

班 级 1302012  
学 号 13020120078

西安电子科技大学

# 本科课程设计报告



题 目 基于颜色统计特征的图像质量

评价算法研究

学 院 电子工程学院

专 业 电子信息工程

学生姓名 李康

导师姓名 何立火

西 安 电 子 科 技 大 学

电 子 工 程 学 院

## 课 程 设 计（报告）任 务 书

---

学生姓名 李康 指导教师 何立火 职称 副教授

学生学号 13020120078 专业 电子信息工程

题目 基于颜色统计特征的图像质量评价算法研究

相关专业课程 数字图像处理、MATLAB

任务与要求

基于彩色图像的失真特征，提取图像的饱和度、颜色协调性等信息，利用 SVR 回归模型完成对彩色图像的质量评价。

---

开始日期 2016 年 12 月 26 日 完成日期 2017 年 1 月 9 日

课程设计所在单位 电子工程学院 2017 年 1 月 9 日

# 基于颜色统计特征的图像质量评价算法研究

## 摘要

目前，社会上的图像信息越来越丰富，发展的非常快，人们越来越重视图像的处理。其中对于彩色图像的质量评价更为重视，一幅图像的质量直接会影响着这个图像的信息。这不仅会影响图像信息的传递以及分析，也会影响人对图像的评价，所以彩色图像的质量评价也逐渐的成为图像研究领域的一个热点。关于彩色图像质量评价的办法也越来越多。

本文通过建立一种算法模型对彩色图像的质量评价进行了重点研究。根据对颜色统计特性的特征值与主观质量值之间的比较，利用支持向量回归模型（SVR）进行学习分析，得到图像质量评价的结果。

实验结果表明，根据算法分析得到的对于彩色图像质量评价分数和主观评价还是比较符合的，但是由于时间有限，颜色特征提取算法算法太慢，个人 PC 难以获得大量图片的颜色特征，导致 SVR 过拟合，训练效果不佳，泛化能力不强。

**关键词：**图像质量评价 支持向量回归模型 颜色分形结构 彩色图像

## **Abstract**

Image information is more and more rich at present, and it's development is very fast. People pay more and more attention to the image processing, Especially for color image quality evaluation. As we all know, The quality of the images will directly affect the image information. It can affect not only transfer and analysis of image information, also can affect people of image evaluation, So color image quality assessment is also gradually become a hot spot in the research field of the image. About color image quality assessment method is also more and more.

This paper is through the establishment of an algorithm model to the study on evaluation about the quality of the color image. We get the results according to the statistical features of the color of the comparison between characteristic value and subjective quality and Support vector regression model.

The experimental results show that the scores for color image quality evaluation of the algorithm is accord with subjective evaluation, but because time is limited, Color feature extraction algorithm algorithm is too slow, and personal PC is hard to get a lot of color characteristics of image, the SVR is too fitting, training effect is not good, and generalization ability is not strong.

**Keywords:** image quality assessment, SVR, color fractal structure, color image

## 一. 引言

在我们日常生活中，我们对信息的获取渠道是非常多的，包括文字，声音，以及图像。获取最多的是来自眼睛，景物和图像可以用语音和文字来描述，但是用图像来表达会更加的简单，也更加的容易让我们理解和接受。随着现代社会生活节奏的加快，科技的不断进步，我们对于信息的获取，更多的依赖于图像，图像不仅包含了更大的信息量，也更加直观的表达了信息，这种信息的传输也得到了人们的喜爱。科技进步使得图像从黑白转为彩色，这就更加方便了信息的传输和表达，也进一步的得到了大家的重视。

新世纪以来，我们见证了数字图像作为传输媒介，这个趋势也一直在上升，而在对图像进行采集、压缩、传输、处理和重现过程中将会造成一定程度的失真。这对图像进行后期处理造成了一定程度的困难，同时也对图像所承载的信息量造成了一定程度的影响。为了保持、控制以及增强图像的质量，可以对图像进行识别和图像质量评价，这对于图像的采集、处理、交流以及处理系统具有很重要的意义<sup>[1]</sup>。

随着现代技术的进步，我们现在处理的图像更多的是彩色图像，颜色对于一幅图像有很大的重要性，如果一幅图片失去了颜色，那么不仅是客观上图像信息的缺失，主观上的观赏性也会随之降低。了解颜色的表征现象并建立模型对颜色进行研究，已经成为了研究图像质量的很大一部分，尤其是在最近的 15—20 年中<sup>[2]</sup>。在评价图片质量的时候，图像颜色也占据了很重要的位置。

