

第6章 数据分析和统计

在本章中将介绍 MATLAB 对数据处理和统计分析的命令。如果没有特别强调，本章中的 A 和 B 是指 $m \times n \times \cdots \times p$ 的多维矩阵， x 是一个向量。

6.1 最大值和最小值

用命令集 55 中列出的命令可以求解最大值。

命令集 55 最大值和最小值

<code>max(x)</code>	返回 x 中最大的元素值，如果 x 是复数，则返回 <code>max(abs(x))</code> 值。
<code>max(A)</code>	返回一个含有 A 中第 1 维最大值的 $1 \times n \times \cdots \times p$ 矩阵。对于二维矩阵来说，返回一个行向量，它的第 1 个元素是 A 中第 1 列的最大的元素；如果 A 为复数时，则返回 <code>max(abs(A))</code> 值。
<code>[y, ind]=max(A)</code>	返回一个含有 A 中第 1 维最大值的 $1 \times n \times \cdots \times p$ 矩阵 y ，并在行向量 <code>ind</code> 中保存每列的最大数的行下标。
<code>max(A, B)</code>	返回一个和 A 、 B 相同维数的矩阵，每一元素都是在 A 和 B 中的相同位置上是最大的元素。
<code>C=max(A, [], dim)</code>	给出在指定的 dim 维内 A 的最大分量。如 <code>max(A, [], 1)</code> ，则给出 A 中最大的行向量。
<code>min(x)</code>	返回向量 x 中最小的元素。该命令关于矩阵的操作和 <code>max</code> 一样，如果 x 是复数，则返回 <code>min(abs(x))</code> 值。

例 6.1

创建一个三维矩阵 A ：

```
A(:,:,1) = [1 2 3; 2 3 1; 3 2 1];
A(:,:,2) = [2 4 6; 4 6 2; 6 4 2];
```

显示结果为：

```
A(:,:,1) =
     1     2     3
     2     3     1
     3     2     1
A(:,:,2) =
     2     4     6
     4     6     2
     6     4     2
```

求其中的最大值可用命令：

```
max(A)
```

```
ans(:,:,1) =
    3     3     3
ans(:,:,2) =
    6     6     6
```

输入whos可获得A的详细信息：

Name	Size	Bytes	Class
A	3x3x2	144	double array
ans	1x3x2	48	double array

Grand total is 24 elements using 192 bytes

6.2 求和、乘积和差分

使用命令sum和cumsum可以求得各种不同的和。

命令集56 求和

sum(x)	返回向量x所有元素的和。
sum(A)	返回一个包含矩阵A各列元素之和的 $1 \times n \times \dots \times p$ 矩阵。
cumsum(x)	返回一个x中元素累计和的向量，也就是第2个元素是x中前两个元素之和，以此类推。
cumsum(A)	返回一个与A同样大小的矩阵，它的列是A中列的累计和。
cumsum(A,dim)	给出A中dim维的元素累计和，命令 cumsum(A) 和命令 cumsum(A,1) 相同。

例6.2

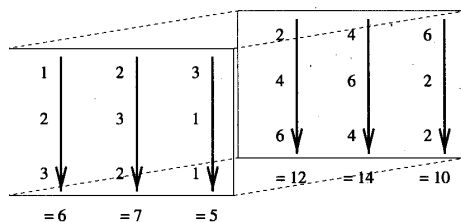
对例6.1中的矩阵A进行求和及累计和，其操作如下：

```
TheSum = sum(A), TheCsum = cumsum(A)
```

```
TheSum(:,:,1) =
    6     7     5
TheSum(:,:,2) =
   12    14    10

TheCsum(:,:,1) =
    1     2     3
    3     5     4
    6     7     5
TheCsum(:,:,2) =
    2     4     6
    6    10     8
   12    14    10
```

