



解方程组

线性方程组误差分析

胡建芳

hujf5@mail.sysu.edu.cn

计算机学院

课程回顾

■ 高斯消元法与LU分解:

高斯消元法

两个步骤：消元，回代

复杂度：消元 $O(n^3/3)$ ，回代 $O(n^2/2)$

LU分解

概述：满足一定条件的矩阵，可以分解为上三角和下三角矩阵的乘积， $A=LU$

条件：前 $n-1$ 个顺序主子式不为0.

两个问题：

怎样的方程组可以用高斯消元法求解？

高斯消元法和LU分解的各自优劣势

解方程组误差

■ 向量的范数:

欧氏空间中向量长度在高维空间的拓展，满足三个条件：

(1) 非负性：当 $\alpha \neq 0$, $\|\alpha\| > 0$ 只有且仅有当 $\alpha = 0$, $\|\alpha\| = 0$

(2) 齐次性： $\|k\alpha\| = |k|\|\alpha\|$, k 为任意数。

(3) 三角不等式：对于 V 中的任意两个向量, β 都有

$$\|\alpha + \beta\| \leq \|\alpha\| + \|\beta\|$$

解方程组误差

■ 几种常用向量范数:

- 向量的“2”范数
(欧几里德范数)

$$\|x\|_2 = \sqrt{x_1^2 + \dots + x_n^2} = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}$$

- 向量的“1”范数

$$\|x\|_1 = |x_1| + \dots + |x_n| = \sum_{i=1}^n |x_i|$$

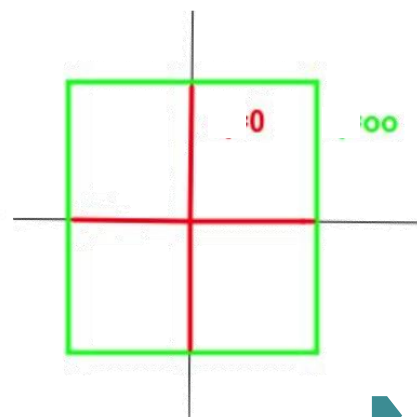
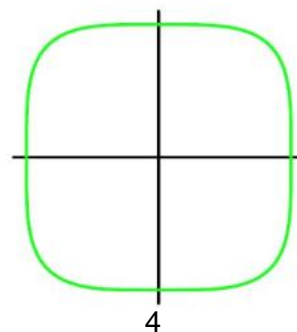
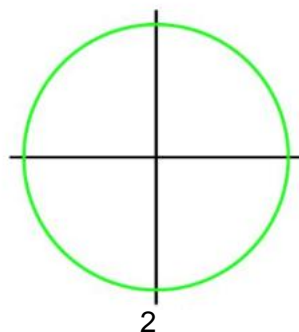
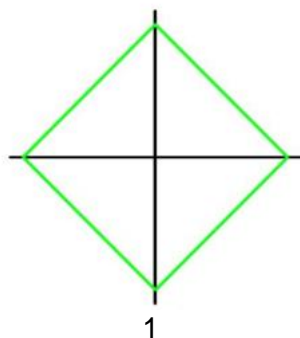
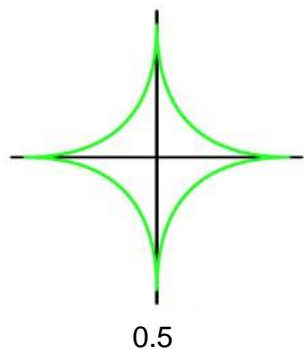
- 向量的“ ∞ ”范数
(极大值范数或一致向量范数)

$$\|x\|_\infty = \max\{|x_1|, \dots, |x_n|\} = \max_{1 \leq i \leq n} \{|x_i|\}$$

- 向量的“ p ”范数

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

1, 证明它们是范数
2, 0范数的特别意义



解方程组误差

■ 矩阵范数:

与向量范数类似, 满足正定性, 正齐次性和三角不等式

(1) 非负性: 当 $A \neq 0$, $\|A\| > 0$ 只有且仅有当 $A = 0$, $\|A\| = 0$

(2) 齐次性: $\|kA\| = |k|\|A\|$, k 为任意复数。

(3) 三角不等式: 对于任意两个同种形状矩阵 A, B 都有

$$\|A + B\| \leq \|A\| + \|B\|$$

解方程组误差

■ 几种常用矩阵范数:

$$\|A\|_1 = \max_j \sum_{i=1}^m |a_{i,j}|$$

1-范数:

, 列和范数, 即所有矩阵列向量绝对值之和的最大值, matlab调用函数`norm(A, 1)`。

2-范数: $\|A\|_2 = \sqrt{\lambda_1}$, λ_1 为 $A^T A$ 的最大特征值。

, 谱范数, 即 $A^T A$ 矩阵的最大特征值的开平方。matlab调用函数`norm(x, 2)`。

证明它们是范数

$$\|A\|_\infty = \max_i \sum_{j=1}^N |a_{i,j}|$$

∞ -范数: 行和范数, 即所有矩阵行向量绝对值之和的最大值, matlab调用函数`norm(A, inf)`。

$$\|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2 \right)^{\frac{1}{2}}$$

F-范数:

, Frobenius范数, 即矩阵元素绝对值的平方和再开平方, matlab调用函数`norm(A, 'fro')`。

还可以按行列组合定义范数, 比如2,1范数

解方程组误差

■ 矩阵范数:

(1) 如果 $A = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_n]$, 那么

$$\|A\|_F^2 = \sum_{i=1}^n \|\alpha_i\|_2^2$$

$$(2) \|A\|_F^2 = \text{TR}(A^H A) = \sum_{i=1}^n \lambda_i(A^H A)$$

(3) 对于任何 m 阶酉矩阵 U 与 n 阶酉矩阵 V 都有等式

$$\begin{aligned} \|A\|_F &= \|UA\|_F = \|A^H\|_F \\ &= \|AV\|_F = \|UAV\|_F \end{aligned}$$

解方程组误差

■ 矩阵范数与向量范数的联系：

$$\|\mathbf{Ax}\| \leq \|\mathbf{A}\| \|\mathbf{x}\|$$

$$\|\mathbf{A}\|_p = \max_{\|\mathbf{x}\|_p \neq 0} \frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p} = \max_{\|\mathbf{x}\|_p = 1} \|\mathbf{Ax}\|_p$$

$$\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\| \quad \text{矩阵的相容性}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 5 & 1 \\ -1 & 2 & -3 \\ 1 & -7 & 0 \end{bmatrix}$$

矩阵 \mathbf{A} 的2范数、0范数、无穷范数分别是？

求出满足 $\|\mathbf{A}\|_\infty = \|\mathbf{Ax}\|_\infty / \|\mathbf{x}\|_\infty$ 的一个向量 \mathbf{x} .

