# PCR, PLS, LBD e Gradiente Coniugato

Barbarino Giovanni

Università di Pisa

8 aprile 2014

# Problema ai Minimi Quadrati

- Least-Squares Problem:

$$\min_{x \in \mathbb{R}^n} \|Ax - y\|$$

- Proiezione su un sottospazio:

$$\min_{x \in \mathbb{R}^k} \|AVx - y\|$$

- Algoritmi Proposti:
  - PCR (Principal Component Regression)
  - PLS (Partial Least-Squares)
  - LBD (Lanczon Bidiagonalization)
  - CG (Conjugated Gradient )

# Problema ai Minimi Quadrati

- Least-Squares Problem:

$$\min_{x \in \mathbb{R}^n} \|Ax - y\|$$

- Proiezione su un sottospazio:

$$\min_{x \in \mathbb{R}^k} \|AVx - y\|$$

- Algoritmi Proposti:
  - PCR (Principal Component Regression)
  - PLS (Partial Least-Squares)
  - LBD (Lanczon Bidiagonalization)
  - CG (Conjugated Gradient )

# Problema ai Minimi Quadrati

- Least-Squares Problem:

$$\min_{x \in \mathbb{R}^n} \|Ax - y\|$$

- Proiezione su un sottospazio:

$$\min_{x \in \mathbb{R}^k} \|AVx - y\|$$

- Algoritmi Proposti:
    - PCR (Principal Component Regression)
    - PLS (Partial Least-Squares)
    - LBD (Lanczon Bidiagonalization)
    - CG (Conjugated Gradient )

# PCR (TSVD)

Scomposizione SVD della matrice:

$$A = S \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} Q^t, \quad \Sigma_r = diag(\sigma_1, \dots, \sigma_r), \quad \sigma_1 \geq \cdots \geq \sigma_r > 0$$

**PCR Solution and Residual**

$$x_{pcr}^{(k)} = \sum_{i=1}^{k} \frac{s_i^t y}{\sigma_i} q_i,$$

$$R_k^2 = \left\| A x_{pcr}^{(k)} - y \right\|^2 = \sum_{i=k+1}^{m} (s_i^t y)^2$$

La scomposizione SVD *non dipende da y*

# PCR (TSVD)

Scomposizione SVD della matrice:

$$A = S \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} Q^t, \quad \Sigma_r = diag(\sigma_1, \dots, \sigma_r), \quad \sigma_1 \geq \cdots \geq \sigma_r > 0$$

## PCR Solution and Residual

$$x_{pcr}^{(k)} = \sum_{i=1}^{k} \frac{s_i^t y}{\sigma_i} q_i,$$

$$R_k^2 = \left\| A x_{pcr}^{(k)} - y \right\|^2 = \sum_{i=k+1}^{m} (s_i^t y)^2$$

La scomposizione SVD *non dipende da y*

# PCR (TSVD)

Scomposizione SVD della matrice:

$$A = S \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} Q^t, \quad \Sigma_r = diag(\sigma_1, \ldots, \sigma_r), \quad \sigma_1 \geq \cdots \geq \sigma_r > 0$$

## PCR Solution and Residual

$$x_{pcr}^{(k)} = \sum_{i=1}^{k} \frac{s_i^t y}{\sigma_i} q_i,$$

$$R_k^2 = \left\| A x_{pcr}^{(k)} - y \right\|^2 = \sum_{i=k+1}^{m} (s_i^t y)^2$$

**La scomposizione SVD *non dipende da* $y$**

# PLS

## Algoritmo PLS NIPALS

$A_0 = A$

**for** $i = 1, 2, \ldots, k$ **do**

$$w_i = A_{i-1}^t y \qquad w_i = w_i / \|w_i\|$$
$$t_i = A_{i-1} w_i \qquad t_i = t_i / \|t_i\|$$
$$p_i = A_{i-1}^t t_i$$
$$A_i = A_{i-1} - t_i p_i^t$$

