13. 1stOpt 数据读取方法及技巧

1stOpt 进行问题求解及具体代码编写时经常会涉及到数据的读取交换,有多种方式可以轻易实现,下面给予简单的介绍和示范。

13.1 直接在代码本写数据

代码本中除了编写求解代码外,所涉及的数据也可以直接写在代码本里,这 也是最直接和最简单的方式。以下面数据拟合为例。

主 1 2 1	か ム 粉 垢
衣 13.1.	1以百蚁犯

Х	-0.08,-0.065,-0.05,-0.03,-0.015,0,0.015,0.03,0.05,0.065,0.08
У	20.260,19.726,19.502,18.727,18.588,18.123,18.592,18.884,19.545,19.887,20.991
拟	合公式:

$$y = b_1 \cdot (x - b_2)^4 + b_3 \cdot (x - b_4)^2 + b_5$$
(13-1)

下面求解代码一中数据为横向形式、代码二为纵向形式,横向形式中每行数据结尾需要以分号";"结束。不论是纵向还是横向,数据间可以用逗号",",空格""或 Tab 制表符空格键隔开,最好统一,不要混用。横向数据模式中数据前的"x="和"y="可以要也可以不要。

代码 13-1:

Variable x,y;	
Function y=b1*(x-b2)^4+b3*(x-b4)^2+b5;	
Data;	
x = -0.08, -0.065, -0.05, -0.03, -0.015, 0, 0.015, 0.03, 0.05, 0.065, 0.08;	
y=20.260,19.726,19.502,18.727,18.588,18.123,18.592,18.884,19.545,19.887,20.991;	

代码 13-2:

Variable x,y;
Function y=b1*(x-b2)^4+b3*(x-b4)^2+b5;
Data;
x y
-0.08 20.260
-0.065 19.726
-0.05 19.502
-0.03 18.727
-0.015 18.588
0 18.123
0.01518.592
0.03 18.884
0.05 19.545

0.06519.887

0.08 20.991

将上面拟合问题转换成函数优化问题,即残差平方和最小,通过"Constant" 关键字定义常数并读取数据:

Min.
$$\sum \left(\left(\left(b_1 \cdot (x - b_2)^4 + b_3 \cdot (x - b_4)^2 + b_5 \right) - y \right)^2 \right)$$
 (13-2)

代码 13-3:

ConstStr f=b1*(x-b2)^4+b3*(x-b4)^2+b5;

Constant

x=[-0.08,-0.065,-0.05,-0.03,-0.015,0,0.015,0.03,0.05,0.065,0.08],

y=[20.260,19.726,19.502,18.727,18.588,18.123,18.592,18.884,19.545,19.887,20.991];

Plot x[x],y,f;

MinFunction Sum(x,y)(sqr(f-y));

运行上面三段代码都可以得到相同的结果。



图 13-1. 拟合结果图

拟合结果:

- 均方差(RMSE): 0.141234695309986
- 残差平方和(SSR): 0.21941963075235
- 相关系数(R): 0.984810090029115
- 相关系数之平方(R^2): 0.969850913423154
- 修正 R 平方(Adj. R^2): 0.962313641778943
- 确定系数(DC): 0.969850913423154
- 卡方系数(Chi-Square): 0.00574474160401383
- F 统计(F-Statistic): 48.2527510884866

参数 最佳估算 -----

-15064.5797919747 b1

b2	0.0130731392885844
b3	451.843129782349
b4	0.00123599198518782
b5	18.3973757899551

代码本里直接写数据虽然直观但也有缺点:当数据量很大时不易查看和编辑,同时一个代码本里有多段代码时,也不方便阅读,此时以文件形式读取数据即是一种很好的方式。

13.2 外部 Excel 文件(.xls, .xlsx) 读取数据

Excel 数据文件使用非常广泛,几乎已成为当今数据的"标本"。1stOpt 可以直接读取和保存 Excel 数据。以上节案例数据为例,如图 13-2 保存为"ExcelTest.xlsx"文件。



图 13-2. Excel 拟合数据文件

代码 13-4:

Variable x,y;	
Function y=b1*(x-b2)^4+b3*(x-b4)^2+b5;	
DataFile "ExcelTest.xlsx[Sheet1[A2:B12]]";	