图像质量评价的主体是人，因此人眼视觉特性(HVS，如多通道结构、掩盖效应、对比度敏感度和视觉非线性定律等)对评价算法的研究具有重要意义。但是，对人眼视觉特性的研究涉及生理、心理等方面，迄今对其仍没有充分理解，特别是对视觉的心理特性还难以找出定量的描述方法<sup>[3]</sup>。因此通过支持向量回归模型进行学习对图片质量评价可以节省人类很多的时间和精力，在需要处理大量的图片信息或者是要对图像质量进行评价的地方都可以有很大的用处。在其他更多的方面也都可以得到应用，将进一步的推进图像信息的发展。

## 二. 基础原理

### 2.1 图像质量评价简介

图像质量通过衡量处理后的图像与原图像差异度来获得，其本质上是对图像处理系统的优劣以及算法的有效性的反映。图像质量外在表现有两个方面，即图像的可懂度和逼真度。图像的可懂度是指被测图像中能被有效理解与接收的信息，即是否完整传递了原图像所表达的信息。图像的逼真度指的是被测量图像和原图像的相似度，即是否如实还原了图像的外观；被测图像与原图像相似度越高，越完整的传递了原图像的信息，则其图像质量越好。

现有的图像质量评价根据视觉系统可分为：自顶向下和自底向上；根据是否有原始图像可分为：全参考型、部分参考型和无参考型；而通常情况下大体上可以分为两大类，即主观评价方法和客观评价方法。

### 2.2 主观评价方法

人是图像的最终接受者，因此主观评价方法是最可靠的评价方法，主观评价方法主要是通过让观察者根据一些事先规定的评价尺度或自己的经验，对测试图像按视觉效果提出质量判断，并给出质量分数，对所有观察者给出的分数进行加权平均。

为得到定量的主观评价结果，需要设计主观评价实验，其中包括三个步骤：(1) 准备样本集；(2) 由观测者对样本集的图像质量进行评价；(3) 对主观评价结果进行加工。所得的结果即为图像的主观质量评价 (Mean Opinion Score, MOS)。这种测量方法虽然较好地反映出了图像的直观质量，但无法应用数学模型对其进行描述，从工程应用的角度看，过于费时费力。在实际应用中，图像的主观质量评价方法受到了严重限制，甚至根本不适合于某些应用场合，如实时图像传输等领域。

### 2.3 客观评价方法

客观评价算法是指使用一个或多个图像的度量指标，建立与图像质量相关的数学模型让计算机自动计算得出图像质量。其目标是客观评价结果与人的主观感受相一致。常用的客观图像质量评价方法：

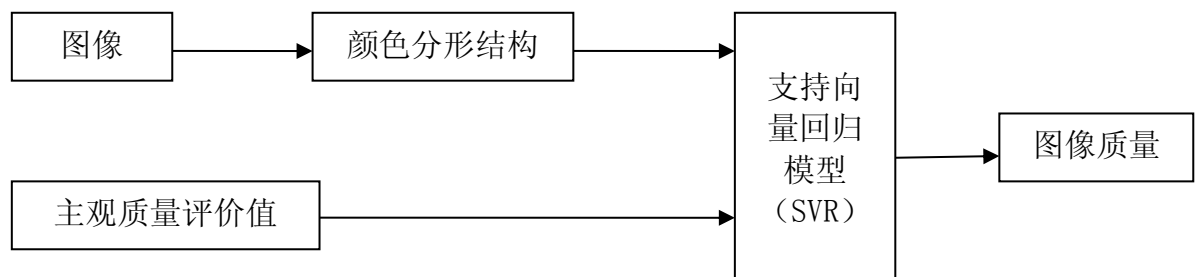
- (1) 均方误差方法 (Mean Squared Error, MSE)，最简单和最古老的方法，比较两幅图片，计算它们的均方差由此得到图片质量评价的结果。
- (2) 峰值信噪比 (PSNR, Peak Signal-to-Noise Ratio)<sup>[4]</sup>，PSNR 的分数无法和人眼看到的视觉品质完全一致，人眼的视觉对于误差的敏感度并不是绝对的，会受

到许多因素的影响而产生变化。

- (3) 结构相似度图像质量评价方法，自然图像信息是高度结构化的<sup>[5]</sup>，他们的像素之间表现出很强的相互依赖性，这种相互的依赖在空间近似时尤其明显，尽管大量的图像质量评价都是建立在线性的分解图像信号差异灵敏度的基础上，但其未去除像素之间的依赖性，因此，提出了基于结构相似度的图像质量评价（The Structural SIMilarity Index，SSIM）方法。

### 三. 方案设计

根据任务的要求以及基础原理，我们根据掌握的知识，对整个实验进行设计。在这个实验中首先采用颜色分形结构提取出图像的颜色特征，然后将得到的数据和主观评价一起作为支持向量回归模型（SVR）的输入量，经过支持向量回归模型（SVR）的训练和预测给出图像质量的评价。