图6-1说明了MATLAB是如何来计算sum(A)的。各列相加后存放至 3×2 的三维数组TheSum中。

图6-1 三维数组中求和 $\text{sum}(A)$ 示意图

乘积的计算也和这相似。

命令集57 乘积

<code>prod(x)</code>	返回 x 中各元素乘积。
<code>prod(A)</code>	返回一个元素是列乘积的多维矩阵。
<code>prod(A,dim)</code>	给出 dim 维内的元素乘积。
<code>cumprod(x)</code>	返回一个 x 中各元素累积积的向量，也就是第2个元素是 x 中前两个元素的累积积，以此类推。
<code>cumprod(A)</code>	返回一个矩阵，其中列元素是 A 中列元素的累积积。
<code>cumprod(A, dim)</code>	给出在 dim 维内的累积积。

例 6.3

假设使用定义在例6.1中的矩阵 A ，执行命令：

```
TheProd = prod(A), TheCprod = cumprod(A)
```

返回得到：

```
TheProd(:, :, 1) =
    6    12     3
TheProd(:, :, 2) =
   48    96    24

TheCprod(:, :, 1) =
    1     2     3
    2     6     3
    6    12     3
TheCprod(:, :, 2) =
    2     4     6
    8    24    12
   48    96    24
```

使用命令`diff`可以进行差分计算，还有一些与命令`diff`相关的其他命令。

命令集58 差分和梯度

<code>diff(x)</code>	给出一个长度为 $n-1$ 的向量，它的元素是长度为 n 的向量 x 中相邻的
----------------------	---------------------------------------------

	元素的差。如果 $\mathbf{x}=(x_1, x_2, \dots, x_n)$, 则 $\text{diff}(\mathbf{x})=(x_2-x_1, x_3-x_2, \dots, x_n-x_{n-1})$ 。
<code>diff(A)</code>	在A的第一维内计算相邻元素的差分。对于二维矩阵来说, 就是 $\text{diff}(A)=A(2:m, :) - A(1:m-1, :)$ 。
<code>diff(x,k)</code>	求出第k次差分, <code>diff(x,2)</code> 和 <code>diff(diff(x))</code> 等价。
<code>diff(A,k,dim)</code>	在dim维内求出第k次差分。
<code>[DAdx,DAdy, DAdz,...]=</code>	在矩阵DAdx、DAdy、DAdz等中返回矩阵A的偏导数, 每个矩阵包含 $\partial A/\partial x$ 、 $\partial A/\partial y$ 、 $\partial A/\partial z$ 等相应的下标。在MATLAB中输入
<code>gradient(A)</code>	<code>help gradient</code> 可得到更多信息, 也可参见例 13.16。
<code>[DAdx,DAdy, DAdz,...]=</code>	返回偏导数 A/x 、 A/y 、 A/z 等, 如果给出参量 $h1, h2, h3, \dots$, 可将它们用作每个变量的步长。
<code>gradient(A,h1, h2,h3,...)</code>	
<code>del2(A)</code>	返回离散拉普拉斯算子, 矩阵中的元素为 A中元素和它相邻的四个元素的平均值的差分。

为了使函数 $z=f(x,y)$ 的梯度的四个极值形象化, 在颜色盘一节的图P-5中使用了`gradient`命令。

例6.4

差分计算很容易, 它还可以当作导数的近似值来用。

```
x = [1 4 9 16 25];
d1 = diff(x), d2 = diff(d1), d3 = diff(d2)
```

得到的结果为：

```
d1 =
     3     5     7     9

d2 =
     2     2     2

d3 =
     0     0
```

注意, 如果将计算得到的差分作为导数的近似值来用, 必须除以两点之间的距离。

6.3 统计命令

在前一节中提到了对矩阵列操作的命令, 比如 `max`、`min`、`sum`和`prod`。下面给出了数据统计分析的命令描述。

命令集59 平均值、中值和标准差

<code>mean(x)</code>	求出向量x的算术平均值。
----------------------	--------------

<code>mean(A,dim)</code>	给出一个 $1 \times n \times \dots \times p$ 的矩阵，它包含 A 中第 1 维的各个平均值。 如果给出了 <i>dim</i> ，就在 <i>dim</i> 维内计算。
<code>median(x)</code>	求出向量 x 中元素的中值。
<code>median(A,dim)</code>	给出一个 $1 \times n \times \dots \times p$ 的矩阵，它包含 A 中第 1 维各列的中值。如果给出了 <i>dim</i> ，就在 <i>dim</i> 维内计算。
<code>std(x)</code>	求出向量 x 中元素的标准差。
<code>std(A,dim)</code>	给出一个 $1 \times n \times \dots \times p$ 的矩阵，它包含 A 中第 1 维的各列标准差。如果给出了 <i>dim</i> ，就在 <i>dim</i> 维内计算标准差。