求出满足 $\|\mathbf{A}\|_1 = \|\mathbf{Ax}\|_1 / \|\mathbf{x}\|_1$ 的一个向量 \mathbf{x} .

解方程组误差

■ 误差:

定义 2.4 设 x_c 为线性方程组 $Ax = b$ 的近似解, 则残差(residual) 是向量 $r = b - Ax_c$. 后向误差是残差范数 $\|b - Ax_c\|_\infty$, 前向误差是 $\|x - x_c\|_\infty$.

例 2.10 求出方程组

$$\begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

的近似解 $x_c = [1, 1]$ 的后向误差与前向误差.

正确的解是 $x = [2, 1]$. 按无穷大范数, 后向误差为

$$\|b - Ax_c\|_\infty = \left\| \begin{bmatrix} 3 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\|_\infty = \left\| \begin{bmatrix} 1 \\ 3 \end{bmatrix} \right\|_\infty = 3$$

前向误差为

$$\|x - x_c\|_\infty = \left\| \begin{bmatrix} 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\|_\infty = \left\| \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\|_\infty = 1.$$

后向误差与前向
误差比较接近

解方程组误差

■ 误差:

例 2.11 求出方程组

$$\begin{cases} x_1 + x_2 = 2, \\ 1.0001x_1 + x_2 = 2.0001 \end{cases}$$

近似解 $[-1, 3.0001]$ 的前向误差与后向误差.

$$\left[\begin{array}{cc|c} 1 & 1 & 2 \\ 1.0001 & 1 & 2.0001 \end{array} \right] \xrightarrow{\text{第2行减去 } 1.0001 \times \text{第1行}} \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & -0.0001 & -0.0001 \end{array} \right]$$

求解相应的方程组

$$\begin{cases} x_1 + x_2 = 2, \\ -0.0001x_2 = -0.0001, \end{cases}$$

得到解 $[x_1, x_2] = [1, 1]$.

解方程组误差

■ 误差:

例 2.11 求出方程组

$$\begin{cases} x_1 + x_2 = 2, \\ 1.0001x_1 + x_2 = 2.0001 \end{cases}$$

近似解 $[-1, 3.0001]$ 的前向误差与后向误差.

后向误差是向量

$$\begin{aligned} \mathbf{b} - \mathbf{A}\mathbf{x}_c &= \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 1.0001 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 3.0001 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix} - \begin{bmatrix} 2.0001 \\ 2 \end{bmatrix} = \begin{bmatrix} -0.0001 \\ 0.0001 \end{bmatrix} \end{aligned}$$

的无穷大范数, 为 0.0001 . 前向误差是差

$$\mathbf{x} - \mathbf{x}_c = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -1 \\ 3.0001 \end{bmatrix} = \begin{bmatrix} 2 \\ -2.0001 \end{bmatrix}$$

的无穷大范数, 为 2.0001 .

**后向误差与前向
误差相差较大**

解方程组误差

■ 相对误差:

记残差为 $r = b - Ax_c$, 则方程组 $Ax = b$ 的相对后向误差定义为

$$\frac{\|r\|_{\infty}}{\|b\|_{\infty}},$$

相对前向误差定义为

$$\frac{\|x - x_c\|_{\infty}}{\|x\|_{\infty}},$$

$Ax = b$ 的误差放大因子(error magnification factor) 是两者的比率, 或

$$\text{误差放大因子} = \frac{\text{相对前向误差}}{\text{相对后向误差}} = \frac{\frac{\|x - x_c\|_{\infty}}{\|x\|_{\infty}}}{\frac{\|r\|_{\infty}}{\|b\|_{\infty}}}.$$

解方程组误差

■ 相对误差:

相对后向误差为

$$\frac{0.000\ 1}{2.000\ 1} \approx 0.000\ 05 = 0.005\%,$$

相对前向误差为

$$\frac{2.000\ 1}{1} = 2.000\ 1 \approx 200\%,$$

误差放大因子为 $2.000\ 1 / (0.000\ 1 / 2.000\ 1) = 40\ 004.000\ 1$.

解方程组误差

■ 条件数:

定义 2.5 一个方阵 A 的条件数 $\text{cond}(A)$ 是对所有的右端项 b 求解 $Ax = b$ 的最大可能误差放大因子.

定理 2.6 $n \times n$ 矩阵 A 的条件数是

$$\text{cond}(A) = \|A\| \cdot \|A^{-1}\|.$$

$$A = \begin{bmatrix} 1 & 1 \\ 1.0001 & 1 \end{bmatrix}$$

的范数是 $\|A\| = 2.0001$, A 的逆是

$$A^{-1} = \begin{bmatrix} -10000 & 10000 \\ 10001 & -10000 \end{bmatrix},$$

其范数 $\|A\|^{-1} = 20001$, A 的条件数为

$$\text{cond}(A) = (2.0001)(20001) = 40004.0001.$$

**条件数越大,
方程解越不可靠,
为病态方程组。**

解方程组误差

■ 条件数与扰动:

b 的扰动

$$A(x + \Delta x) = b + \Delta b \Rightarrow \Delta x = A^{-1} \Delta b$$

$$\|\Delta x\| \leq \|A^{-1}\| \|\Delta b\|$$

$$\|b\| = \|Ax\| \leq \|A\| \|x\| \Rightarrow \frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|}$$

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\| \|A^{-1}\| \frac{\|\Delta b\|}{\|b\|}$$

A 的扰动

$$(A + \Delta A)(x + \Delta x) = b \Rightarrow \Delta x = -A^{-1} \Delta A (x + \Delta x) \Rightarrow \frac{\|\Delta x\|}{\|x + \Delta x\|} \leq \|A\| \|A^{-1}\| \frac{\|\Delta A\|}{\|A\|}$$