#### 3.1 颜色分形结构模型

在分形结构模型中，不考虑空间和实践方面，将要研究自然图像的另一个重要方面，即颜色组成。在其他的研究中出现了表征自然图像颜色比色统计<sup>[6]</sup>。然而，基于统计特征建立起来的自然图像分形色彩组成是一个全新的方法<sup>[7] [8]</sup>。这个方法将会给彩色图像的建模及相关应用做出一定的贡献<sup>[9]</sup>。

- (1) 提取图片的 rgb 信息

代码见附录一

- (2) 颜色分形结构的特征提取

在 RGB 空间中的彩色图像有  $N$  个像素点， $X_n = (R, G, B)$  其中  $n = 1-N$ ，我们计算以任意像素点为参考点，在其周围距离范围为小于  $r$  的点的个数，其定义为：

$$C_2(r) = \sum_{n=1}^N \sum_{n' > n}^N \Gamma(r - \|X_n - X_{n'}\|)$$

在颜色块中最大的分布为  $[0, Q-1]$ <sup>3</sup>，当  $r=r_{\max} = \sqrt{3(Q-1)}$  时， $C_2(r)$  达到最大，此时  $N(N-1)/2$  为像素点的对数，对它归一化得到：

$$C_0(r) = \frac{C_2(r)}{(N(N-1)/2)}$$

$C_0(r)$  为一个非递减函数，当  $r=0$  时趋于零，当  $r$  趋于  $r_{\max}$  时趋于 1。

### 3.2 支持向量回归 (SVR) 模型

支持向量机 (support vector machine, SVM)，SVM 起初是用于解决线性可分情况下两类样本的分类问题 (SVC)，其核心思想是找到一个最优分类超平面  $w \cdot x + b = 0$ ，使两类样本的分类间隔最大化，SVR 与 SVC 相似，但是，SVR 所求超平面是使所有样本点到超平面的距离为最小。其寻优问题转化相应的二次规划问题为：

$$\min \frac{1}{2} \|w\|^2 + c \sum_{i=1}^n (\xi_i + \xi_i^*)$$

约束条件： (1)  $y_i - w \cdot x_i - b \leq \epsilon + \xi_i$

(2)  $w \cdot x_i + b - y_i \leq \epsilon + \xi_i^*$

(3)  $\xi_i, \xi_i^* \geq 0$

最优函数为：

$$f(x) = w \cdot x + b = \sum_{i=1}^n (\alpha_i - \alpha_i^*) (x \cdot x_i) + b$$

通常，特征空间具有很高甚至无穷维数，只是空间变换后计算量剧增而面临维数灾难等问题，而 SVM 中待解对偶问题只包含一个变换后特征空间内积运算，而这种运算能在原空间中通过核函数来实现，根据 Mercer 定理可构造系列核函数，常见如线性核 (t=0)、多项式核 (t=1, d=2)、多项式核 (t=1, d=3)、径向基核 (t=2) 和 sigmoid 核 (t=3) 等<sup>[10]</sup>。



## 四、算法实现分析

在上述整体模型中，颜色分形结构模型的算法使用 matlab 实现，而对于实现较为复杂的 SVR 算法，本设计采用对 python 的 libsvm 库直接封装的方式使用。

### 4.1 颜色分型结构模型算法

本算法主要是将图片转化为其颜色特征曲线，在实现上分为两个子函数：img2rgb 和 rgb2C, 其中 img2rgb 将图片转化为二维 rgb 数组，再通过 rgb2C 计算其颜色特征曲线。

img2rgb 代码如下：

```
function [rgb] = img2rgb(path, sample)
    %读入图片文件 size(rawFormatRgb) = x, y, 3
    rawFormatRgb = imread(path);
    [x, y, z] = size(rawFormatRgb);
    %将图片的 rgb 数据转化为较为容易处理的格式
    for i = 1:1:x
        for j = 1:1:y
            if mod(i, sample) == 0 && mod(j, sample) == 0
                item.r = rawFormatRgb(i, j, 1);
                item.g = rawFormatRgb(i, j, 2);
                item.b = rawFormatRgb(i, j, 3);
                rgb(i, j) = item;
            end
        end
    end
end
```