使用"DataFile"关键字调用 Excel 文件。如果 Excel 文件路径与 1stOpt 运行的.mmf 文件路径相同,则代码调用中可省去路径,仅需文件名即可;反之,如果.xlsx 文件与.mmf 文件路径不同,则需将全路径加上,如"DataFile"c:\Mytest\ExcelTest.xlsx[Sheet1[A2:B12]]";"。另外调用的 Excel 文件如果正被 Excel 打开,运行 1stOpt 代码前需从 Excel 关闭要调用的 Excel 文件。

对应上节公式 13.2 及代码三,从 Excel 文件中读取常数,以最小值优化问题 求解拟合问题的 Excel 数据如图 13-2,求解代码如下代码五。

×I		5	» -			Exce	ITest - Excel		6		? 🛧		×
Ż	て件	开始	插入 〕	页面布局	公式 数	数据 审阅	视图					-	7
A1		•	×	fx	X								~
	A	В	С	D	Е	F	G	Н	Ι	J	K	L	
1	X	-0.08	-0.065	-0.05	-0.03	-0.015	0	0.015	0.03	0.05	0.065	0.08	
2	у	20.26	19.726	19.502	18.727	18.588	18.123	18. 592	18.884	19.545	19.887	20.991	
3													
4													
5													
6													-
7		Sheet1	Sheet2	(\pm)				1				•	
就绪	í				平均值: 9.6	73863636	计数: 24 求	和: 212.825			l	+ 100%	6

图 13-3. Excel 常数数据文件

代码 13-5:

ConstStr f=b1*(x-b2)^4+b3*(x-b4)^2+b5; Constant x="ExcelTest.xlsx[Sheet2[B1:L1]]", y="ExcelTest.xlsx[Sheet2[B2:L2]]";

Plot x[x],y,f;

MinFunction Sum(x,y)(sqr(f-y));

运行代码四和代码五均可获得与上节相同的结果。

13.3 外部文本文件(.txt,.csv,.ssf)读取数据

".txt"是文本文件,同一行数据间可用逗号","或者空格号或者 Tab 制表符空格号隔开;".csv"与".txt"文本文件基本相同,只是数据是用逗号分隔;".ssf"是 1stOpt 自带电子表格的数据文件。

🔳 test - ji	己事本		_	×
文件(F) 编辑	量(E)格式(O) 3	查看(V) 帮助)(H)	
-0.08	20.260			<u>^</u>
-0.065	19.726			
-0.05	19.502			
-0.03	18.727			
-0.015	18.588			
0.015	18.592			
0.03	18.884			
0.05	19.545			
0.065	19.887			
0.08	20.991			
0	18.123			
1				~

图 13-4. ".txt "数据文件

Test - 记事本	-	×
文件(F) 编辑(E) 格式(O) 查看(V)	帮助(H)	
-0.08,20.26		^
-0.065,19.726		
-0.05,19.502		
-0.03,18.727		
-0.015,18.588		
0,18.123		
0.015,18.592		
0.03,18.884		
0.05,19.545		
0.065,19.887		
0.08,20.991		

图 13-5. ".csv "数据文件

💠 1stOpt电子表格 - [0	C:\My	App\1stOpt	9.0\Test.ssf]		-		×
文件视图 编辑格	式	工具 统计分标	Я				
🗅 🚅 🕶 🖬 🐰 🖻	•	🛱 🚾 <u>A</u>	🛓 + 🕭 + 🖻 ·	-	-		
≜↓ X↓ +⊩ 🕒 - 3+	• 'f'	∌ ₩ X					
🗅 🖪 🕆 ት ት		A	В	С	D	E	
Base Folder	1	x	У				~
Sheet1	2	-0.08	20.260				
Sheet2	3	-0.065	19.726				
Sheet3	4	-0.05	19.502				
	5	-0.03	18.727				
	6	-0.015	18.588				
	7	0.015	18.592				
	8	0.03	18.884				
	9	0.05	19.545				
	10	0.065	19.887				
	11	0.08	20.991				
	12	0	18.123				~
	2						-

图 13-6. ".ssf "数据文件

".txt"与".csv"数据文件调用方式完全一样,文件名不同而已

代码 13-6

Variable x,y; Function y=b1*(x-b2)^4+b3*(x-b4)^2+b5; DataFile "Test.txt";

".ssf "数据文件调用与 Excel 表格调用方式一致

代码 13-7

Variable x,y;

Function $y=b1*(x-b2)^{4}+b3*(x-b4)^{2}+b5;$

DataFile "Test.ssf[Sheet1[A2:B12]]";

路径问题与调用 Excel 文件相同。

13.4 外部 Matlab 数据文件(.mat)读取数据

".mat"是 Matlab 专用数据文件格式,如下图。目前 1stOpt 还不能直接在 代码本里调用".mat"文件,但其内置的电子表格可以读入".mat"文件,因此 可以通过先将".mat"文件读入电子表格,再在代码本里直接调用,或转存为".xlsx" 或".csv"或".ssf"文件格式后再调用.

