Introducción a Identificación de Sistemas Dinámicos: métodos subespacio

Antonio Sala Piqueras

Notas de clase sobre control de sistemas multivariables/complejos

Dept. Ing. Sistemas y Automatica (DISA)
Universitat Politècnica de València (UPV)

Vídeo presentación en:
http://personales.upv.es/asala/YT/V/lsmatr.html

Subsección 1

Mínimos cuadrados matriciales

Video-presentación disponible en:

personales.upv.es/asala/videos/**lsmatr**.html

Mínimos cuadrados versión matricial

- Identificar matriz Θ en $y_{m\times 1} = \Theta_{m\times n} x_{n\times 1}$ se plantea como m problemas **separados** de mínimos cuadrados (uno por cada **fila** de Θ) –No cambia nada desde pto. de vista teórico–.
- Formando, con núm. de muestras (vectoriales) $N \geq n$

$$\Xi_y := \begin{pmatrix} y_1(1) & \dots & y_1(N) \\ \vdots & & \vdots \\ y_m(1) & \dots & y_m(N) \end{pmatrix}_{m\times N} \qquad \Xi_x := \begin{pmatrix} x_1(1) & \dots & x_1(N) \\ \vdots & & \vdots \\ x_n(1) & \dots & x_n(N) \end{pmatrix}_{n\times N}$$

el problema es

$$\hat{\Theta} := \min_{\Theta} \| \overbrace{\underbrace{\Xi_y}_{m\times N} - \underbrace{\Theta}_{m\times n} \underbrace{\Xi_x}_{n\times N}}^{E} \|_F$$

se tiene:

$$\boxed{\hat{\Theta} = \Xi_y \cdot pinv(\Xi_x)} = \Xi_y \cdot \Xi_x^T (\Xi_x \Xi_x^T)^{-1}$$

- Norma Frobenius de una matriz $\|\cdot\|_F$ es la suma de los cuadrados de todos los elementos... o sea, que minimizar $\|E\|_F$ es minimizar el índice mínimo-cuadrático $\sum_{k=1}^{N}\sum_{i=1}^{m} e_i(k)^2$.

Mínimos cuadrados versión matricial

- Identificar matriz Θ en $y_{m\times 1} = \Theta_{m\times n} x_{n\times 1}$ se plantea como m problemas **separados** de mínimos cuadrados (uno por cada **fila** de Θ) –No cambia nada desde pto. de vista teórico–.
- Formando, con núm. de muestras (vectoriales) $N \geq n$

$$\Xi_y := \begin{pmatrix} y_1(1) & \dots & y_1(N) \\ \vdots & & \vdots \\ y_m(1) & \dots & y_m(N) \end{pmatrix}_{m\times N} \qquad \Xi_x := \begin{pmatrix} x_1(1) & \dots & x_1(N) \\ \vdots & & \vdots \\ x_n(1) & \dots & x_n(N) \end{pmatrix}_{n\times N}$$

el problema es

$$\hat{\Theta} := \min_{\Theta} \| \overbrace{\underbrace{\Xi_y}_{m\times N} - \underbrace{\Theta}_{m\times n} \underbrace{\Xi_x}_{n\times N}}^{E} \|_F$$

se tiene:

$$\boxed{\hat{\Theta} = \Xi_y \cdot pinv(\Xi_x)} = \Xi_y \cdot \Xi_x^T (\Xi_x \Xi_x^T)^{-1}$$

- Norma Frobenius de una matriz $\|\cdot\|_F$ es la suma de los cuadrados de todos los elementos... o sea, que minimizar $\|E\|_F$ es minimizar el índice minimo-cuadrático $\sum_{k=1}^{N} \sum_{i=1}^{m} e_i(k)^2$.