rgb2C 代码如下：

```
function [C] = rgb2C(rgb, r)
    len = length(r);
    C = zeros(1, len);
    [x, y] = size(rgb);
    for i = 1:1:x
        for j = 1:1:y
            %取任意一点 p
            item = rgb(i, j);
            p = [item.r, item.g, item.b];
            %计算满足距离 p 小于 r 的所有点的个数
            count = fit_count(p, rgb, r);
            %满足条件的点计数累加
            C = C + count;
        end
    end
end
N = x*y;
```

```

    %进行归一化
    C = C / (N * (N - 1) / 2);
end

function [d] = fit_count(p, rgb, r)
    [x, y] = size(rgb);
    len = length(r);
    d = zeros(1, len);
    for i = 1:1:x
        for j = 1:1:y
            item = rgb(i, j);
            p1 = [item.r, item.g, item.b];
            pd = p - p1;
            distance = sqrt(dot(pd, pd));
            for cnt = 1:1:len
                if distance <= r(cnt)
                    d(cnt) = d(cnt) + 1;
                end
            end
        end
    end
end
end
end
end

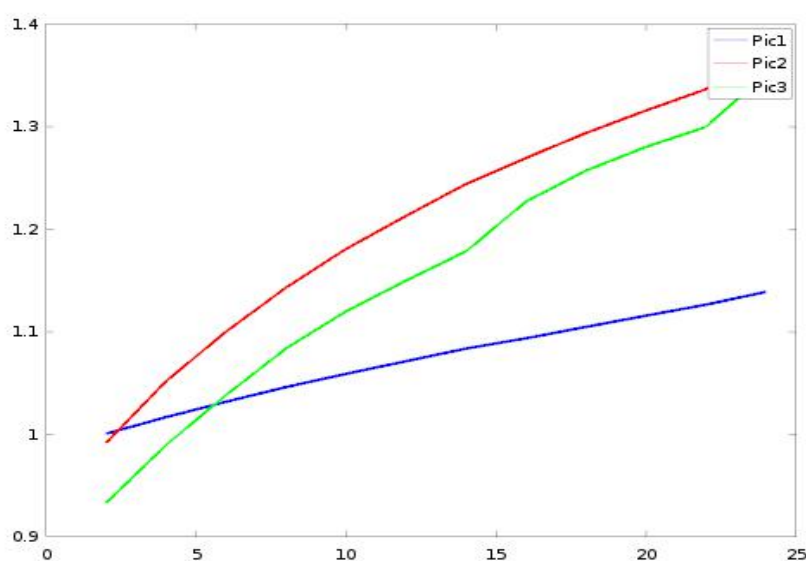
```

## 4.2 颜色分型结构例子

利用上述实现，给出三张图片 Pic1, Pic2, Pic3 提取其颜色特征曲线如下：

下面 3 张图片分别为：Pic1, Pic2, Pic3，图片和颜色特征曲线如下





### 4.3 SVR 算法

出三张使用 python 对 libsvm 进行封装，形成 SVR 类，代码如下：

```
from svmutil import *

class SVR(object):
    def __init__(self, data_path):
        #读取 matlab 产生的图片颜色特征数据 character 和主观评价 label
        label, character = svm_read_problem(data_path)
        self.y = label
        self.x = character

        #设置使用 epsilon-SVR 算法，线性核，损失参数 c 为 1
        self.parameter = svm_parameter('-s 3 -t 0 -c 1')
        self.model = svm_train(self.y, self.x, self.parameter)

    def predict(self, yt, xt):
        return svm_predict(yt, xt, self.model)
```

### 4.4 其他脚本

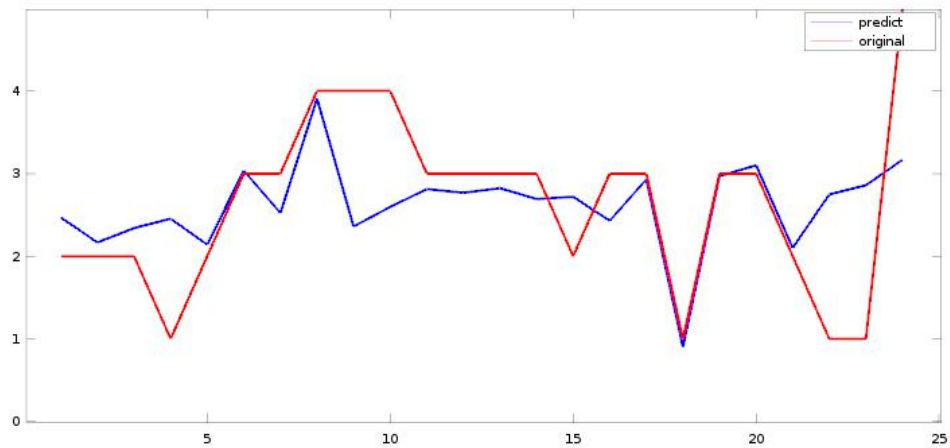
由于颜色特征曲线提取算法复杂度较大  $O(n^2)$ ，对于大图片处理十分费时，故而需要在处理前对图片进行统一压缩，而处理后的数据需要保存为 libsvm 所需要的格式，为了方便训练和测试过程，另外实现了数个脚本工具（代码见附录）。