同时扰动

$$(A + \Delta A)(x + \Delta x) = b + \Delta b$$

$$\|A^{-1}\| \|\Delta A\| \leq 1$$

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\|A\| \|A^{-1}\|}{1 - \|A\| \|A^{-1}\| \frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right)$$

解方程组误差

■ 淹没:

例 2.13 考虑方程组

$$\begin{cases} 10^{-20}x_1 + x_2 = 1, \\ x_1 + 2x_2 = 4. \end{cases}$$

我们将 3 次求解这个方程组：一次用完全精度，第二次根据 IEEE 双精度运算模拟计算机进行求解，还有一次是我们首先交换方程的次序。

1. 精确解 高斯消元过程表形式为

$$\left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 1 & 2 & 4 \end{array} \right] \xrightarrow{\substack{\text{第2行减去} \\ 10^{20} \times \text{第1行}}} \left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 0 & 2 - 10^{-20} & 4 - 10^{20} \end{array} \right],$$

底下的方程为

$$(2 - 10^{20})x_2 = 4 - 10^{20} \rightarrow x_2 = \frac{4 - 10^{20}}{2 - 10^{20}},$$

上面的方程得到

$$10^{-20}x_1 + \frac{4 - 10^{20}}{2 - 10^{20}} = 1 \rightarrow x_1 = 10^{20} \left(1 - \frac{4 - 10^{20}}{2 - 10^{20}} \right) \rightarrow x_1 = \frac{-2 \times 10^{20}}{2 - 10^{20}}.$$

精确解为

$$[x_1, x_2] = \left[\frac{2 \times 10^{20}}{10^{20} - 2}, \frac{4 - 10^{20}}{2 - 10^{20}} \right] \approx [2, 1].$$

解方程组误差

■ 淹没:

3. IEEE 双精度, 经行交换 经过交换两个方程的次序后, 我们重复高斯消元过程的计算机格式:

$$\left[\begin{array}{cc|c} 1 & 2 & 4 \\ 10^{-20} & 1 & 1 \end{array} \right] \xrightarrow{\substack{\text{第 2 行减去} \\ 10^{-20} \times \text{第 1 行}}} \left[\begin{array}{cc|c} 1 & 2 & 4 \\ 0 & 1 - 2 \times 10^{-20} & 1 - 4 \times 10^{-20} \end{array} \right]$$

按 IEEE 双精度, $1 - 2 \times 10^{-20}$ 被储存为 1, $1 - 4 \times 10^{-20}$ 也被储存为 1. 方程组现在是

$$\begin{cases} x_1 + 2x_2 = 4, \\ x_2 = 1, \end{cases}$$

这导出计算解 $x_1 = 2$ 和 $x_2 = 1$.

解方程组误差

■ 淹没:

2. IEEE 双精度 高斯消元过程的计算机格式稍微有些不同:

$$\left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 1 & 2 & 4 \end{array} \right] \xrightarrow{\substack{\text{第 2 行减去} \\ 10^{20} \times \text{第 1 行}}} \left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 0 & 2 - 10^{20} & 4 - 10^{20} \end{array} \right].$$

按 IEEE 双精度, 由于舍入误差, $2 - 10^{20}$ 与 -10^{20} 是一样的. 类似地, $4 - 10^{20}$ 按 -10^{20} 被储存. 现在底下的方程是

$$-10^{20}x_2 = -10^{20} \rightarrow x_2 = 1.$$

上面方程的机器运算格式变成了

$$10^{-20}x_1 + 1 = 1,$$

所以 $x_1 = 0$, 计算出的解恰好为 $[x_1, x_2] = [0, 1]$.

与精确解相比, 这个解有较大的相对误差.

高斯消元中的乘子须尽可能的小

作业

■ 作业:

就方程组 $x_1 + 2x_2 = 3$, $2x_1 + 4.01x_2 = 6.01$ 的下列近似解, 求出相对前向误差和相对后向误差以及误差放大因子:

- (a) $[-10, 6]$; (b) $[-100, 52]$; (c) $[-600, 301]$; (d) $[-599, 301]$;
(e) 系数矩阵的条件数是什么?

$$A = \begin{bmatrix} 1 & 5 & 1 \\ -1 & 2 & -3 \\ 1 & -7 & 0 \end{bmatrix}$$

矩阵A的2范数、0范数、无穷范数分别是?

求出满足 $\|A\|_\infty = \|Ax\|_\infty / \|x\|_\infty$ 的一个向量 x .