例6.5

令A为：

$$A = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 100 \end{pmatrix}$$

然后执行下列命令：

```
average = mean(A), med = median(A), dev = std(A)
```

返回得到：

```
average =
    2.5000    26.5000
```

```
med =
    2.5000    2.5000
```

```
dev =
    1.2910    49.0068
```

MATLAB中命令 `cov` 和 `corrcoef` 是用来求协方差和相关系数的，这些命令只能用在二维矩阵中。

命令集60 协方差和相关系数

<code>cov(x)</code>	求向量 x 的协方差。
<code>cov(A)</code>	求协方差矩阵，对角线元素是 A 中各列的方差。
<code>cov(x,y)</code>	等同于 <code>cov([x y])</code> ， x 和 y 是列向量。
<code>corrcoef(A)</code>	求相关矩阵。
<code>corrcoef(x,y)</code>	等同于 <code>corrcoef([x y])</code> ， x 和 y 是列向量。

例6.6

假设定义如下向量：

$$\mathbf{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} \quad \mathbf{z} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$$

(a) 通过下列命令来求得方差：

```
varx = cov(x), vary = cov(y), varz = cov(z)
```

```
varx =  
0
```

```
vary =  
0.3333
```

```
varz =  
1
```

(b) 协方差为：

```
Cvxy = cov(x,y), Cvxz = cov(x,z), Cvyz = cov(y,z)
```

```
Cvxy =  
0 0  
0 0.3333
```

```
Cvxz =  
0 0  
0 1
```

```
Cvyz =  
0.3333 0  
0 1.0000
```

(c) 通过下列命令来求得相关矩阵：

```
Corrxy = corrcoef(x,y), Corrxz = corrcoef(x,z), ...  
Corryz = corrcoef(y,z)
```

Warning: Divide by zero.

> In /opt/matlab52/toolbox/matlab/datafun/corrcoef.m at line 31

```
Corrxy =  
NaN NaN  
NaN 1
```

Warning: Divide by zero.

> In /opt/matlab52/toolbox/matlab/datafun/corrcoef.m at line 31

```
Corrxz =  
NaN NaN  
NaN 1
```

```
Corryz =  
1 0  
0 1
```

6.4 排序

在MATLAB中可以用命令`sort`来进行数据排序。

命令集61 排序

<code>sort(x)</code>	返回一个向量 x 的元素按递增排序的向量。如果元素是复数，则使用绝对值进行排序，即 <code>sort(abs(x))</code> 。
<code>[y,ind]=sort(x)</code>	返回下标向量 ind 。就是 <code>y=x(ind)</code> 。另外向量 y 是 x 中元素按递增排序得到的。
<code>sort(A,dim)</code>	对 A 中各列按递增排序，注意矩阵的行已被改变。如果给出了 <i>dim</i> ，则在 <i>dim</i> 维内进行排序。
<code>[B, Ind]=sort(A)</code>	返回矩阵 Ind 和矩阵 B ，矩阵 B 的列为矩阵 A 中按递增排序的列，矩阵 Ind 的每列相对应于上面提到的向量中列 ind 。
<code>sortrows(X,col)</code>	对矩阵 A 的各行按递增排序。如果行的元素是复数，它们以 <code>abs(x)</code> 为主，以 <code>angle(x)</code> 为辅进行排序。如果给出 <i>col</i> ，则根据指定的列数对行进行排序。

例6.7

假设矩阵 **A** 为：

$$\mathbf{A} = \begin{pmatrix} 0 & 4 & 4 \\ 2 & 0 & 2 \\ 4 & 2 & 0 \end{pmatrix}$$