## 五、结果与总结

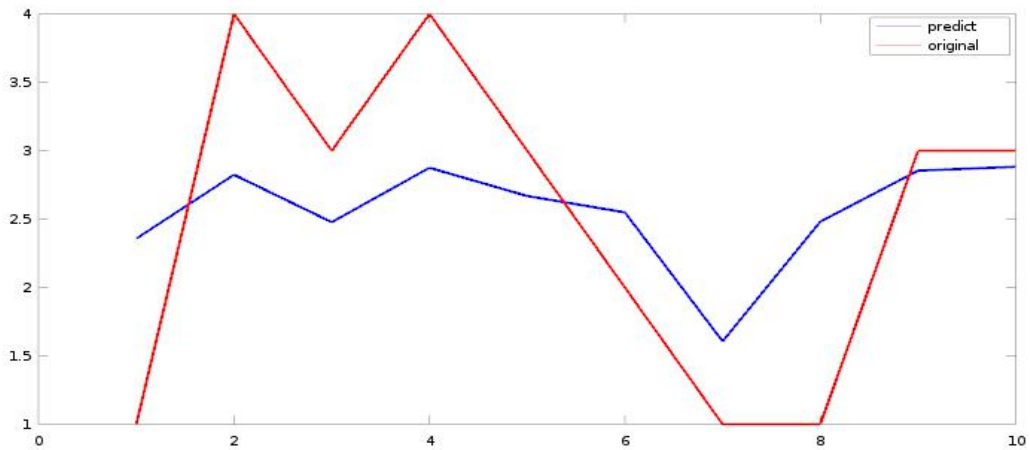
### 4.1 结果分析

随机选取照片 34 张，组员个人对其进行主观评价后取平均值，得到的数据用于训练和测试。其中 24 组数据用于进行 SVR 训练，10 组数据用于测试。训练曲线和测试曲线如下：

**SVR 训练曲线：**



**SVR 测试曲线：**



其中训练的均方差(MSE)为 0.764869，平方相关系数(SCC)为 0.284156；测试均方差(MSE)为 0.777196，平方相关系数(SCC)为 0.556655。

从测试结果看出，该模型具有一定的泛华能力，预测结果大体走势与实际主观评价结果相当，但仍有一定误差，并且在个别数据上出现了较大的分歧。由于颜色特征曲线的提取算法涉及大规模运算，难以获得大量样本数据，造成 SVR 过拟合，导致了该误

差,在后续工作中可以使用高性能服务器进行运算,提取大量样本的特征曲线进行拟合,从而得到更为优异的 SVR 模型。

详细数据见附录四。

## 4.2 总结

本次实验通过颜色分形结构提取彩色图像特征,构造 SVR 模型对彩色图像进行了质量评价。算法得到的结果和主观性的评价是比较接近的,但是由于颜色特征曲线的提取算法涉及大规模运算,难以获得大量样本数据,造成 SVR 过拟合,导致了实验结果存在一定的误差。需要改进的方面就是在后续工作中可以使用高性能服务器进行运算,提取大量样本的特征曲线进行拟合,从而得到更为优异的 SVR 模型。总体而言,实验结果也比较符合要求,整个实验是比较成功的。

我们小组成员这次在这个实验过程中进行了很好的合作分工,在实验过程中也多次进行商量讨论,针对出现的问题进行交流并解决,本人主要负责算法设计和实现,及数据处理和结果分析等工作。

## 五. 参考文献

- 【1】Wang Z., Bovik A. C, “A universal image quality index,” IEEE Signal Processing Letters. vol.9, no.3, pp. 81-84, 2002.
- 【2】Mark D. Fairchild, “Color Appearance Models,” USA: Munsell Color Science LaboratoryRochester Institute of Technology, 2005.
- 【3】周景超, 戴汝为, 肖柏华, 图像质量评价研究综述. 计算机科学 vol.35, pp.7-10, 2008.
- 【4】Huynh-Thu, M. Ghanbari, “Scope of validity of PSNR in image/video quality assessment,” Electronics Letters, vol. 44, pp. 800-801, 2008.
- 【5】Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli., “Image quality assessment: Fromerror visibility to structural similarity,” IEEE Trans. on Image Processing. vol.13, no.1,pp.600-612, Jan. 2004.
- 【6】W. S. Kim and R. H. Park, “Color image palette construction based on the HIS color system forminimizing the reconstruction error,” IEEE International Conference on Image Processing, vol.3, pp.1041-1044, 1996.
- 【7】Francis Chapeau-Blondeau, JulienChauveau, David Rousseau and Paul Richard, “Fractalstructure in the color distribution of natural images,” Chaos, Solitons & Fractals, vol. 42, no. 1, pp. 472-482, Oct. 2009.
- 【8】Mihai, Ivanovici and Noël Richard, “Fractal Dimension of Color Fractal Images,” IEEEtransactions on image processing vol. 20, no. 1, Jan. 2011.
- 【9】Mao X, Chen B and Muta I. “Affective property of image and fractal dimension,” Chaos, Solitons & Fractals vol.15, pp.905-910, 2003.
- 【10】Deng N Y and Tian Y J, “Support Vector Machine—A New Method in Data Mining,” Beijing:Science Press, pp.77-162, 2004.
- 【11】徐海松. 颜色信息工程, 浙江大学出版社, 2005 年 12 月出版
- 【12】VQEG. Final report from the video quality experts group on the validation of objective modelsof video quality assessment, <http://www.vqeg.org/>, 2000, 3.
- 【13】Winkler S.. Vision Models and Quality Metrics for Processing Applications, Ph.D. thesis.Diplom-Ingenieur der Elektrotechnik, Technische University Wien de nationality autrichienne,2000.