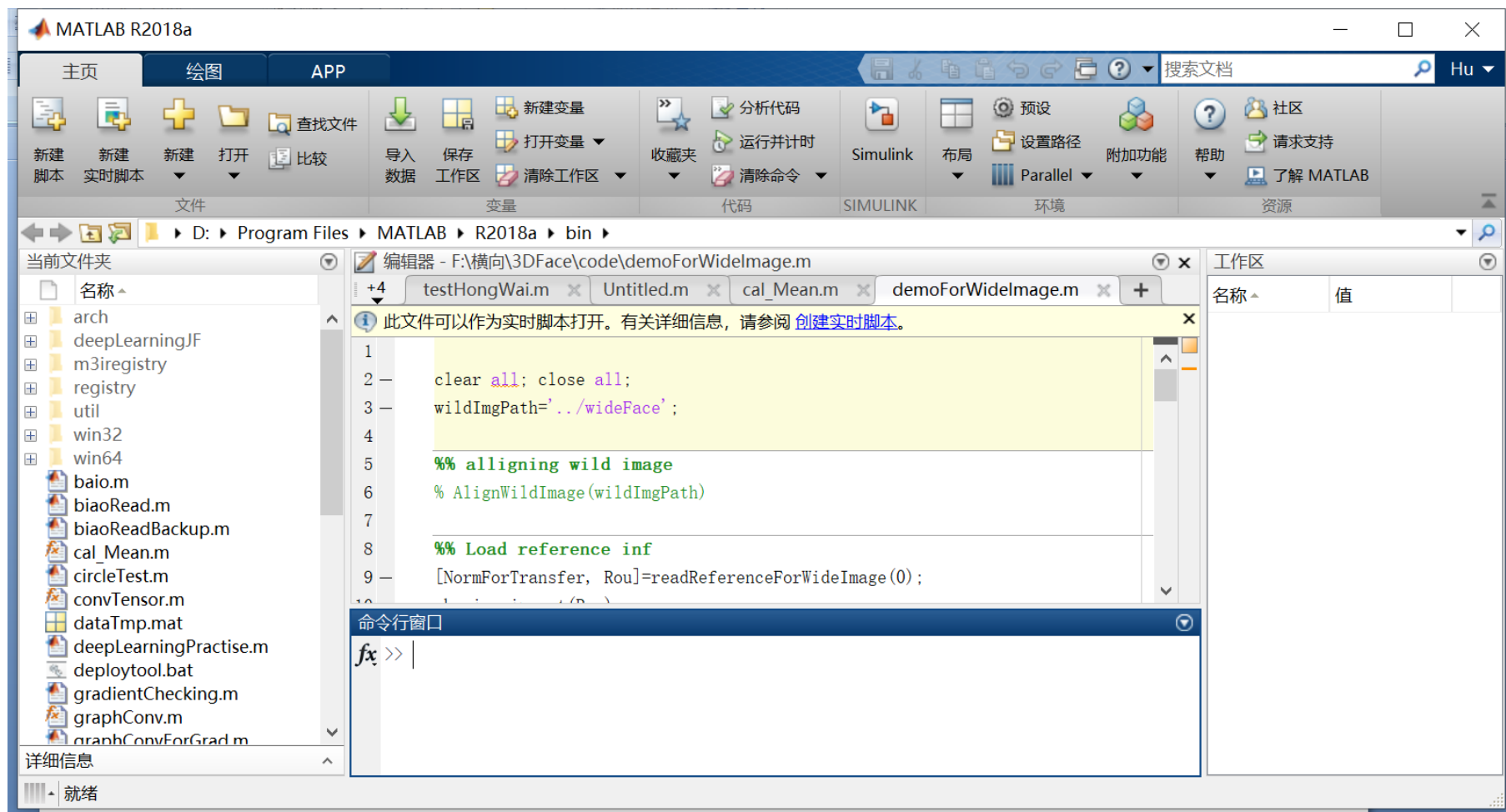
求出满足 $\|A\|_1 = \|Ax\|_1 / \|x\|_1$ 的一个向量 x .

Matlab 基础介绍

Matlab 基础

■ Matlab 基础: 官方教程

https://ww2.mathworks.cn/help/matlab/index.html?s_tid=CRUX_lftnav



Matlab 基础

```
>> ver

-----
MATLAB 版本: 9.4.0.813654 (R2018a)
MATLAB 许可证编号: 968398
操作系统: Microsoft Windows 10 专业版 Version 10.0 (Build 19041)
Java 版本: Java 1.8.0_144-b01 with Oracle Corporation Java HotSpot(TM)
-----

MATLAB                版本 9.4
Simulink               版本 9.1
Aerospace Blockset    版本 3.21
Aerospace Toolbox     版本 2.21
Antenna Toolbox       版本 3.1
Audio System Toolbox  版本 1.4
Automated Driving System Toolbox  版本 1.2
Bioinformatics Toolbox  版本 4.10
Communications System Toolbox  版本 6.6
Computer Vision System Toolbox  版本 8.1
```

```
IEC Certification Kit    版本 3.11
Image Acquisition Toolbox  版本 5.4
Image Processing Toolbox  版本 10.2
Instrument Control Toolbox  版本 3.13
LTE HDL Toolbox          版本 1.1
LTE System Toolbox       版本 2.6
MATLAB Coder             版本 4.0
MATLAB Compiler          版本 6.6
MATLAB Compiler SDK      版本 6.5
MATLAB Distributed Computing Server  版本 6.12
MATLAB Report Generator  版本 5.4
Mapping Toolbox          版本 4.6
Model Predictive Control Toolbox  版本 6.1
Model-Based Calibration Toolbox  版本 5.4
Neural Network Toolbox   版本 11.1
OPC Toolbox              版本 4.0.5
Optimization Toolbox     版本 8.1
Parallel Computing Toolbox  版本 6.12
```

Control System Toolbox	版本 10.4	(R2018a)
Curve Fitting Toolbox	版本 3.5.7	(R2018a)
DO Qualification Kit	版本 3.5	(R2018a)
DSP System Toolbox	版本 9.6	(R2018a)
Data Acquisition Toolbox	版本 3.13	(R2018a)
Database Toolbox	版本 8.1	(R2018a)
Datafeed Toolbox	版本 5.7	(R2018a)
Econometrics Toolbox	版本 5.0	(R2018a)
Embedded Coder	版本 7.0	(R2018a)
Filter Design HDL Coder	版本 3.1.3	(R2018a)
Financial Instruments Toolbox	版本 2.7	(R2018a)
Financial Toolbox	版本 5.11	(R2018a)
Fixed-Point Designer	版本 6.1	(R2018a)
Fuzzy Logic Toolbox	版本 2.3.1	(R2018a)
GPU Coder	版本 1.1	(R2018a)
Global Optimization Toolbox	版本 3.4.4	(R2018a)
HDL Coder	版本 3.12	(R2018a)
HDL Verifier (R2018a)	版本 5.4	(R2018a)
Model Predictive Control Toolbox	版本 6.1	(R2018a)
Model-Based Calibration Toolbox	版本 5.4	(R2018a)
Neural Network Toolbox	版本 11.1	(R2018a)
OPC Toolbox	版本 4.0.5	(R2018a)
Optimization Toolbox	版本 8.1	(R2018a)
Parallel Computing Toolbox	版本 6.12	(R2018a)
Partial Differential Equation Toolbox	版本 0	(R2018a)
Phased Array System Toolbox	版本 3.6	(R2018a)
Polyspace Bug Finder	版本 2.5	(R2018a)
Polyspace Code Prover	版本 9.9	(R2018a)
Powertrain Blockset	版本 1.3	(R2018a)
Predictive Maintenance Toolbox	版本 1.0	(R2018a)
RF Blockset	版本 7.0	(R2018a)
RF Toolbox	版本 3.4	(R2018a)
Risk Management Toolbox	版本 1.3	(R2018a)
Robotics System Toolbox	版本 2.0	(R2018a)

深度学习,
计算机视觉,
神经网络
最优化,
机器人控制,
并行计算,
控制系统,
金融,
图形用户界面,
.....

