Mínimos cuadrados totales (TLS): ejemplo multivariable 5 señales

© 2022, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Este código funcionó sin errores en Matlab **R2022a**

Presentaciones en vídeo:

http://personales.upv.es/asala/YT/V/tls51.html , http://personales.upv.es/asala/YT/V/tls52.html .

Objetivo: ilustrar un caso de mínimos cuadrados totales multivariable (relacionando 5 variables, aunque luego despejaremos dos de ellas).

Tabla de Contenidos

Generación de los datos (we are God)....1
Estimado de modelo por mínimos cuadrados ordinarios LS (sesgado)....2
Estimado modelo mínimos cuadrados totales TLS....2
Con los datos organizados con 1 muestra 1 FILA....2
Con los datos organizados con 1 muestra 1 COLUMNA....4
Addenda: ¿y si no hubiera escalado a ruido idéntico en todos los canales?... pues ¡no funciona!....6
Conclusiones....7

Generación de los datos (we are God)

Vamos a plantear un modelo donde 2 variables (y_1, y_2) sean función de otras tres (x_1, x_2, x_3). Bueno, realmente TLS buscaria encontrar "relaciones" en el vector $(y_1, y_2, x_1, x_2, x_3)$.

En concreto, el modelo sería

$$(y_1 \;\; y_2)_{N\times 2} = (x_1 \;\; x_2 \;\; x_3)_{N\times 3} \cdot \Theta_{3\times 2}$$

donde los datos están en "fila".

Obviamente, si estuvieran en columna, el mismo modelo se expresaría transponiendo todo.

```
%Genero datos
Xlimpia=randn(90000,3)*diag([25 1.7 18]);%datos X al azar
ThetaLimpio=[2 40 5;-2 10 3]'
```

```
ThetaLimpio = 3×2
     2    -2
    40    10
     5     3
```

```
Ylimpia=Xlimpia*ThetaLimpio;
%Los datos estarán corrompidos por ruido de cierta desv. típica
dtx1=6; dtx2=0.4; dtx3=5; dty1=0.5; dty2=7.5;
DesTX=diag([dtx1 dtx2 dtx3]); DesTY=diag([dty1 dty2]);
X=Xlimpia+randn(size(Xlimpia))*DesTX; %añado ruido de medida X
Y=Ylimpia+randn(size(Ylimpia))*DesTY;%añado ruido de medida Y
```

```
%lo de arriba no estaría en el código de análisis de datos, es simplemente
%para comprobar que lo de abajo da correcto (sesgado vs. no sesgado).
%Realmente lo de arriba sería "hacer el experimento" y obtener los datos "X Y"...
```

TLS no sabe ni de entradas ni salidas, simplemente de relación entre variables... el qué se despeja en función de qué ya depende del usuario final cómo interprete esas relaciones.

*en la "vida real" los datos no serán distribución normal + ruido de medida de distribución normal de desviación típica conocida, ni tendré casi cien mill de ellos... esto está preparado para que concuerde con lo que la teoría espera... manejar "pocos" datos reales, que tampoco esté claro que se ajusten a distribución normal ni que haya relaciones sólo "lineales" entre ellos (puede haber relación no lineal), ni que el ruido sea "aditivo distribución normal" es, bueno, más arriesgado.

Estimado de modelo por mínimos cuadrados ordinarios LS (sesgado)

$Y = X\theta$, se podría obtener con:

```
Th_LS_sesgado=pinv(X)*Y
```

```
Th_LS_sesgado = 3×2
    1.8859   -1.8951
   37.8493    9.4852
    4.6439    2.7909
```

```
ThetaLimpio
```

```
ThetaLimpio = 3×2
     2    -2
    40    10
     5     3
```

Es "sesgado" porque con "muuuchos datos" no recuperamos el valor ThetaLimpio.

Estimado modelo mínimos cuadrados totales TLS

*Aquí empieza la solución TLS propiamente dicha si sólo conozco X,Y y las desviaciones típicas del ruido que corrompe cada dato.

Con los datos organizados con 1 muestra 1 FILA

```
size(X)
```

```
ans = 1×2
       90000           3
```

```
size(Y)
```

```
ans = 1×2
       90000           2
```

```
DesTX=diag([dtx1 dtx2 dtx3]); %Dios construyó esta matriz al genera los datos, pero el
```

```
DesTY=diag([dty1 dty2]); %Dios construyó esta matriz al genera los datos, pero el inger
```

Escalamos y centramos los datos a media cero y, supuestamente, desviación típica del "ruido que contamina la señal" unitaria:

```
Xesc=(X-mean(X))*inv(DesTX);
Yesc=(Y-mean(Y))*inv(DesTY);
Datos_esc=[Yesc Xesc];
[N,m]=size(Datos_esc)
```

```
N = 90000
m = 5
```