	绘图	变量	视图					H % H	196	0 ? 0
		▲打开▼ 行		列	æ 3					
根据	新选内容	打印 🕶 1		1	插入册	除。非非多,				
	新建▼			le Viceos		• ***				
-	变量		所选	内容		编辑				
	Sheet1 ×									
1	1x2 double									
	1	2	3	4	5	6	7	8	9	10
1	-0.0800	20.2600								
2	-0.0650	19.7260								
3	-0.0500	19.5020								
4	-0.0300	18.7270								
5	-0.0150	18.5880								
6	0	18.1230								
7	0.0150	18.5920								
8	0.0300	18.8840								
9	0.0500	19.5450								
10	0.0650	19.8870								
11	0.0800	20.9910								
			111	ш		110				•

图 3-7. Matlab 数据文件

File View Edit F	ormat	Tool Stat	tistic Analysis 🛕 - 🕭 - 🐧	• 🖩 🖽	■ 🗖 🖽 🕶	1	
≜↓ Z↓ ⊕ 🕒 - 3		<u></u> שיע ו א	(
🗅 🔳 住 介 사		A	В	С	D	E	F
Base Folder	1	-0.08	20.26				
	2	-0.065	19.726				1
	3	-0.05	19.502				
	4	-0.03	18.727				
	5	-0.015	18.588				
	6	0	18.123	-			
	7	0.015	18.592				
	8	0.03	18.884		-		-
	9	0.05	19.545				1
	10	0.065	19.887				
	11	0.08	20.991				
	12	< III	1				· · ·

图 13-8. 1stOpt 内置电子表格读入".mat"文件

13.5 内部电子表格读取数据

1stOpt 内置电子表格有两种形式,一种是相对独立的电子表格如图 13-9,可 以编辑及读入输出不同格式的数据文件,一般用"Sheet1","Sheet2"...表示各个 表单;另一种是附属于代码本电子表格,如图 13-10,一般用"CodeSheet1", "CodeSheet2"...表示各个表单,该表格数据可与代码本代码共同保存于同一个 ".mff"文件里。



图 13-9.1stOpt 内置独立电子表格

		- Invohinfflx	省法沿署	结果					
	代码编辑本	代码本称据表	异么成血 88 附任						
				The Arial	- 8 -	1			
文件 代码块 关键词		00 - E 6 E 6- 6 6		1 010					
已开文件 ×	🗋 🗟 🕶 🛅 🗄	3 🎟 🖬 🗐 🛛		₽ ≠ • 3e	뿜 글→ 밭				
🗋 untitledff.mff	A	В	С	D	E	F	G	Н	I
] untitled2.mff	1 x	у							1
🛑 prob.mff *	2 -0.08	20.26							
	3 -0.065	19.726							
	4 -0.05	19.502							
	5 -0.03	18.727							
	6 -0.015	18.588		1					
	7 0	18.123		((
) 🔁 🎓 🖡 🖾	8 0.015	18.592		1	[
代码块	9 0.03	18.884							
1 - CodeBlock [5]	10 0.05	19.545							
2 - CodeBlock [16]	11 0.065	19.887			s				
3 - CodeBlock [7]	12 0.08	20.991							
4 - CodeBlock [4] -	13	- 18	- 18						8
) 🕇 🕯 🖡 🗙	14 CodeSheet1	CodeSheet2	CodeSheet3	2					,

图 13-10.1stOpt 内置代码本附属电子表格

代码 13-8 (从独立电子表格中读取数据):

Variable x,y;

Function y=b1*(x-b2)^4+b3*(x-b4)^2+b5;

DataFile "Sheet1[A2:B12]";

注意,使用该方式时,独立电子表格窗口必须启动。

代码 13-9 (从代码本附属电子表格中读取数据):

13.6 内部存储文件中读取数据

代码本除了有附属电子表格外,还有附属笔记本和附属附件容器,如图 11。 缺省状态下附属笔记本和附属附件容器页面是不可见的,可点击鼠标右键确定是 否可视。

在附件容器页面,点击添加文件,将之前用到的3个文件"Test.txt","Test.ssf" 和"ExcelTest.xlsx"添加进来,保存该".mff"文件后,所有附件容器内的文件都 会保存在同一个".mff"文件中.