Matlab 基础

■ Matlab 基础:

【功能演示-1】

求方程 $2x^5 - 3x^3 + 71x^2 - 9x + 13 = 0$ 的全部根。

p = [2,0,-3,71,-9,13]; %建立多项式系数向量

x = roots(p); 求根

x =

-3.4914

1.6863 + 2.6947i

1.6863 - 2.6947i

0.0594 + 0.4251i

0.0594 - 0.4251i

```
>> norm(p, 1)
```

```
ans =
```

```
98
```

```
>> norm(p, 2)
```

```
ans =
```

```
72.8286
```

```
>> norm(p, 'fro')
```

```
ans =
```

```
72.8286
```

求范数

Matlab 基础

■ Matlab 基础:



【功能演示-2】 求解线性方程组

$$\begin{cases} 2x + 3y - z = 2 \\ 8x + 2y + 3z = 4 \\ 45x + 3y + 9z = 23 \end{cases}$$

`a = [2,3,-1;8,2,3;45,3,9]; % 建立系数矩阵a`

`b = [2;4;23]; % 建立列向量b`

`x = inv(a)*b`

`x =`

`0.5531`

`0.2051`

`-0.2784`

Matlab 基础

■ Matlab 基础:



符号计算

syms x y z %建立符号变量

[x,y,z]=solve(2*x+3*y-z-2,8*x+2*y+3*z-4,45*x+3*y+9*z-23)

x =

151/273

y =

8/39

z =

-76/273

Matlab 基础

■ Matlab 基础：函数调用

【功能演示-4】多项式曲线拟合

考虑如下 x-y 一组实验数据：

$x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

$y=[1.2, 3, 4, 4, 5, 4.7, 5, 5.2, 6, 7.2]$

注： $y(x) = x^3 - 2x^2 - 5$ In MATLAB $y=[1 \quad -2 \quad 0 \quad -5]$

■ 一次多项式拟合：

$p1 = \text{polyfit}(x,y,1)$ →

■ 三次多项式拟合：

$p3 = \text{polyfit}(x,y,3)$ →

■ plot 原始数据、一次拟合曲线和三次拟合曲线

$x2=1:0.1:10;$

$y1=\text{polyval}(p1,x2)$

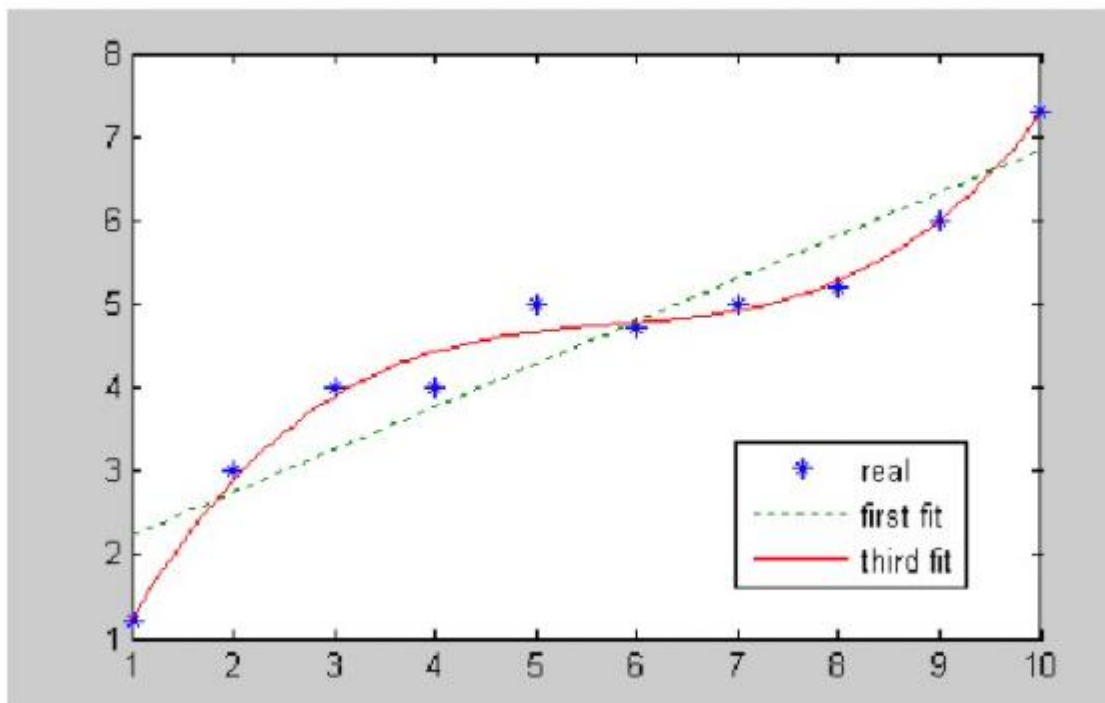
$y3=\text{polyval}(p3,x2)$

$\text{plot}(x, y, '*', x2, y1, ':', x2, y3)$

Matlab 基础

■ Matlab 基础:

