

1. 模型的规格选择和函数约束的充分性测试

2.4. 规格选择和充分性测试

除了关于误差项属性的通常考虑之外，关于MIDAS回归模型的规范还有两个主要问题。

- 首先，需要选择合适的函数约束，因为它们的选择会影响模型的精度。
 - 其次，需要选择合适的最大滞后阶数。
- 同时解决这两个问题的一种方法是使用一些信息准则，根据参数限制和使用样本内或样本外精度测量的滞后阶选择最佳模型。包的midas_r_ic_table和amidas_table函数允许用户使用一些常用的信息标准(如AIC和BIC)和用户指定的函数约束列表进行样本内选择。

另一种方法是测试所选函数约束的充分性。例如，当模型(3)中存在自回归项($p > 0$)时，Ghysels等(2006b)指出，在一般情况下， $\varphi(L) = \beta(L)/a(B)$ 具有季节性规律，因此对应于纯分布滞后模型(即不存在自回归项)中解释变量对因变量的某些季节性影响。使用包的hAh_test和hAhr_test函数实现了所选函数约束的充分性检验。

实例

GDP增长预测

复制了Ghysels(2013)中提供的例子。使用MIDAS回归来预测季度GDP增长和月度非农就业增长。预测方程如下

$$y_{t+1} = \alpha + \rho y_t + \sum_{j=0}^8 \theta_j x_{3t-j} + \varepsilon_t$$

其中 y_t 为经季节调整的美国季度实际GDP的对数差值， x_{3t} 为月度非农就业总人数的对数差值。数据来自圣路易斯FRED网站。

什么情况要取对数：我们的模型中是否要对季度数据取对数需要实验

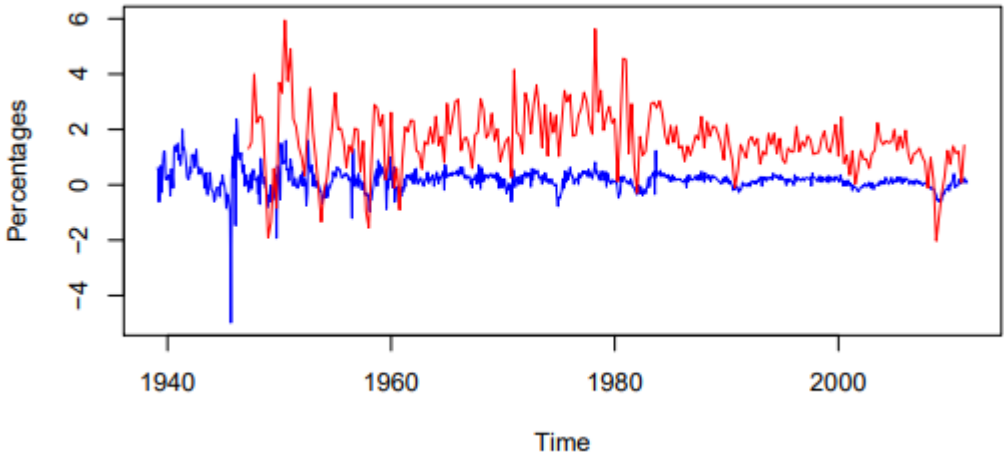
- 第一，如果理论模型中的变量为对数形式，则应取对数。
- 第二，如果变量有指数增长趋势（exponential growth），比如 GDP，则一般取对数，使得 lnGDP 变为线性增长趋势linear growth）。
- 第三，如果取对数可改进回归模型的拟合优度（比如R2或显著性），可考虑取对数。
- 第四，如果希望将回归系数解释为弹性或半弹性（即百分比变化），可将变量取对数。
- 第五，如果无法确定是否该取对数，可对两种情形都进行估计，作为稳健性检验（robustness check）。若二者的回归结果类似，则说明结果是稳健的。

首先，我们加载数据并执行必要的转换。

```
data("USqgdp", package = "midasr")
data("USpayems", package = "midasr")
y <- window(USqgdp, end = c(2011, 2))
x <- window(USpayems, end = c(2011, 7))
yg <- diff(log(y)) * 100 R> xg <- diff(log(x)) * 100
nx <- ts(c(NA, xg, NA, NA), start = start(x), frequency = 12)
ny <- ts(c(rep(NA, 33), yg, NA), start = start(x), frequency = 4)
```

数据需要是 Time Series(ts将数据转换成时间序列类型)类型

最后两行是为了平衡原始数据中不同的样本量。我们只需在数据的开头和结尾添加额外的NA值。数据的图形表示如图3所示。



为了指定midas_r函数的模型，我们将其改写为以下等价形式:

$$y_t = \alpha + \rho y_{t-1} + \sum_{j=3}^{11} \theta_j x_{3t-j} + \varepsilon_t$$