1stOpt - [F:\MyApp\1stOpt	目关\公众号\prob.mff]	
文件编辑程序 工具 工具	精代码本帮助 回回 作码本 3 - [nuch still) 省法识票 维里	
2014 10月秋 天雄问 已开文件 ×	文件名	大小
untitledff.mff Untitled2.mff	test.txt Test.ssf	1KB 28KB
prob.mit *		
154900 2 - CodeBlock [16] 3 - CodeBlock [7] 4 - CodeBlock [4] 5 - CodeBlock [6]		
Avillierselasser0		

图 13-11. 1stOpt 内置附件容器

代码本里调用格式与前述代码四、五、六和七完全一样。需要注意的是优先顺序问题:比如代码本里调用文件如"ExcelTest.xlsx",如果没有给出全路径而只给出了文件名,则会首先搜索与".mff"文件相同的文件夹,该文件夹中如果 "ExcelTest.xlsx"不存在,就会到附属容器里寻找"ExcelTest.xlsx"文件;当然如 果代码中调用的文件给出了全路径,就不存在前述优先顺序问题了。

13.7 ".txt"文本文件的特殊读取

前面已给过文本文件的读取方式,下面以 0-1 背包问题(0-1 Knapsack Problem) 为例,再给出一种特殊的读取格式:指定行读取。

0-1 背包:假设有一个背包,其容量为 C,一堆物品,其数量为 N,每个物品的质量及价值分别为 W 和 P,试问如何装包使得所装物品价值最大?

构建模型: 假设 Xi 代表第 i 个物品的取舍, 1 表示选取, 0 表示不选取,则 有如下模型:

Max.
$$\sum_{i=1}^{N} (X_i \cdot P_i)$$
(13-3)

St.
$$\sum_{i=1}^{N} (X_i \cdot W_i) \le C$$

已知 C=2557, N=10, W=[986 589 848 314 255 905 435 607 979 749], P=[675 233 95 576 495 625 509 544 904 548], 一种求解代码如下:

代码 13-10

Algorithm = LP;		
Constant C=2557, N=10;		
Constant W=[986 589 848 314 255 905 435 607 979 749],		
P=[675 233 95 576 495 625 509 544 904 548];		
BinParameter X(N);		
MaxFunction Sum(i=1:N,P,X)(X*P);		
Sum(i=1:N,W,X)(X*W)<=C;		

上面是 N=10 情况下的 0-1 背包问题求解代码,如果有 N=10,20,50,100...1000 多种情况,此时该如何保存数据并能快速方便求解呢?每一种情况下编写一段代 码,涉及到众多数据,显然会很费时间。可以将数据按如下格式保存成文本文件, 即第一行为 N 数据,第二、第三行分别为质量 W 和价值 P 数据,第四行为背包 容量数据,保存为文本文件如 "D:\Mytest\kp10.txt"。

表 2. 数据保存格式

10	
986 589 848 314 255 905 435 607 979 749	
675 233 95 576 495 625 509 544 904 548	
2557	
代码 13-11:	
Algorithm = LP;	
Constant N = " D:\Mytest\kp10.txt[1]";	
Constant W(N) = " D:\Mytest\kp10.txt[2]";	
Constant $P(N) = "D:Mytest\kp10.txt[3]";$	
Constant C = " D:\Mytest\kp10.txt[4]";	

BinParameter X(N);

MaxFunction Sum(i=1:N,P,X)(P*X);

Sum(i=1:N,W,X)(W*X) <= C;

上述代码中注意读取文本文件时与前面略有不同,是按指定行读取数据。比如"D:\Mytest\kp10.txt[1]",后面中括号中的数值表示行数,也即读第几行,其它类似。这样对于不同 N 的情况,只要按表 2 格式保存数据,在代码十一中仅需改变文件名即可,非常方便。

13.8 小结

1stOpt 可以非常方便和多种常用数据文件进行"交流",掌握对不同数据文件的读取调用方式,对编写求解代码无疑具有事半功倍的作用。