Interpretación estadística

Como en el caso θ vector, se mantienen conceptos de ML, mejor predicción lineal, de un modelo $y = \Theta x + \mathcal{E}$ siendo $\mathcal{E}$ un vector de variables de varianza identidad:

- la solución es la mejor predicción lineal del vector 'y' dado el vector 'x', esto es:

$$\hat{\Theta} = \Sigma_{yx}\Sigma_x^{-1}$$

donde Σ_{yx} es matriz $m \times n$, Σ_x es matriz $n \times n$ dadas por:

$$\Sigma_{yx} = E(yx^T) = \Xi_y\Xi_x^T/(N-1), \quad \Sigma_x = E(xx^T) = \Xi_x\Xi_x^T/(N-1)$$

Subsección 2

Algoritmo subespacio

Video-presentación disponible en:

personales.upv.es/asala/videos/**subsp**.html

Identificación subespacio de modelos espacio de estado

Concepto de estado: información pasada que sirve para predecir el futuro. En una ec. en diferencias discreta como:

$$y_{k+1} = -a_1 y_k - a_2 y_{k-1} + b_1 u_k + b_2 u_{k-1}$$

El "estado" podría ser $$x_k := \begin{pmatrix} y_{k-1} \\ y_k \\ u_{k-1} \end{pmatrix}$$

$$\begin{pmatrix} y_k \\ y_{k+1} \\ u_k \end{pmatrix} = x_{k+1} = \begin{pmatrix} 0 & 1 & 0 \\ -a_2 & -a_1 & b_2 \\ 0 & 0 & 0 \end{pmatrix} x_k + \begin{pmatrix} 0 \\ b_1 \\ 1 \end{pmatrix} u_k$$

$$y_k = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} x_k + 0 u_k$$

*La repr. es correcta, pero no **mínima**: hay 1 estado **no** observable; existe repr. interna de orden 2.

```
A=[0 1 0;-1.5 .3 .2;0 0 0]; B=[0 0.1 1]'; C=[0 1 0]; svd(ctrb(A,B))
svd(obsv(A,C)) null(obsv(A,C))
```

Generalización de la idea

En proceso lineal, determinista, orden n, combinaciones lineales de entradas y salidas pasadas permiten predecir con exactitud presente y futuro (junto con entradas presentes y futuras). Eligiendo horizontes de predicción h_1, y de memoria pasada h_2, h_3 (mayores de n):

$$\underbrace{\begin{pmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+h_1} \end{pmatrix}}_{Y_{fut,k}} = \bar{M}_1 \underbrace{\begin{pmatrix} y_{k-h_2} \\ \vdots \\ y_k \\ u_{k-h_3} \\ \vdots \\ u_{k-1} \end{pmatrix}}_{W_{past,k}} + M_2 \underbrace{\begin{pmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+h_1} \end{pmatrix}}_{U_{fut,k}}$$

*Como $h_1 > n$, $\boxed{rango(M_1) = n}$ – sólo se necesitan n combinaciones de variables pasadas para predecir el futuro –.

En caso no determinista (datos con ruido) el estimado $\bar{M}_1$ que mejor ajusta por mínimos cuadrados los datos no tiene rango n igual al del sistema que los genera, debido al ruido... la idea es hacer SVD y suponer cero los valores más pequeños, detalles a continuación.

Generalización de la idea

En proceso lineal, determinista, orden n, combinaciones lineales de entradas y salidas pasadas permiten predecir con exactitud presente y futuro (junto con entradas presentes y futuras). Eligiendo horizontes de predicción h_1, y de memoria pasada h_2, h_3 (mayores de n):

$$\underbrace{\begin{pmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+h_1} \end{pmatrix}}_{Y_{fut,k}} = \bar{M}_1 \underbrace{\begin{pmatrix} y_{k-h_2} \\ \vdots \\ y_k \\ u_{k-h_3} \\ \vdots \\ u_{k-1} \end{pmatrix}}_{W_{past,k}} + M_2 \underbrace{\begin{pmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+h_1} \end{pmatrix}}_{U_{fut,k}}$$

*Como $h_1 > n$, $\boxed{rango(M_1) = n}$ – sólo se necesitan n combinaciones de variables pasadas para predecir el futuro –.