(a) 执行命令 `[Ascend,Ind]=sort(A)`，结果为：

Ascend =

```
0      0      0
2      2      2
4      4      4
```

Ind =

```
1      2      3
2      3      2
3      1      1
```

(b) 如果以递减排序，可以使用下列命令：

`Descend=flipud(sort(A))`，执行后的结果为：

Descend =

```
4      4      4
2      2      2
0      0      0
```

在4.1节定义了命令 `flipud`。

6.5 统计频数直方图和棒图

使用命令 `hist`、`bar` 和 `stairs`，将数据集合以统计频数直方图和棒图显示出来。

命令集62 统计频数直方图和棒图

<code>hist(x)</code>	在10个等分点内画出x中数据的统计频数直方图。
<code>hist(x,n)</code>	在n个等分点内画出x中数据的统计频数直方图。
<code>hist(x,y)</code>	在由向量y定义的等分点内画出x中数据的统计频数直方图，向量y中元素按递增排序。
<code>bar(x)</code>	画出x的棒图。
<code>bar(z,x)</code>	在由向量z定义的位置上画出的棒图，z中的值必须是递增的均一分布。
<code>bar(x,...,str)</code>	画出如上的棒图，但可根据字符串str来设定颜色和形状。有关str的值，可参见13.1节。
<code>bar(A)</code>	画出以行分组的二维矩阵A的棒图。
<code>stairs(x)</code>	画出阶梯图，也就是内部没有线条的棒图。
<code>stairs(z,x)</code>	在由向量z定义的位置上画出储存在x中数据的棒图。
<code>barh(x,A,format)</code>	把 $m \times n$ 矩阵A画成m组n个棒图，可以通过字符串format来指定颜色，参见13.1节；或者用字符串'stacked'表示把同一列的数据画在一个直方条上。
<code>barh(A)</code>	和barh一样，但是 $x=1:m$ 。
<code>stem(y)</code>	在x轴上画y的离散火柴杆图，以小圆代表数据，并以其为结束。
<code>stem(z,y)</code>	在x轴上由向量x指定的位置上画y的离散火柴杆图，以小圆代表数据，并以其为结束。
<code>pareto(y,x)</code>	按递减顺序画出向量y的棒图。给出的向量x可作为x轴的坐标。如果x坐标没有给出，则可用向量y中元素的下标。该命令也是用元素的累加和来画直线。
<code>pie(x,extract)</code>	画向量x的饼图。如果 $\text{sum}(x) \leq 1$ ，则画出的是一个不完整的饼图，向量extract的大小和x一样；从x中取出与extract中每个非零的元素相同的元素。

命令hist、bar和stairs还可以用来在向量中存储数据。这些命令中的一些命令如bar和pie还可作用于三维数组。

命令集63 图表

<code>[m,y]=hist(x)</code>	在x的最大值和最小值之间等分成10个区间，在这个区间上画出统计频数直方图。向量y的元素为将 $\min(x)$ 和 $\max(x)$ 之间分成10个等间距的值，向量m为在每个区间内值的个数。这个直方图也可用命令 <code>bar(y,m, '.')</code> 来画。
<code>[m,y]=hist(x,n)</code>	在n个等间距区间上画统计频数直方图。
<code>[m,y]=hist(x,y)</code>	在由向量y指定的区间上画统计频数直方图。
<code>[xb,yb]=bar(y)</code>	画y的棒图，这个棒图可以用命令 <code>plot(xb,yb)</code> 来画。
<code>[xb,yb]=bar(x,y)</code>	在由向量x指定的位置上画y的棒图
<code>[xb,yb]=stairs(y)</code>	画y的阶梯图。
<code>[xb,yb]=stairs(x,y)</code>	在向量x定义的区间上画y的阶梯图。

