标签个数×无重复的特征值的数目*

for j = 1:length(Uc) *%遍历每个标签*

for k = 1:Nbins *%遍历每个特征值*

indices = find((targets == Uc(j)) & (patterns(i,:) == Ud(k)));

*% &适用于矩阵间的逻辑运算 &&不适用，只能用于单个元素 &的意思也是与*

*%找到 （样本标签==第j个标签 并且 当前所有样本的这个特征==第k个特征值） 的样本个数*

P(j,k) = length(indices); *%记为P(j,k)*

end

end

Pk = sum(P); *%取P的每一列的和，也就是得到当前所有样本中，这个特征的特征值==这个特征值的样本数目 Pk(1×特征值数目)表示这个特征的特征值等于每个特征值的样本数目*

P1 = repmat(Pk, length(Uc), 1); *%把Pk复制成 判别标签个数 行*

P1 = P1 + eps\*(P1==0); *%这主要在保证P1作被除数时不等于0*

P = P./P1; *%得到当前所有样本中，这个特征的值等于每个特征值且标签等于每个标签的样本，占当前这个特征值中的样本的比例*

Pk = Pk/sum(Pk); *%得到当前所有样本中，这个特征的值等于每个特征值的样本，占当前样本总数的比例*

info = sum(-P.\*log(eps+P)/log(2)); *%对特征集中每个特征分别计算信息熵 info(1×特征值数目)*

delta\_Ib(i\_idx) = (Inode-sum(Pk.\*info))/(-sum(Pk.\*log(eps+Pk)/log(2))); *%计算得到当前特征的信息增益率*

*%计算信息增益率(GainRatio)，公式为Gain(A)/I(A),*

*%其中Gain(A)=Inode-sum(Pk.\*info)就是属性A的信息增益*

*%其中I(A)=-sum(Pk.\*log(eps+Pk)/log(2))为属性A的所包含的信息*

else *%对于连续特征(主要要找到合适的分裂值，使数据离散化)*

*%This is a continuous pattern*

P = zeros(length(Uc), 2); *% P(判别标签数目×2) 列1代表前..个样本中的标签分布情况 列2代表除前..个样本之外的标签分布情况 这个..就是各个分裂位置*

*%Sort the patterns*

[sorted\_data, indices] = sort(data); *%data里存放的是当前所有训练样本的这个特征的特征值 从小到大排序 sorted\_data是排序好的数据 indices是索引*

sorted\_targets = targets(indices); *%当然，判别标签也要随着样本顺序调整而调整*

*%Calculate the information for each possible split 计算分裂信息度量*

I = zeros(1,Nbins);

delta\_Ib\_inter = zeros(1, Nbins);

for j = 1:Nbins-1

P(:, 1) = hist(sorted\_targets(find(sorted\_data <= Ud(j))) , Uc); *%记录<=当前特征值的样本的标签的分布情况*

P(:, 2) = hist(sorted\_targets(find(sorted\_data > Ud(j))) , Uc); *%记录>当前特征值的样本的标签的分布情况*

Ps = sum(P)/L; *%sum(P)是得到分裂位置前面和后面各有样本数占当前样本总数的比例*

P = P/L; *%占样本总数的比例*

Pk = sum(P); *%%sum(P)是得到分裂位置前面和后面各有多少个样本 比例*

P1 = repmat(Pk, length(Uc), 1); *%把Pk复制成 判别标签个数 行*

P1 = P1 + eps\*(P1==0);

info = sum(-P./P1.\*log(eps+P./P1)/log(2)); *%计算信息熵（分类期望信息）*

I(j) = Inode - sum(info.\*Ps); *%Inode-sum(info.\*Ps)就是以第j个样本分裂的的信息增益*

delta\_Ib\_inter(j) = I(j)/(-sum(Ps.\*log(eps+Ps)/log(2))); *%计算得到当前特征值的信息增益率*

end

[~, s] = max(I); *%找到信息增益最大的划分方法 delta\_Ib(i)中存放的是对于当前第i个特征而言，最大的信息增益作为这个特征的信息增益 s存放这个划分方法*

delta\_Ib(i\_idx) = delta\_Ib\_inter(s); *%得到这个分类特征值对应的信息增益率*

split\_loc(i\_idx) = Ud(s); *%对应特征i的划分位置就是能使信息增益最大的划分值*

end

end

*%Find the dimension minimizing delta\_Ib %找到当前要作为分裂特征的特征*

[m, dim] = max(delta\_Ib); *%找到所有特征中最大的信息增益对应的特征，记为dim*

dims = 1:NiChoose; *%Ni特征数目*

dim\_all = 1:N\_all;

dim\_to\_all = nodeChoose(dim);