En caso no determinista (datos con ruido) el estimado $\bar{M}_1$ que mejor ajusta por mínimos cuadrados los datos no tiene rango n igual al del sistema que los genera, debido al ruido... la idea es hacer SVD y suponer cero los valores más pequeños, detalles a continuación.

Modelo de regresión

Si planteamos la ecuación para múltiples datos de las muestras entrada-salida tomadas del proceso, debemos identificar M_1 y M_2 por mínimos cuadrados matriciales:

$$\underbrace{\begin{pmatrix} y_k & y_{k+1} & \cdots \\ y_{k+1} & y_{k+2} & \cdots \\ \vdots & \vdots & \\ y_{k+h_1} & y_{k+h_1+1} & \cdots \end{pmatrix}}_{Y_{fut}\ [(h1+1)\times\nu]} \approx M_1 \underbrace{\begin{pmatrix} y_{k-h_2} & y_{k-h_2+1} & \cdots \\ \vdots & \vdots & \\ y_k & y_{k+1} & \cdots \\ u_{k-h_3} & u_{k-h_3+1} & \cdots \\ \vdots & \vdots & \\ u_{k-1} & u_k & \cdots \end{pmatrix}}_{W_{past}\ [(h_2+h_3+1)\times\nu]} + M_2 \underbrace{\begin{pmatrix} u_k & u_{k+1} & \cdots \\ u_{k+1} & u_{k+2} & \cdots \\ \vdots & \vdots & \\ u_{k+h_1} & u_{k+h_1+1} & \cdots \end{pmatrix}}_{U_{fut}\ [(h1+1)\times\nu]}$$

$$Y_{fut} \approx (M_1\ \ M_2)\cdot\begin{pmatrix} W_{past} \\ U_{fut} \end{pmatrix}$$

*si M_1 tiene rango n, $n \leq (h_2+h_3+1)$, el SVD de M_1 nos permitirá establecer un predictor con menos variables. Detalles a continuación.

Identificación subespacio

[1.] Normalización y preblanqueado: Dividir el pasado en componentes "independientes" no correlados

- para que "tener" o "no tener" unos no influya en la mejor predicción dados los otros*, que será usado en la estimación del orden del sistema o para obtener "*el mejor modelo de un orden prefijado*").

$$W_{past}^{white} = \left(\sqrt{\Sigma_{W_{past}}}\right)^{\dagger} \cdot W_{past}$$

*Dada una matriz $\Sigma \geq 0$, entenderemos $(\sqrt{\Sigma})^{\dagger} = V \cdot pinv(\sqrt{D}) \cdot V^T$, siendo Σ diagonalizada como $\Sigma = VDV^T$, y $pinv(\sqrt{D})$ la inversa de la raíz cuadrada de los elementos diagonales, o cero en los elementos diagonales de D que sean cero.

*detalles en regresión PLS, http://personales.upv.es/asala/videos/pcareg.html

Identificación subespacio (cont.)

[2.] Identificamos M_1 y M_2 por mínimos cuadrados matriciales a partir de series de datos experimentales entrada-salida (ARX multivariable), después de preblanqueado de la matriz verde (paso 1):

$$\underbrace{\begin{pmatrix} y_k & y_{k+1} & \cdots \\ y_{k+1} & y_{k+2} & \cdots \\ \vdots & \vdots & \\ y_{k+h_1} & y_{k+h_1+1} & \cdots \end{pmatrix}}_{Y_{fut}\ [(h1+1)\times\nu]} \approx M_1 \underbrace{\begin{pmatrix} y_{k-h_2} & y_{k-h_2+1} & \cdots \\ \vdots & \vdots & \\ y_k & y_{k+1} & \cdots \\ u_{k-h_3} & u_{k-h_3+1} & \cdots \\ \vdots & \vdots & \\ u_{k-1} & u_k & \cdots \end{pmatrix}}_{W_{past}^{white}\ [(h_2+h_3+1)\times\nu]} + M_2 \underbrace{\begin{pmatrix} u_k & u_{k+1} & \cdots \\ u_{k+1} & u_{k+2} & \cdots \\ \vdots & \vdots & \\ u_{k+h_1} & u_{k+h_1+1} & \cdots \end{pmatrix}}_{U_{fut}\ [(h1+1)\times\nu]}$$