例6.8

假设x为：

```
x = [1 1 3 4 5 1 9 8];
```

(a) 输入`hist(x); title('Histogram of x using hist'(x))`

运行的结果如图6-2所示。

命令`title`可在图上添加文本标题，参见13.3节。

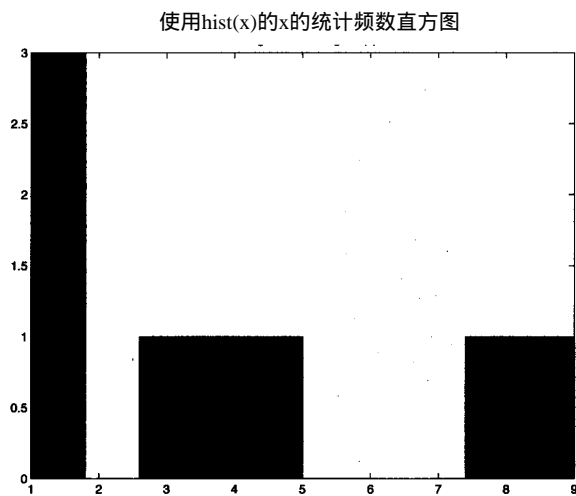


图6-2 在标准区间内的统计频数直方图

(b) 要画出在三个区间内的统计频数直方图，可输入：

```
hist(x,3); title('Histogram of x using hist(x,3)')
```

结果如图6-3所示：

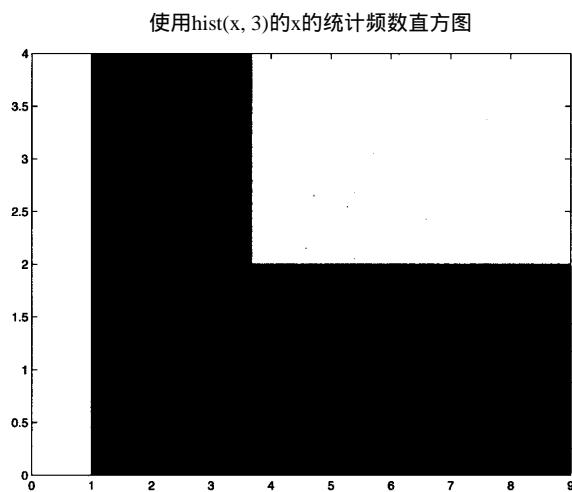


图6-3 在三个区间内的统计频数直方图

(c) 画棒图可以输入：

```
bar(x); title('bar(x)');
```

给出的图形如图6-4所示。

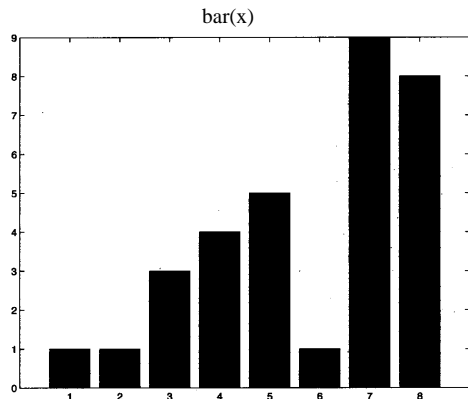


图6-4 x的棒图

(d) 如果输入 `[m,y]=hist(x)` ; MATLAB将创建出向量 **m**和**y**。如果再用命令 `bar(y,m,'w')` , 将再画出统计频数直方图; 见图 6-5。为了使图形更有趣, 可以画出白色的图形。用命令`plot`能画出其他色彩的图形来, 参见 13.1节。