T.dim = dim\_to\_all; *%记为树的分裂特征*

*%Split along the 'dim' dimension*

Nf = unique(patterns(dim\_to\_all,:)); *%得到选择的这个作为分裂特征的特征的那一行 也就是得到当前所有样本的这个特征的特征值*

Nbins = length(Nf); *%得到这个特征的无重复的特征值的数目*

T.Nf = Nf; *%记为树的分类特征向量 当前所有样本的这个特征的特征值*

T.split\_loc = split\_loc(dim); *%把这个特征的划分位置记为树的分裂位置 可是如果选择的是一个离散特征，split\_loc(dim)是初始值inf啊？？？*

*%If only one value remains for this pattern, one cannot split it.*

if (Nbins == 1) *%无重复的特征值的数目==1，即这个特征只有这一个特征值，就不能进行分裂*

H = hist(targets, length(Uc)); *%统计当前所有样本的标签，分别属于每个标签的数目 H(1×标签数目)*

[m, largest] = max(H); *%得到包含样本数最多的那个标签的索引，记为largest 包含多少个样本，记为m*

T.Nf = []; *%因为不以这个特征进行分裂，所以Nf和split\_loc都为空*

T.split\_loc = [];

T.child = Uc(largest); *%姑且将这个特征的标签就记为包含样本数最多的那个标签*

return

end

if (dim(dim\_to\_all)) *%如果当前选择的这个作为分裂特征的特征是个离散特征*

*%Discrete pattern*

for i = 1:Nbins *%遍历这个特征下无重复的特征值的数目*

indices = find(patterns(dim\_to\_all, :) == Nf(i)); *%找到当前所有样本的这个特征的特征值为Nf(i)的索引们*

T.child(i) = make\_tree(patterns(dim\_all, indices), targets(indices), node, dim(dim\_all), maxNbin, base, flag);*%递归*

*%因为这是个离散特征，所以分叉成Nbins个，分别针对每个特征值里的样本，进行再分叉*

end

else

*%Continuous pattern %如果当前选择的这个作为分裂特征的特征是个连续特征*

indices1 = find(patterns(dim\_to\_all,:) <= split\_loc(dim)); *%找到特征值<=分裂值的样本的索引们*

indices2 = find(patterns(dim\_to\_all,:) > split\_loc(dim)); *%找到特征值>分裂值的样本的索引们*

if ~(isEmpty(indices1) | isEmpty(indices2)) *%如果<=分裂值 >分裂值的样本数目都不等于0*

T.child(1) = make\_tree(patterns(dim\_all, indices1), targets(indices1), node, dim(dim\_all), maxNbin, base+1, flag);*%递归*

*%对<=分裂值的样本进行再分叉*

T.child(2) = make\_tree(patterns(dim\_all, indices2), targets(indices2), node, dim(dim\_all), maxNbin, base+1, flag);*%递归*

*%对>分裂值的样本进行再分叉*

else

H = hist(targets, length(Uc)); *%统计当前所有样本的标签，分别属于每个标签的数目 H(1×标签数目)*

[m, largest] = max(H); *%得到包含样本数最多的那个标签的索引，记为largest 包含多少个样本，记为m*

T.child = Uc(largest); *%姑且将这个特征的标签就记为包含样本数最多的那个标签*

T.dim = 0; *%树的分裂特征记为0*

end

end

经过行采样和列采样之后，我们对于随机森林模型进行测试。

function [result] = statistics(tn, rnode, PValue, discrete\_dim)

TypeArr = {'1','2'};

TypeNum = [0 0];

test\_patterns = PValue(:,1:end-1)';

num = length(TypeNum);

type = zeros(tn,size(test\_patterns,2));

for i = 1:tn

type(tn,:) = vote\_C4\_5(test\_patterns, 1:size(test\_patterns,2), rnode{i,1}, discrete\_dim, num);

end

result = mode(type,1)';

end

最后，我们通过随机森林模型判定的直肠癌特征情况，预测病人的生存时长，最终得到样本测试结果。

function targets = vote\_C4\_5(patterns, indices, tree, discrete\_dim, Uc)

trgs = zeros(1, size(patterns,2)); *%设置每个样本的初始预测标签都是0*

if (tree.dim == 0) *%这说明达到了树的叶子节点*

*%Reached the end of the tree*

trgs(indices) = tree.child; *%得到样本对应的标签是tree.child*

return

end

dim = tree.dim; *%得到分裂特征*

dims= 1:size(patterns,1); *%得到特征索引*

if (discrete\_dim(dim) == 0) *%如果当前分裂特征是个连续特征*

*%Continuous pattern*

in = indices(find(patterns(dim, indices) <= tree.split\_loc)); *%找到当前测试样本中这个特征的特征值<=分裂值的样本索引*

trgs = trgs + vote\_C4\_5(patterns(dims, :), in, tree.child(1), discrete\_dim(dims), Uc); *%对这部分样本再分叉*