$$\boxed{Y_{fut} \approx (M_1\ \ M_2)\cdot\begin{pmatrix} W_{past}^{white} \\ U_{fut}\end{pmatrix}, \qquad M_1 \text{ es } [(h1+1)\times(h_2+h_3+1)]}$$

Identificación subespacio: selección de orden

[3.] Hacemos SVD de $M_1 = USV^T$. **Orden estimado** n del sistema: núm. de elementos de S "significativamente más grandes que el resto" (no nulos, si los datos fueran sin ruido, lo que nunca ocurre).

$$Y_{fut} = USV^T W_{past}^{white} + M_2 U_{fut}$$

$$S \approx [\bar{S}_{(h_1+1)\times n} \ \ 0]$$

*Estamos determinando cuántas combinaciones lineales de los elementos de W_{past}^{white} (incorrelados entre sí) tienen un efecto significativo para predecir Y_{fut}.

Identificación subespacio: generación de estado

Continuamos de pág. anterior, por comodidad:

$$Y_{fut} = USV^T W_{past}^{white} + M_2 U_{fut}$$

$$S \approx [\bar{S}_{(h_1+1)\times n} \;\; 0]$$

[4.] Seleccionando las columnas de V asociadas a los valores singulares "no nulos", $\widetilde{V}_x = V(:, 1:n)$, tenemos las combinaciones de entradas y salidas pasadas que influyen en el futuro: hemos encontrado **estados de la realización**.

$$Y_{fut} = U\;\bar{S}\,\widetilde{V}_x^T W_{past}^{white} + M_2 U_{fut}$$

Hacemos el cambio $\Psi_x = \widetilde{V}_x^T W_{past}$; Ψ_x tiene n filas. Las columnas de Ψ_x pueden asimilarse a los valores estimados $\hat{x}_k, \hat{x}_{k+1}, \ldots$

$$Y_{fut} = U\;\bar{S}\;\underbrace{\Psi_x}_{n\times\nu} + M_2 U_{fut}$$

*Por "equilibrar" numéricamente, el cambio en el ejemplo del video de Matlab será $\Psi_x = \sqrt{\bar{S}}\,\widetilde{V}_x^T W_{past}^{white}$.

Identificación subespacio: generación de estado

Continuamos de pág. anterior, por comodidad:

$$Y_{fut} = USV^T W_{past}^{white} + M_2 U_{fut}$$

$$S \approx [\bar{S}_{(h_1+1)\times n} \ 0]$$

[4.] Seleccionando las columnas de V asociadas a los valores singulares "no nulos", $\widetilde{V}_x = V(:, 1:n)$, tenemos las combinaciones de entradas y salidas pasadas que influyen en el futuro: hemos encontrado **estados de la realización**.

$$Y_{fut} = U \ \bar{S} \widetilde{V}_x^T W_{past}^{white} + M_2 U_{fut}$$

Hacemos el cambio $\Psi_x = \widetilde{V}_x^T W_{past}$; Ψ_x tiene n filas. Las columnas de Ψ_x pueden asimilarse a los valores estimados $\hat{x}_k$, $\hat{x}_{k+1}$, ...

$$Y_{fut} = U \ \bar{S} \underbrace{\Psi_x}_{n\times\nu} + M_2 U_{fut}$$

*Por "equilibrar" numéricamente, el cambio en el ejemplo del video de Matlab será $\Psi_x = \sqrt{\bar{S}} \widetilde{V}_x^T W_{past}^{white}$.