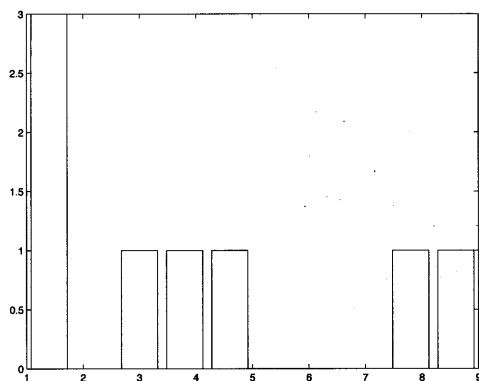


图6-5 用bar命令画的统计频数直方图

(e) 用命令`stem(x)`来画出向量x的火柴杆图, 结果如图6-6所示。

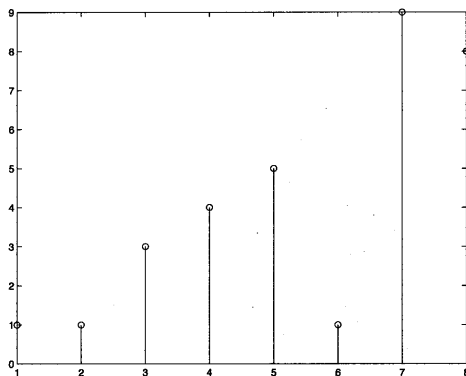


图6-6 用stem画的x火柴杆图

如果再定义一个向量：

```
xvalues = [1.1 1.3 2 2.4 2.5 1.8 3 3.2];
```

用它来做 x 轴的坐标值，运行命令 `stem(xvalues,x,'-')` 后可得到如图6-7所示的图形。

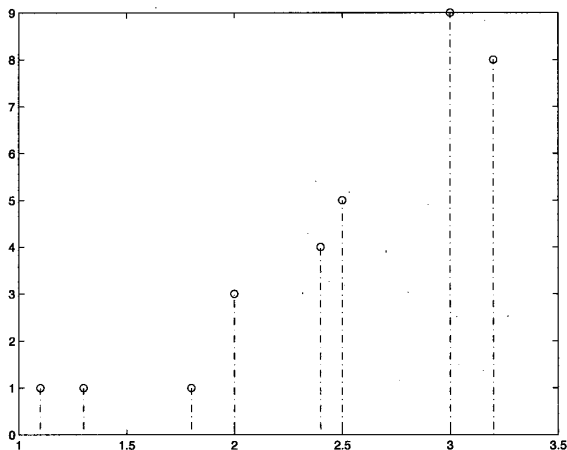


图6-7 以向量 `xvalues` 为 x 轴坐标画出的 x 数据火柴杆图

注意，向量 x 的元素不必按递增排列。命令 `stem` 的第3个参数用来确定线型，就象命令 `bar` 在画图时可指定颜色一样。

6.6 区域的三角分解

MATLAB中有下列关于区域三角形的命令。

命令集64 三角分解

`TRI=delaunay(x,y,'sorted')` 画一个三角形将向量 x 和 y 连接起来。如果给定参数 `'sorted'`，则假定没有重复数据，对 y 中的数据按递增排序。如果 y 的数据相同，则对 x 中的数据按递增排序。

`voroni(x,y,TRI)` 画集合 x 和 y 的 Voronoi 图形。如果给定 **TRI**，则画出集合 x 和 y 的 Delaunay 三角形图形。

有两个使用 Delaunay 三角形的函数，它们能给出与三角集合相关的集合信息。

命令集65 三角分解时的搜索函数

`dsearch(x,y,TRI,px,py)` 找到最接近 (px, py) 的由向量 x 和 y 定义的点的下标，矩阵 **TRI** 是 x 和 y 的三角矩阵。

`tsearch(x,y,TRI,px,py)` 找到由向量 x 和 y 形成的集合定义的三角形下标和最接近点 (px, py) 的三角矩阵 **TRI**，它是 x 和 y 的三角矩阵。

6.7 多边形分析

以上是两个关于多边形属性的函数。

命令集66 多边形

<code>polyarea(x,y)</code>	画一个由A和B的列组成的集合定义的多边形。如果给定 <i>dim</i> , 则画出定义在 <i>dim</i> 维内的多边形。
<code>polyarea(A,B,dim)</code>	画一个由A的第1维组成的集合定义的多边形。如果给定 <i>dim</i> , 则画出定义在 <i>dim</i> 维内的多边形。
<code>IN=inpolygon(x,y,px,py)</code>	返回一个和x,y大小相同的向量IN。如果点(x,y)在由px和py定义的多边形内, 则将IN中相等元素赋值为1; 如果点在多边形上, 则赋值为0.5; 在多边形外, 则赋值为0。
<code>rectint(x,y)</code>	画由向量x和y定义的矩形。
<code>rectint(A,B)</code>	从 <code>rectint(A(i,:),B(j,:))</code> 中返回所有可能组合的 $n \times m$ 矩阵, 如果A是一个 $n \times 4$ 的矩阵, B是一个 $m \times 4$ 的矩阵。
<code>convhull(x,y,TRI)</code>	返回由x和y定义的点的下标, 这个点在集合的凸起的位置上。如果给定了TRI, 则用它; 否则计算三角形。