in = indices(find(patterns(dim, indices) > tree.split\_loc)); *%找到当前测试样本中这个特征的特征值>分裂值的样本索引*

trgs = trgs + vote\_C4\_5(patterns(dims, :), in, tree.child(2), discrete\_dim(dims), Uc); *%对这部分样本再分叉*

else *%如果当前分裂特征是个离散特征*

Uf = unique(patterns(dim,:)); *%得到这个样本集中这个特征的无重复特征值*

for i = 1:length(Uf) *%遍历每个特征值*

if any(Uf(i) == tree.Nf) *%tree.Nf为树的分类特征向量 当前所有样本的这个特征的特征值*

in = indices(find(patterns(dim, indices) == Uf(i))); *%找到当前测试样本中这个特征的特征值==分裂值的样本索引*

trgs = trgs + vote\_C4\_5(patterns(dims, :), in, tree.child(find(Uf(i)==tree.Nf)), discrete\_dim(dims), Uc);*%对这部分样本再分叉*

end

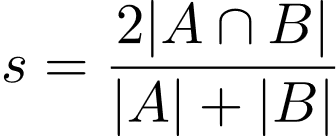
end

end

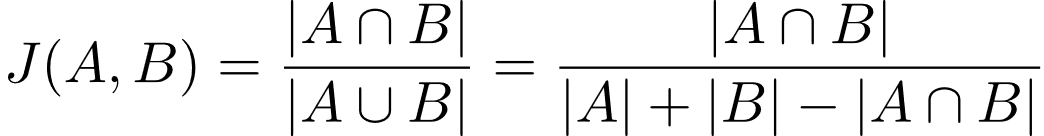
4.2.3 问题三的求解

关于图像的分析：

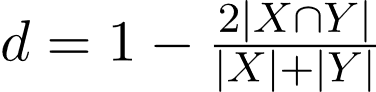
Dice距离用于度量两个集合的相似性，因为可以把字符串理解为一种集合，因此Dice距离也会用于度量字符串的相似性。此外，Dice系数的一个非常著名的使用即实验性能评测的F1值。Dice系数定义如下：



其中分子是A与B的交集数量的两倍，分母为X和Y的长度之和，所以他的范围也在0到1之间。从公式看，Dice系数和Jaccard非常的类似。Jaccard是在分子和分母上都减去了|A∩B|。

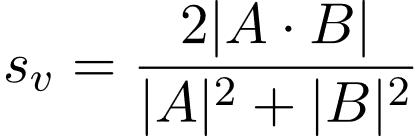


与Jaccard不同的是，相应的差异函数



不是一个合适的距离度量措施，因为它没有三角形不等性的性质。例如给定 {a}, {b}, 和 {a,b}, 前两个集合的距离为1， 而第三个集合和其他任意两个集合的距离为三分之一。

与Jaccard类似, 集合操作可以用两个向量A和B的操作来表示:



利用Python，实现代码为：

def dice\_coefficient(a, b):

"""dice coefficient 2nt/na + nb."""

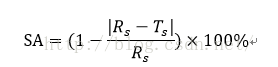
a\_bigrams = set(a)

b\_bigrams = set(b)

overlap = len(a\_bigrams & b\_bigrams)

return overlap \* 2.0/(len(a\_bigrams) + len(b\_bigrams))

关于分割精度：



其中Rs表示专家手工勾画出的分割图像的参考面积，Ts表示算法分割得到的图像的真实面积，|Rs-Ts|表示错误分割的像素点个数。

分割在图像参考面积之外的像素点的比率即为过分割率：

https://img-blog.csdn.net/20161101002459046?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

含义同上，Os表示本不应该包含在分割结果中的像素点个数，实际却在分割结果中的像素点个数。换句话讲，Os中的像素点出现在实际分割图像中，但不出现在理论分割图像Rs中。

分割在GT图像参考面积之中欠缺的像素点的比率即为欠分割率：

https://img-blog.csdn.net/20161101002503838?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

含义同上，Us表示本应该包含在分割结果中的像素点个数，实际却不在分割结果中的像素点个数。换句话讲，Us中的像素点出现在理论分割图像中，但不出现在实际分割图像中。

5.模型评价与改进

5.1 误差分析

5.2 模型评价

5.3 模型改进

6.参考文献

[1]李铭.CT金属伪影校正研究[D].长春:中国科学院研究生院(长春光学精密机械与物理研究所),2015.

[2]于海峰.医学显微图像识别与分析[J].计算机光盘软件与应用,2011,1:73-75.

[3]王洁.基于图学习的医学图像标注方法[J].中国电子商情：通信市场,2013.

[4]赵志平,陈雷月.基于深度传感器图像分割技术的研究[J].信息技术,2014,1:109-112.