Identificación subespacio: Matrices A, B, C, D.

[5.] Las matrices del sistema se generan obteniendo $\hat{A}$, $\hat{B}$ con mínimos cuadrados matriciales, siendo $\hat{x}_k$ la columna k de Ψ_x, tamaño $n \times 1$, en

$$\underbrace{\begin{pmatrix} \hat{x}_{k+1} & \hat{x}_{k+2} & \dots \end{pmatrix}}_{\Psi_{x+}} \approx \hat{A} \cdot \underbrace{\begin{pmatrix} \hat{x}_k & \hat{x}_{k+1} & \dots \end{pmatrix}}_{\Psi_x} + \hat{B} \cdot \begin{pmatrix} u_k & u_{k+1} & \dots \end{pmatrix}$$

$$\underbrace{\begin{pmatrix} \hat{x}_{k+1} & \hat{x}_{k+2} & \dots \end{pmatrix}}_{\Psi_{x+}} \approx \begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix} \cdot \underbrace{\begin{pmatrix} \hat{x}_k & \hat{x}_{k+1} & \dots \\ u_k & u_{k+1} & \dots \end{pmatrix}}_{\Xi}$$

obteniendo: $\begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix} = \Psi_{x+} \cdot pinv(\Xi)$. Luego, resolviendo $\hat{C}$, $\hat{D}$ de forma análoga en:

$$\begin{pmatrix} y_k & y_{k+1} & \dots \end{pmatrix} \approx \hat{C} \cdot \begin{pmatrix} \hat{x}_k & \hat{x}_{k+1} & \dots \end{pmatrix} + \hat{D} \cdot \begin{pmatrix} u_k & u_{k+1} & \dots \end{pmatrix}$$

[FIN DEL ALGORITMO]

Conclusiones/Discusión

El algoritmo **subespacio** permite obtener matrices A, B, C, D a partir de series de datos entrada-salida (multivariables: u_k, y_k pueden ser vectores) usando **mínimos cuadrados** (matriciales) y **SVD** (para determinar el orden de la realización).

*Existen variaciones en la literatura, con pesos W_1, W_2 en salidas y regresores (detalles omitidos por brevedad).

Matlab System ID Toolbox: `n4sid`.

Identificación del observador óptimo (Kalman Filter)

Como resultado adicional de gran interés, resolviendo por mínimos cuadrados matriciales:

$$\begin{pmatrix}\hat{x}_{k+1} & \hat{x}_{k+2} & \dots\end{pmatrix} \approx \hat{A}_{KF} \cdot \begin{pmatrix}\hat{x}_k & \hat{x}_{k+1} & \dots\end{pmatrix} \\ + \hat{B}_{KF} \cdot \begin{pmatrix}u_k & u_{k+1} & \dots\end{pmatrix} + \hat{L} \cdot \begin{pmatrix}y_{k+1} & y_{k+2} & \dots\end{pmatrix}$$

se obtienen **directamente de datos** las aproximaciones de las matrices del filtro de
Kalman (observ. adelantado, `dlqe`) cuyo valor exacto dado un modelo sería:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_{k+1} - C(A\hat{x}_k + Bu_k)) = \bar{A}_{KF}\hat{x} + \bar{B}_{KF}u + Ly_{k+1}$$

siendo $\bar{A}_{KF} = (I - LC)A$, $\bar{B}_{KF} = (I - LC)B$.

*Se identifican directamente **observadores**. No necesario estimar matrices de varianza de ruido de proceso/medida Q, R en `dlqe(A,C,G,Q,R)`. Los detalles teóricos de cómo se obtiene el observador dado un modelo aparecen en la video-presentación http://personales.upv.es/asala/videos/est2k.html.

Ejemplo Matlab

A partir de datos entrada-salida, se identifican estados, representación A, B, C, D, y matrices del filtro de Kalman según el algoritmo presentado en:

personales.upv.es/asala/videos/**subspml**.html