例6.9

用矩阵DartBoard可以定义一个方形的镖盘。

```
DartBoard(1,:,1) = [ 2 3 3 2 2 ];
DartBoard(2,:,1) = [ 2 2 3 3 2 ];
DartBoard(1,:,2) = [ 1 4 4 1 1 ];
DartBoard(2,:,2) = [ 1 1 4 4 1 ];
```

见图6-8。

现在有：

```
DartBoard(:, :, 1) =
    2    3    3    2    2
    2    2    3    3    2

DartBoard(:, :, 2) =
    1    4    4    1    1
    1    1    4    4    1
```

下面的函数文件对于了解一个训练有素的镖手很有用。

```
function answer = dartresult(x,y, DartBoard)
% Returns a value corresponding to the score.

answer =
    2*sum(inpolygon(x,y,DartBoard(1,:,1),DartBoard(2,:,1)))+...
    3*sum(inpolygon(x,y,DartBoard(1,:,2),DartBoard(2,:,2)));
```

现在让MATLAB投一些镖。

```
RandomDarts(1,:) = 5.*rand(1,5);
RandomDarts(2,:) = 5.*rand(1,5)
```

```
RandomDarts =
```

```
4.7506    1.1557    3.0342    2.4299    4.4565
3.8105    2.2823    0.0925    4.1070    2.2235
```

可以得到一个好看的图形和有以下序列的结果，在图上把镖作为星画出来，在镖盘周围是空条纹。最后的结果放在图的上边。

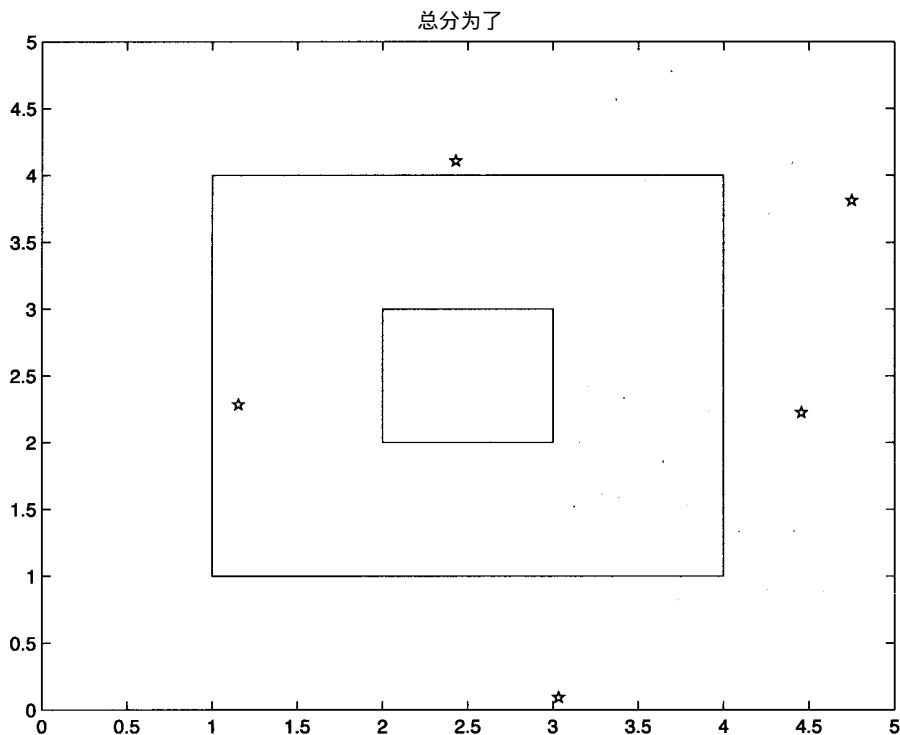


图6-8 MATLAB投镖的结果图

```
plot(DartBoard(1,:,1),DartBoard(2,:,1),'r',...
     DartBoard(1,:,2),DartBoard(2,:,2),'b')
hold on
plot(RandomDarts(1,:),RandomDarts(2,:), 'pentagram')
axis([0 5 0 5])
title(cat(2,'The total score is ',num2str(...
dartresult(RandomDarts(1,:),RandomDarts(2,:),DartBoard))))
```

结果如图6-8所示。