拟合曲线图



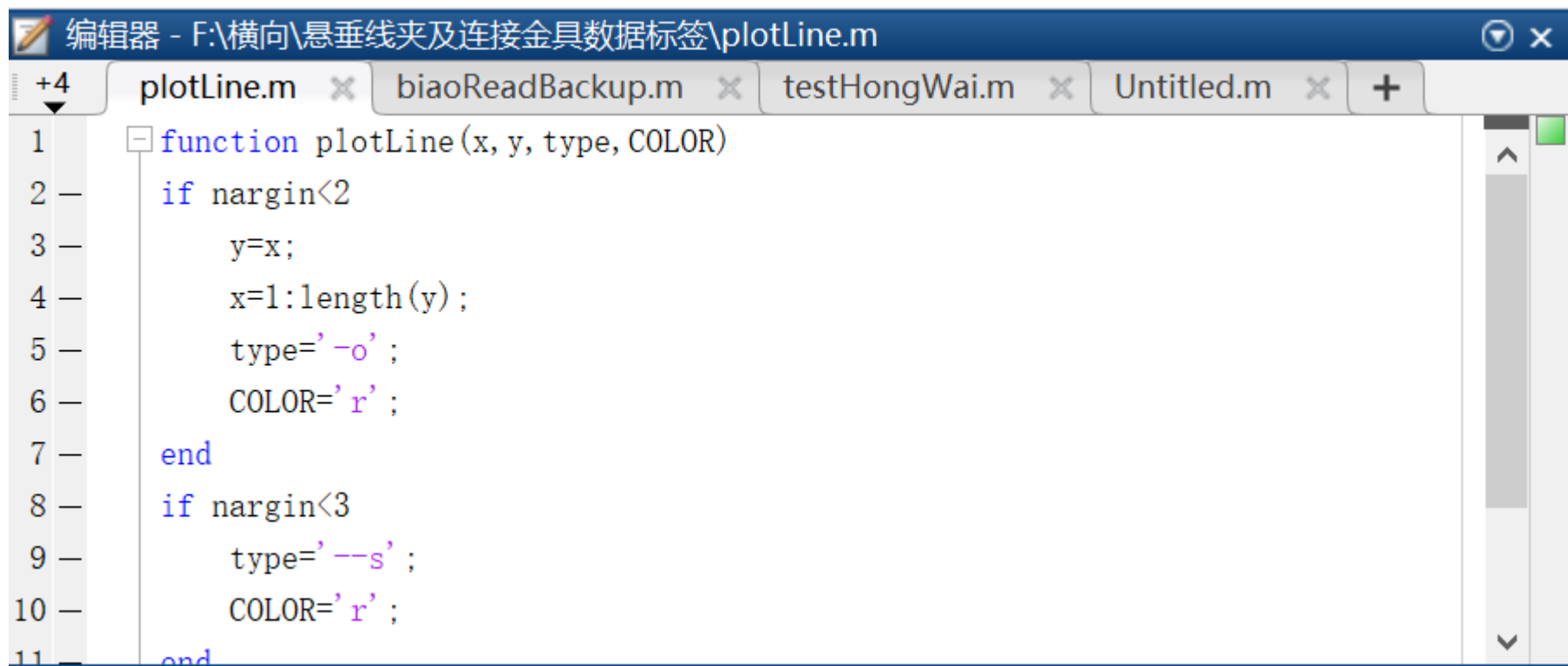
由图可见，三次拟合结果较好。

Matlab 基础

■ Matlab 基础:

函数定义 `function output=functionName(inputs)`

调用简单:



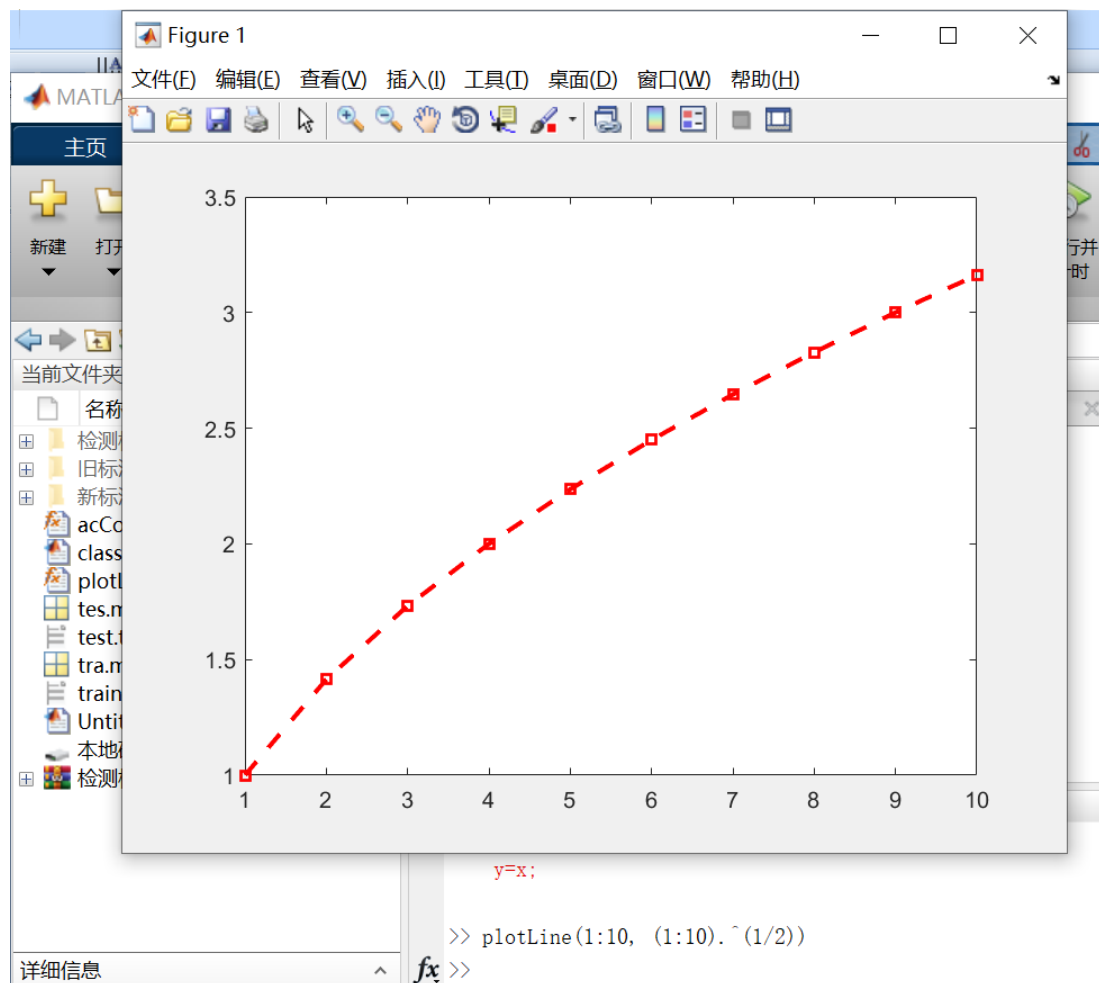
```
编辑器 - F:\横向\悬垂线夹及连接金具数据标签\plotLine.m
+4
plotLine.m x biaoReadBackup.m x testHongWai.m x Untitled.m x +
1 function plotLine(x, y, type, COLOR)
2     if nargin<2
3         y=x;
4         x=1:length(y);
5         type='-o';
6         COLOR='r';
7     end
8     if nargin<3
9         type='--s';
10        COLOR='r';
11    end
```