正如Ghysels(2013)，将估计样本从1985年第一季度限制到2009年第一季度。
用Beta多项式、非零Beta和U-MIDAS权重规范对模型进行评估。

window () 函数取出时间序列的一段， 如果指定 frequency=TRUE 还可以仅取出某个月（季度）
start 是最优化的初始值，必须是带有命名元素的列表
nbeta nebtaMT 分别对应Beta多项式的权重规范和非零beta的权重规范
U-MIDAS 对最优化初始值设为空， start = null

```
xx <- window(nx, start = c(1985, 1), end = c(2009, 3))
yy <- window(ny, start = c(1985, 1), end = c(2009, 1))
beta0 <- midas_r(yy ~ mls(yy, 1, 1) + mls(xx, 3:11, 3, nbeta), start = list(xx = c(1.7, 1, 5)))

coef(beta0)
```

(Intercept)	yy	xx1	xx2
0.8311331	0.1056152	2.5924606	1.0149478
	xx3		
12.4761333			

```
betan <- midas_r(yy ~ mls(yy, 1, 1) + mls(xx, 3:11, 3, nbetaMT), start = list(xx = c(2, 1, 5, 0)))

coef(betan)
```

(Intercept)	yy	xx1	xx2	xx3	xx4
0.93868569	0.06618413	2.27682100	0.98674913	1.50874150	-0.09164647

```
um <- midas_r(yy ~ mls(yy, 1, 1) + mls(xx, 3:11, 3), start = NULL)

coef(um)
```

(Intercept)	yy	xx1	xx2	xx3
0.92989757	0.08358393	2.00047205	0.88134597	0.42964662
xx4	xx5	xx6	xx7	xx8
-0.17596814	0.28351010	1.16285271	-0.53081967	-0.73391876
xx9				
-1.18732001				

在2009Q2 - 2011Q2的9个季度的样本外数据上评价这三种模型的预测性能:

```

fulldata <- list(xx = window(nx, start = c(1985, 1), end = c(2011, 6)), yy =
window(ny, start = c(1985, 1), end = c(2011, 2)))

insample <- 1:length(yy)
outsample <- (1:length(fulldata$yy))[-insample]

avgf <- average_forecast(list(beta0, betan, um), data = fulldata, insample =
insample, outsample = outsample)

sqrt(avgf$accuracy$individual$MSE.out.of.sample)

```

```
[1] 0.5361953 0.4766972 0.4457144
```

avg函数对 平均MIDAS模型使用指定的加权方案进行预测, 算出样本内和样本外的准确性测量。

我们看到，不受限制的MIDAS回归模型给出了最佳的样本外均方误差

```
accuracy$individual$MSE.out.of.sample
```

averg_forecast函数很重要，可以进行各模型预测性能的比较

```
fweights = c("EW", "BICW", "MSFE", "DMSFE")
```

代码

进行简单的数据读取，开了个头：

```
import pandas as pd  
import datetime
```

读取 股票 证券代码|交易时间|涨跌幅

```
data_stock = pd.read_excel(r"C:/Users/jc/Documents/大学/0大三其他/"+  
                           "_NEW金融数据挖掘课题/PyProgram/_eanring announcement"+  
                           "/日个股回报率文件225335089/TRD_Dalyr.xlsx",  
                           header=0,skiprows=[0,2],usecols="A,B,D",  
                           dtype={'证券代码':str,'交易日期':datetime})
```

读取 财报 证券代码|报告类型|报告公布日期|调整前每股收益|调整后每股收益

```
data_ea = pd.read_excel(r"C:/Users/jc/Documents/大学/0大三其他"+  
                        "_/NEW金融数据挖掘课题/PyProgram/_eanring announcement/" +  
                        "年、中、季报基本情况文件011833799/IAR Rept.xlsx",
```

```
header=0,skiprows=[0,2],usecols="A,C,D,E,F,G",
dtype={'证券代码':str,'统计截止日期':datetime,
      '报告公布日期':datetime})
```

创建表格 存有数据 证券代码|时间（交易时间）|股票-涨跌幅|财报-调前收益|财报-调后收益|可见性(可见时调整为对应季度财报数据)

```
dict_data = { '证券代码': list(data_stock['证券代码']),
              '日期': list(data_stock['交易日期']),
              '股票-收益率':list(data_stock['涨跌幅'])}
data = pd.DataFrame(dict_data)
```