*Nota, en el vídeo primero he dividido por la desv. típica y luego y restado la media de lo que resultaba. Es correcto pero, vaya, la forma en la que todo el mundo centra los datos es "restar la media y luego dividir por desv. típica", y no al revés. Por ello, lo he cambiado en los materiales.

```
tic
[U,S,V]=svd(Datos_esc/sqrt(N-1),'econ'); %TLS y SVD son lo mismo si todas las variables
%dividiendo los datos por sqrt(N-1), S tiene unidades de "desviación típica muestral"
toc %svd es rápido...
```

```
Elapsed time is 0.004629 seconds.
```

La desviación típica de los componentes principales de Datos_esc es:

```
diag(S)' %si creo que dos cosas se pueden despejar de 3, dos valores sing. deben ser "p
```

```
ans = 1×5
  246.7774   10.2468    4.2047    1.0043    1.0009
```

```
size(U)
```

```
ans = 1×2
       90000           5
```

```
size(V) %hay que coger la 5x5 en TLS
```

```
ans = 1×2
     5     5
```

```
ModEsc=V(:,4:5) %modelo que relaciona vbles. escaladas
```

```
ModEsc = 5×2
   -0.0170   -0.0036
    0.0964    0.3849
    0.5618    0.7037
    0.4923   -0.0900
    0.6576   -0.5903
```

El modelo me dice que "[Yesc Xesc]*ModEsc ≈ 0", datos en fila. TLS no sabe qué es entrada ni salida... como nosotros sí sabemos que $Y = X\theta$, vamos a despejar Y.

$$YM_1 + XM_2 = 0,\ YM_1 = -XM_2,\ Y = -X \cdot M_2 M_1^{-1}$$

```
ModXesc=ModEsc(3:5,:) %lo que multiplica a Xesc en el modelo TLS
```

```
ModXesc = 3×2
    0.5618    0.7037
    0.4923   -0.0900
    0.6576   -0.5903
```

```
ModYesc=ModEsc(1:2,:)  %lo que multiplica a Yesc en el modelo TLS
```

```
ModYesc = 2×2
   -0.0170   -0.0036
    0.0964    0.3849
```

```
ModX=DesTX\ModXesc %en unidades originales, lo que multiplica a X
```

```
ModX = 3×2
    0.0936    0.1173
    1.2308   -0.2250
    0.1315   -0.1181
```

```
ModY=DesTY\ModYesc  %en unidades originales, lo que multiplica a Y
```

```
ModY = 2×2
   -0.0340   -0.0072
    0.0129    0.0513
```

```
ThetaEstimado=-ModX*inv(ModY)  %despejando Y en función de X
```

```
ThetaEstimado = 3×2
    1.9959   -2.0039
   39.9786   10.0192
    5.0034    3.0056
```

Es casi igual a ThetaLimpio, no sesgado:

```
ThetaLimpio
```

```
ThetaLimpio = 3×2
     2    -2
    40    10
     5     3
```

Con los datos organizados con 1 muestra 1 COLUMNA

Si los datos vinieran en "columna":

```
X=X';Y=Y'; %esto sería parte del "haz experimento"...
size(X)
```

```
ans = 1×2
           3       90000
```

```
size(Y)
```

```
ans = 1×2
           2       90000
```

El código cambiaría a:

```
DesTX=diag([dtx1 dtx2 dtx3]);
DesTY=diag([dty1 dty2]);
Xesc=inv(DesTX)*X;
Yesc=inv(DesTY)*Y;
Datos_esc=[Yesc; Xesc]; %OJO: en datos "reales" habría que restar su MEDIA, porque todo
[m,N]=size(Datos_esc)
```

```
m = 5
N = 90000
```

```
Datos_esc=Datos_esc-sum(Datos_esc,2)/N; %sumamos la dimension 2
[U,S,V]=svd(Datos_esc/sqrt(N-1),'econ'); %TLS y SVD son lo mismo si todas las variables
diag(S)'
```

```
ans = 1×5
  246.7774   10.2468    4.2047    1.0043    1.0009
```

```
size(U) %hay que coger la 5x5 en TLS
```

```
ans = 1×2
     5     5
```

```
size(V)
```

```
ans = 1×2
       90000           5
```

```
ModEsc=U(:,4:5)' %modelo que relaciona vbles. escaladas
```

```
ModEsc = 2×5
   -0.0170    0.0964    0.5618    0.4923    0.6576
   -0.0036    0.3849    0.7037   -0.0900   -0.5903
```