Matlab 基础

■ Matlab 基础:

函数定义 `function output=functionName(inputs)`

调用简单:



Matlab 基础

- **Matlab 基础:**
错误提示, 容易debug:
可直接搜索内容

```
>> plotLine
```

输入参数的数目不足。

出错 plotLine (line 3)

```
y=x;
```

目录 关闭

< Documentation Home

< Computer Vision System Toolbox

< Deep Learning, Object Detection and Recognition

< Deep Learning for Computer Vision

< Computer Vision System Toolbox

< Functions

detect

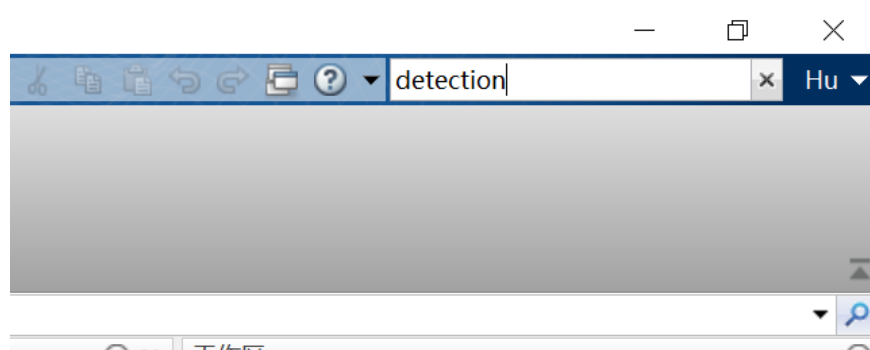
ON THIS PAGE

Description

Input Arguments

Output Arguments

See Also



detect

Class: fastRCNNObjectDetector

Detect objects using Fast R-CNN object detector

Syntax

```
bboxes = detect(detector,I)
[bboxes,scores] = detect(detector,I)
[ ___,labels] = detect(detector,I)
[ ___] = detect( ___,roi)
[ ___] = detect( ___,Name,Value)
```

Description

bboxes = **detect**(**detector**,**I**) detects objects within image **I** by using a Fast R-CNN (regions with confidence scores) and returns the bounding boxes in **bboxes**.

[bboxes,scores] = **detect**(**detector**,**I**) also returns the detection scores for each bounding box.

[___,labels] = **detect**(**detector**,**I**) also returns a categorical array of labels assigned to the bounding boxes defined during training using the **trainFastRCNNObjectDetector** function.

[___] = detect(___,roi) detects objects within the rectangular search region specified by **roi**.

[___] = detect(___,Name,Value) uses additional options specified by one or more **Name,Value** pairs.

Input Arguments

Matlab 基础

■ Matlab 基础：工具箱介绍，demo

Computer Vision System Toolbox

Design and simulate computer vision and video processing systems

Computer Vision System Toolbox™ provides algorithms, functions, and apps for designing and simulating computer vision and video processing systems. You can perform feature detection, extraction, and matching, as well as object detection and tracking. For 3-D computer vision, the system toolbox supports single, stereo, and fisheye camera calibration; stereo vision; 3-D reconstruction; and 3-D point cloud processing.

Algorithms for deep learning and machine learning enable you to detect faces, pedestrians, and other common objects using pretrained detectors. You can train a custom detector using ground truth labeling with training frameworks such as Faster R-CNN and ACF. You can also classify image categories and perform semantic segmentation.

Algorithms are available as MATLAB® functions, System objects, and Simulink® blocks. For rapid prototyping and embedded system design, the system toolbox supports fixed-point arithmetic and C-code generation.

Exa
Fur
Rel
PDI

Getting Started

Learn the basics of Computer Vision System Toolbox

Feature Detection and Extraction

Image registration, interest point detection, extracting feature descriptors, and point feature matching

Deep Learning, Object Detection and Recognition

Deep learning, object detection, recognition, bag of features, template matching, background estimation, and ground truth labeling

Object Tracking and Motion Estimation

Optical flow, activity recognition, motion estimation, and tracking

Camera Calibration

Estimate camera intrinsics, distortion coefficients, and camera extrinsics

Multiple View Geometry

Extract 3-D information from 2-D images, perform stereo rectification, depth estimation, 3-D reconstruction, triangulation, and structure from motion

3-D Point Cloud Processing

Downsample, denoise, transform, visualize, register, and fit geometrical shapes of 3-D point clouds

Matlab 基础

■ Matlab 基础：工具箱介绍，demo

Tracking Pedestrians from a Moving Car

R20

This example shows how to track pedestrians using a camera mounted in a moving car.

Open Script

Overview

This example shows how to perform automatic detection and tracking of people in a video from a moving camera. It demonstrates the flexibility of a tracking system adapted to a moving camera, which is ideal for automotive safety applications. Unlike the stationary camera example, [The Motion-Based Multiple Object Tracking](#), this example contains several additional algorithmic steps. These steps include people detection, customized non-maximum suppression, and heuristics to identify and eliminate false alarm tracks. For more information please see [Multiple Object Tracking](#).