## 附录:

resize bash 脚本:

将图片统一压缩为 50x50 大小, 以减小颜色特征曲线提取的运算量

```
#!/bin/bash
files=$(ls data/)
for file in $files
do
    convert data/$file -resize 40x40 data/$file
done
```

test 脚本:

对 data/ 目录下的所有图片进行颜色特征曲线的提取, 并且将结果保存成为 libsvm 所需要的.svm 文件格式

```
function [result] = test()
    test_files = ls('data/');
    [files_num, max_str_width] = size(test_files);

    for cnt = 1:1:files_num
        item.file_name = test_files(cnt,:);
        item.C = file2C(strcat("data/", item.file_name));
        result(cnt) = item;
    End
    save_result(result, 'result.svm');
end

function [C] = file2C(path)
    r = 1:1:25;
    r = r * 2;
    rgb = img2rgb(path, 1);
    C = rgb2C(rgb, r);
end

function save_result(result, filepath)
    fd = fopen(filepath, 'w');
    len = length(result);
    for i = 1:1:len
        item = result(i);
        fprintf(fd, strcat(item.file_name, ","));
        cLen = length(item.C);
        for j = 1:1:cLen
            s = strcat(num2str(j), ":", num2str(item.C(j)), ",");
            fprintf(fd, s);
        end
    end
end
```

```
        fprintf(fd, "\n");
    end
    fclose(fd);
end
```

训练数据:

2 1:1.0006 2:1.0167 3:1.0316 4:1.046 5:1.0589 6:1.0713 7:1.0836 8:1.0936 9:1.1047 10:1.1157 11:1.1264  
12:1.139 13:1.1538 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017 21:2.0017  
22:2.0017 23:2.0017 24:2.0017 25:2.0017  
2 1:0.99157 2:1.0518 3:1.1001 4:1.1431 5:1.1808 6:1.213 7:1.2441 8:1.2696 9:1.2938 10:1.3158 11:1.3366  
12:1.3699 13:1.3993 14:2.0013 15:2.0013 16:2.0013 17:2.0013 18:2.0013 19:2.0013 20:2.0013 21:2.0013  
22:2.0013 23:2.0013 24:2.0013 25:2.0013  
2 1:0.9329 2:0.98948 3:1.0383 4:1.0837 5:1.1199 6:1.1498 7:1.1784 8:1.227 9:1.2572 10:1.2803 11:1.2998  
12:1.3517 13:1.3707 14:2.0014 15:2.0014 16:2.0014 17:2.0014 18:2.0014 19:2.0014 20:2.0014 21:2.0014  
22:2.0014 23:2.0014 24:2.0014 25:2.0014  
1 1:0.96787 2:1.0004 3:1.0291 4:1.0525 5:1.0733 6:1.0897 7:1.1052 8:1.1203 9:1.1396 10:1.1571  
11:1.1738 12:1.2056 13:1.2357 14:2.0023 15:2.0023 16:2.0023 17:2.0023 18:2.0023 19:2.0023 20:2.0023  
21:2.0023 22:2.0023 23:2.0023 24:2.0023 25:2.0023  
2 1:1.0068 2:1.0647 3:1.1157 4:1.1538 5:1.1879 6:1.2174 7:1.2465 8:1.2708 9:1.3183 10:1.3897 11:1.4621  
12:1.5868 13:1.6869 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017 21:2.0017  
22:2.0017 23:2.0017 24:2.0017 25:2.0017  
3 1:0.76872 2:0.78961 3:0.80976 4:0.82984 5:0.85038 6:0.86967 7:0.88892 8:1.0482 9:1.1036 10:1.1398  
11:1.169 12:1.3589 13:1.3953 14:2.0019 15:2.0019 16:2.0019 17:2.0019 18:2.0019 19:2.0019 20:2.0019  
21:2.0019 22:2.0019 23:2.0019 24:2.0019 25:2.0019  
3 1:0.87282 2:0.91832 3:0.96369 4:1.0067 5:1.0505 6:1.0885 7:1.1284 8:1.1819 9:1.2373 10:1.2822  
11:1.3224 12:1.3968 13:1.4403 14:2.0018 15:2.0018 16:2.0018 17:2.0018 18:2.0018 19:2.0018 20:2.0018  
21:2.0018 22:2.0018 23:2.0018 24:2.0018 25:2.0018  
4 1:0.35881 2:0.399 3:0.4414 4:0.48312 5:0.52517 6:0.56345 7:0.60109 8:1.1232 9:1.2054 10:1.2518  
11:1.2838 12:1.7984 13:1.8203 14:2.0013 15:2.0013 16:2.0013 17:2.0013 18:2.0013 19:2.0013 20:2.0013  
21:2.0013 22:2.0013 23:2.0013 24:2.0013 25:2.0013  
4 1:0.91101 2:0.98063 3:1.0383 4:1.0822 5:1.1181 6:1.1474 7:1.1735 8:1.2427 9:1.2778 10:1.3018  
11:1.3221 12:1.4554 13:1.5263 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017  
4 1:0.83117 2:0.88906 3:0.94176 4:0.9873 5:1.0282 6:1.0607 7:1.0913 8:1.1754 9:1.2441 10:1.2888  
11:1.3229 12:1.4301 13:1.4677 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017  
3 1:0.85306 2:0.87493 3:0.89687 4:0.91654 5:0.93653 6:0.95458 7:0.97282 8:1.0595 9:1.1016 10:1.1307  
11:1.1546 12:1.2793 13:1.3113 14:2.0013 15:2.0013 16:2.0013 17:2.0013 18:2.0013 19:2.0013 20:2.0013  
21:2.0013 22:2.0013 23:2.0013 24:2.0013 25:2.0013  
3 1:0.86543 2:0.88784 3:0.9101 4:0.93137 5:0.95281 6:0.97297 7:0.99368 8:1.0658 9:1.1113 10:1.1435  
11:1.1707 12:1.2832 13:1.3209 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017  
3 1:0.85003 2:0.87074 3:0.89126 4:0.91065 5:0.93064 6:0.94924 7:0.96839 8:1.0611 9:1.103 10:1.1329