El modelo me dice que "ModEsc*[Yesc;Xesc] ≈ 0", datos en fila. TLS no sabe qué es entrada ni salida... como nosotros sí sabemos que $Y = \theta \cdot X$, vamos a despejar Y. Despues de deshacer el escalado, quedara algo como $M_1 Y + M_2 X = 0$, con lo que $Y = -M_1^{-1} M_2 \cdot X$.

```
ModXesc=ModEsc(:,3:5)  %lo que multiplica a Xesc en el modelo TLS
```

```
ModXesc = 2×3
    0.5618    0.4923    0.6576
    0.7037   -0.0900   -0.5903
```

```
ModYesc=ModEsc(:,1:2)   %lo que multiplica a Yesc en el modelo TLS
```

```
ModYesc = 2×2
   -0.0170    0.0964
   -0.0036    0.3849
```

```
ModX=ModXesc/DesTX  %en unidades originales, lo que multiplica a X
```

```
ModX = 2×3
    0.0936    1.2308    0.1315
    0.1173   -0.2250   -0.1181
```

```
ModY=ModYesc/DesTY %en unidades originales, lo que multiplica a Y
```

```
ModY = 2×2
   -0.0340    0.0129
   -0.0072    0.0513
```

```
ThetaEstimado=-inv(ModY)*ModX %despejando Y en función de X
```

```
ThetaEstimado = 2×3
    1.9959   39.9786    5.0034
   -2.0039   10.0192    3.0056
```

```
ThetaLimpio'
```

```
ans = 2×3
     2    40     5
    -2    10     3
```

Ambos métodos (datos en fila o columna) dan, obviamente, el mismo modelo... los datos son los mismos, se trata de multiplicar por la izquierda o por la derecha para escalar, y de seleccionar U o V del SVD.

Addenda: ¿y si no hubiera escalado a ruido idéntico en todos los canales?... pues ¡no funciona!

Bueno, pues no sería "teóricamente correcto"... Si la relación señal-ruido es grande de modo que la varianza del "ruido" no supera la varianza de ninguna "señal", daría parecido, pero en caso contrario no:

```
Datosmal=[Y; X];
[U,S,V]=svd(Datosmal/sqrt(N-1),'econ');
diag(S)' %uyyy... parece que sólo hay "una" ecuación rigiendo los datos?
```

```
ans = 1×5
  129.0372   72.1867   11.9517    5.8589    0.7374
```

```
Modelo_escalamal=U(:,4:5)' %no necesito deshacer escalado, porque no había ningún escal
```

```
Modelo_escalamal = 2×5
   -0.0711    0.3056    0.7522    0.0200   -0.5791
    0.0220   -0.0020   -0.0454   -0.9940   -0.0969
```

```
ThetaMalEscalado=-inv(Modelo_escalamal(:,1:2))*Modelo_escalamal(:,3:5)
```

```
ThetaMalEscalado = 2×3
    1.8753   46.0590    4.6678
   -2.0249   10.6571    2.9817
```

```
ThetaLimpio'
```

```
ans = 2×3
     2    40     5
    -2    10     3
```

```
Modelo_escalabien=[ModY ModX] %the "good one", en coordenadas originales no escalado
```

```
Modelo_escalabien = 2×5
   -0.0340    0.0129    0.0936    1.2308    0.1315
   -0.0072    0.0513    0.1173   -0.2250   -0.1181
```

```
subspace(Modelo_escalamal',Modelo_escalabien')*180/pi
```

```
ans = 3.0200
```

No son "el mismo modelo": nos creemos que los datos 5D viven en un subespacio 3D + ruido, pero los subespacios forman un ángulo de 0.05 radianes (3 grados).

```
subspace([-eye(2);Th_LS_sesgado],Modelo_escalabien')*180/pi
```

```
ans = 1.1850
```

La desviación por "no escalar bien" es comparable (bueno, aquí hasta "peor") a la del "sesgo" por usar LS en vez de TLS.

```
ThetaLimpio'
```

```
ans = 2×3
     2    40     5
    -2    10     3
```

```
ThetaEstimado
```

```
ThetaEstimado = 2×3
    1.9959   39.9786    5.0034
   -2.0039   10.0192    3.0056
```

```
Th_LS_sesgado'
```

```
ans = 2×3
    1.8859   37.8493    4.6439
   -1.8951    9.4852    2.7909
```

*Nota: los valores de 3 grados de offset angular podrían ser aceptables, depende de la aplicación. De hecho, también depende del escalado (pasar algo de voltios a milivoltios cambiaría el ángulo que los comandos de arriba sacan); detalles no son objetivo de este material y, bastantes veces, pueden ser dependientes de aspectos concretos de cada aplicación.

Conclusiones

Si los datos son "muchos", y están generados conforme la teoría espera y conozco las desviaciones típicas de los ruidos, todo funciona como la teoría dice que debería funcionar.

Si los datos son "pocos" o no conozco muy bien las desviaciones típicas para escalar, pues aunque el LS sea "sesgado" el TLS puede que acabe más sesgado todavía si no acierto en el escalado.

La vida "real" es dura: (a) es fácil inferir un resultado incorrecto (engañarme a mí mismo), (b) es fácil convencer a otros de un resultado incorrecto (engañar con la estadística).