This example is a function with the main body at the top and helper routines in the form of [nested functions](#) below.

```
function PedestrianTrackingFromMovingCameraExample()  
  
% Create system objects used for reading video, loading prerequisite data file, detecting pedestrians, and displaying the results.  
videoFile      = 'vippedtracking.mp4';  
scaleDataFile  = 'pedScaleTable.mat'; % An auxiliary file that helps to determine the size of a pedestrian at different pixel locations.  
  
obj = setupSystemObjects(videoFile, scaleDataFile);  
  
detector = peopleDetectorACF('caltech');  
  
% Create an empty array of tracks.  
tracks = initializeTracks();  
  
% ID of the next track.  
nextId = 1;  
  
% Set the global parameters.  
option.ROI      = [40 95 400 140]; % A rectangle [x, y, w, h] that limits the processing area to ground locations.  
option.scThresh = 0.3;             % A threshold to control the tolerance of error in estimating the scale of a detected pedestrian.  
option.gatingThresh = 0.9;         % A threshold to reject a candidate match between a detection and a track.  
option.gatingCost = 100;           % A large value for the assignment cost matrix that enforces the rejection of a candidate match.  
option.costOfNonAssignment = 10;    % A tuning parameter to control the likelihood of creation of a new track.  
option.timeWindowSize = 16;        % A tuning parameter to specify the number of frames required to stabilize the confidence score of a track.  
option.confidenceThresh = 2;       % A threshold to determine if a track is true positive or false alarm.  
option.ageThresh = 8;              % A threshold to determine the minimum length required for a track being true positive.  
option.visThresh = 0.6;            % A threshold to determine the minimum visibility value for a track being true positive.
```

激活 Windows

Matlab 基础

■ Matlab 基础：工具箱介绍，demo

Tracking Pedestrians from a Moving Car

R2019a

This example shows how to track pedestrians using a camera mounted in a moving car.

Overview

This example shows how to track pedestrians using a camera mounted in a moving car. It is ideal for automated systems that need to detect and track people in a video stream.

This example is a

`function` Pedes

```
% Create system objects
videoFile = 'pedestrians.avi';
scaleDataFile = 'pedestrians.mat';

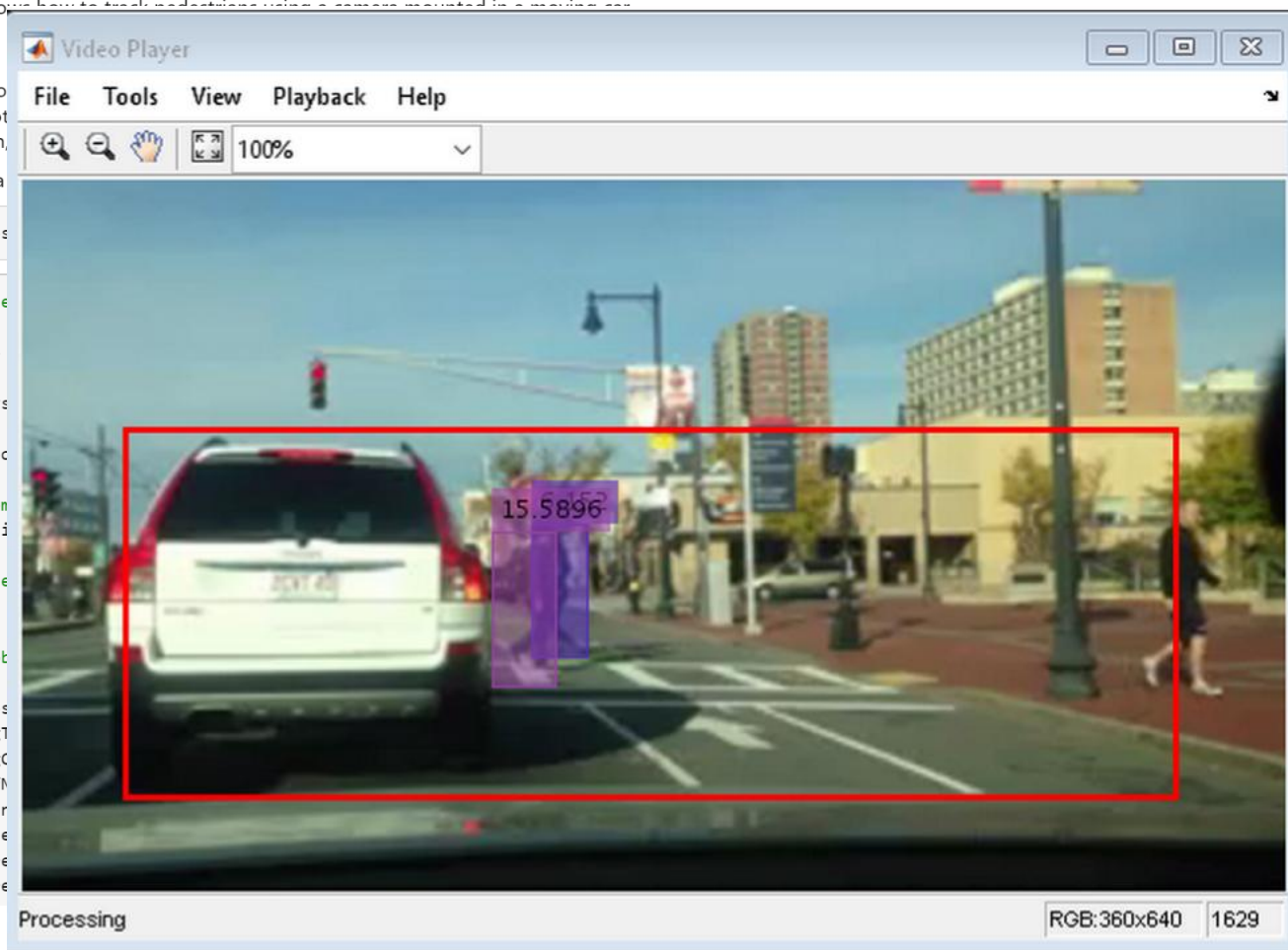
obj = setupSystem('pedestrians', videoFile, scaleDataFile);

detector = obj.Detector;

% Create an empty track
tracks = initTrack(detector);

% ID of the next track
nextId = 1;

% Set the global options
option.ROI = [0 0 1000 1000];
option.scThreshold = 0.5;
option.gatingThreshold = 0.5;
option.gatingCost = 0.5;
option.costOfMiss = 1;
option.timeWindow = 10;
option.confidenceThreshold = 0.5;
option.ageThreshold = 10;
option.visibilityThreshold = 0.5;
```



Open Script

m adapted to a moving camera, which includes several algorithmic steps. These steps include object detection and object tracking.

tion.

ch.

ure of a track.

激活 Windows

Matlab 基础

■ Matlab 基础:

Matlab 工具很强大，涵盖各个方向行业

但是不能自动求导数；

人工智能从业者大多使用 `pytorch`, `tenserflow`

THE END