11:1.1589 12:1.2806 13:1.3131 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017  
3 1:0.88256 2:0.90844 3:0.93381 4:0.95768 5:0.98155 6:1.0024 7:1.0239 8:1.0887 9:1.1312 10:1.1624  
11:1.1887 12:1.2776 13:1.3126 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017  
2 1:0.87946 2:0.90364 3:0.92652 4:0.94833 5:0.96993 6:0.99014 7:1.0102 8:1.0755 9:1.1226 10:1.1562  
11:1.1845 12:1.2782 13:1.3161 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017  
3 1:0.97701 2:1.004 3:1.0293 4:1.0529 5:1.076 6:1.0973 7:1.1189 8:1.1395 9:1.1634 10:1.1856 11:1.2062  
12:1.2375 13:1.2659 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017 21:2.0017  
22:2.0017 23:2.0017 24:2.0017 25:2.0017  
3 1:0.81212 2:0.83269 3:0.85243 4:0.87111 5:0.88993 6:0.90744 7:0.92552 8:1.0573 9:1.1016 10:1.1308  
11:1.155 12:1.3178 13:1.3491 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017  
1 1:1.3127 2:1.4486 3:1.5313 4:1.5836 5:1.6264 6:1.6599 7:1.6979 8:1.72 9:1.7465 10:1.7669 11:1.7835  
12:1.8031 13:1.837 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017 21:2.0017  
22:2.0017 23:2.0017 24:2.0017 25:2.0017  
3 1:0.78621 2:0.80933 3:0.83246 4:0.85513 5:0.87795 6:0.89951 7:0.92212 8:1.0551 9:1.1091 10:1.1453  
11:1.1751 12:1.3527 13:1.3905 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017  
3 1:0.63321 2:0.68494 3:0.73577 4:0.78341 5:0.83141 6:0.87446 7:0.91784 8:1.1555 9:1.2672 10:1.3433  
11:1.4044 12:1.6676 13:1.7306 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017  
2 1:1.0605 2:1.1226 3:1.1508 4:1.1722 5:1.1913 6:1.2085 7:1.2245 8:1.2466 9:1.2781 10:1.2942 11:1.3052  
12:1.3488 13:1.3726 14:2.0023 15:2.0023 16:2.0023 17:2.0023 18:2.0023 19:2.0023 20:2.0023 21:2.0023  
22:2.0023 23:2.0023 24:2.0023 25:2.0023  
1 1:0.87493 2:0.8971 3:0.91916 4:0.93981 5:0.96066 6:0.97912 7:0.99825 8:1.0721 9:1.1075 10:1.1334  
11:1.1559 12:1.2601 13:1.2907 14:2.0013 15:2.0013 16:2.0013 17:2.0013 18:2.0013 19:2.0013 20:2.0013  
21:2.0013 22:2.0013 23:2.0013 24:2.0013 25:2.0013  
1 1:0.83309 2:0.85548 3:0.87733 4:0.89837 5:0.91949 6:0.93923 7:0.95925 8:1.0547 9:1.1049 10:1.138  
11:1.165 12:1.3067 13:1.3445 14:2.0013 15:2.0013 16:2.0013 17:2.0013 18:2.0013 19:2.0013 20:2.0013  
21:2.0013 22:2.0013 23:2.0013 24:2.0013 25:2.0013  
5 1:0.68362 2:0.71529 3:0.74666 4:0.77699 5:0.80731 6:0.83576 7:0.86472 8:1.0699 9:1.1394 10:1.1855  
11:1.2231 12:1.4951 13:1.5404 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017

测试数据:

1 1:1.0157 2:1.0447 3:1.0672 4:1.0852 5:1.1013 6:1.1162 7:1.1283 8:1.1387 9:1.1518 10:1.1635 11:1.1747  
12:1.1919 13:1.2077 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017 21:2.0017  
22:2.0017 23:2.0017 24:2.0017 25:2.0017  
4 1:0.86132 2:0.87867 3:0.89576 4:0.91249 5:0.92938 6:0.94532 7:0.96141 8:1.0497 9:1.0869 10:1.113  
11:1.1352 12:1.2514 13:1.2814 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017  
3 1:0.98966 2:1.0081 3:1.025 4:1.0404 5:1.0562 6:1.0698 7:1.0853 8:1.0995 9:1.1141 10:1.1281 11:1.1415  
12:1.1619 13:1.182 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017 21:2.0017

22:2.0017 23:2.0017 24:2.0017 25:2.0017

4 1:0.82997 2:0.8505 3:0.87068 4:0.89025 5:0.91003 6:0.92884 7:0.94816 8:1.0537 9:1.1049 10:1.139  
11:1.1675 12:1.298 13:1.3328 14:2.0013 15:2.0013 16:2.0013 17:2.0013 18:2.0013 19:2.0013 20:2.0013  
21:2.0013 22:2.0013 23:2.0013 24:2.0013 25:2.0013

3 1:0.90139 2:0.92378 3:0.94571 4:0.96701 5:0.98837 6:1.0081 7:1.0289 8:1.0774 9:1.1179 10:1.1479  
11:1.1745 12:1.247 13:1.2808 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017

2 1:0.90796 2:0.94299 3:0.97686 4:1.0065 5:1.0352 6:1.0603 7:1.0868 8:1.1309 9:1.1807 10:1.214  
11:1.241 12:1.3 13:1.3319 14:2.0013 15:2.0013 16:2.0013 17:2.0013 18:2.0013 19:2.0013 20:2.0013  
21:2.0013 22:2.0013 23:2.0013 24:2.0013 25:2.0013

1 1:1.0706 2:1.1773 3:1.2469 4:1.3234 5:1.3885 6:1.4465 7:1.5016 8:1.5275 9:1.5533 10:1.5782 11:1.6038  
12:1.6329 13:1.6737 14:2.0023 15:2.0023 16:2.0023 17:2.0023 18:2.0023 19:2.0023 20:2.0023 21:2.0023  
22:2.0023 23:2.0023 24:2.0023 25:2.0023

1 1:0.97998 2:1.0004 3:1.0198 4:1.0382 5:1.0561 6:1.0718 7:1.0886 8:1.105 9:1.1235 10:1.1398 11:1.1555  
12:1.1812 13:1.2039 14:2.002 15:2.002 16:2.002 17:2.002 18:2.002 19:2.002 20:2.002 21:2.002 22:2.002  
23:2.002 24:2.002 25:2.002

3 1:0.8189 2:0.84443 3:0.86985 4:0.89416 5:0.91834 6:0.94104 7:0.96421 8:1.0745 9:1.1376 10:1.1788  
11:1.2121 12:1.3464 13:1.3862 14:2.0023 15:2.0023 16:2.0023 17:2.0023 18:2.0023 19:2.0023 20:2.0023  
21:2.0023 22:2.0023 23:2.0023 24:2.0023 25:2.0023

3 1:0.82642 2:0.84723 3:0.86786 4:0.88767 5:0.90782 6:0.92658 7:0.946 8:1.0589 9:1.1062 10:1.1383  
11:1.1652 12:1.3069 13:1.3416 14:2.0017 15:2.0017 16:2.0017 17:2.0017 18:2.0017 19:2.0017 20:2.0017  
21:2.0017 22:2.0017 23:2.0017 24:2.0017 25:2.0017