**end for**

## Soluzione del PLS

$$x_{pls}^{(k)} = W_k (P_k^t W_k)^{-1} T_k^t y$$

La soluzione è ristretta al Sottospazio di Krylov $K_k(A^tA, A^ty)$

# PLS

## Algoritmo PLS NIPALS

$A_0 = A$

**for** $i = 1, 2, \ldots, k$ **do**

$$w_i = A_{i-1}^t y \qquad w_i = w_i / \|w_i\|$$

$$t_i = A_{i-1} w_i \qquad t_i = t_i / \|t_i\|$$

$$p_i = A_{i-1}^t t_i$$

$$A_i = A_{i-1} - t_i p_i^t$$

**end for**

## Soluzione del PLS

$$x_{pls}^{(k)} = W_k (P_k^t W_k)^{-1} T_k^t y$$

La soluzione è ristretta al Sottospazio di Krylov $K_k(A^t A, A^t y)$

# LBD

## Algoritmo LBD

$v_1 = A^t y \qquad v_1 = 1/\|v_1\|$

$u_1 = A v_1 \qquad \alpha_1 = 1/\|u_1\| \qquad u_1 = \alpha_1 u_1$

**for** $i = 2, 3, \ldots, k$ **do**

$\quad v_i = A^t u_{i-1} - \alpha_{i-1} v_{i-1} \qquad \gamma_{i-1} = 1/\|v_i\| \qquad v_i = \gamma_{i-1} v_i$

$\quad u_i = A v_i - \gamma_{i-1} u_{i-1} \qquad \alpha_i = 1/\|u_i\| \qquad u_i = \alpha_i u_i$

**end for**

## Soluzione dell'LBD

$$x_{lbd}^{(k)} = V_k B_k^{-1} U_k^t y$$

La soluzione è ristretta al Sottospazio di Krylov $K_k(A^t A, A^t y)$

Nel passo iterativo, si perde la relazione con $y$

# LBD

## Algoritmo LBD

$$
\begin{array}{lll}
v_1 = A^t y & v_1 = 1/\|v_1\| & \\
u_1 = A v_1 & \alpha_1 = 1/\|u_1\| & u_1 = \alpha_1 u_1
\end{array}
$$

**for** $i = 2, 3, \ldots, k$ **do**

$$
\begin{array}{lll}
v_i = A^t u_{i-1} - \alpha_{i-1} v_{i-1} & \gamma_{i-1} = 1/\|v_i\| & v_i = \gamma_{i-1} v_i \\
u_i = A v_i - \gamma_{i-1} u_{i-1} & \alpha_i = 1/\|u_i\| & u_i = \alpha_i u_i
\end{array}
$$

**end for**

## Soluzione dell'LBD

$$x_{lbd}^{(k)} = V_k B_k^{-1} U_k^t y$$

La soluzione è ristretta al Sottospazio di Krylov $K_k(A^t A, A^t y)$

Nel passo iterativo, si perde la relazione con $y$

# LBD

## Algoritmo LBD

$$v_1 = A^t y \qquad v_1 = 1/\|v_1\|$$
$$u_1 = A v_1 \qquad \alpha_1 = 1/\|u_1\| \qquad u_1 = \alpha_1 u_1$$
**for** $i = 2, 3, \ldots, k$ **do**
$$v_i = A^t u_{i-1} - \alpha_{i-1} v_{i-1} \qquad \gamma_{i-1} = 1/\|v_i\| \qquad v_i = \gamma_{i-1} v_i$$
$$u_i = A v_i - \gamma_{i-1} u_{i-1} \qquad \alpha_i = 1/\|u_i\| \qquad u_i = \alpha_i u_i$$
**end for**

## Soluzione dell'LBD

$$x_{lbd}^{(k)} = V_k B_k^{-1} U_k^t y$$

La soluzione è ristretta al Sottospazio di Krylov $K_k(A^t A, A^t y)$

**Nel passo iterativo, si perde la relazione con** $y$

# Gradiente Coniugato (CG)

## Algoritmo del CG

$x = 0 \qquad r = v = y - Ax$

**while** $\|r\| < \epsilon$ **do**

$\quad t = Av \qquad \alpha = v^t r / v^t t \qquad x = x + \alpha v$

$\quad r = y - Ax \qquad \beta = r^t t / v^t t \qquad v = r - \beta v$

**end while**

## Soluzione del CG

Se applicato all'equazione normale $A^t Ax = A^t y$, dopo $k$ passi ritorna la soluzione del problema ai minimi quadrati ristretto al Sottospazio di Krylov $K_k(A^t A, A^t y)$

Gli ultimi tre algoritmi coincidono algebricamente, quindi dobbiamo solo confrontarli numericamente

# Gradiente Coniugato (CG)

## Algoritmo del CG

$x = 0 \qquad r = v = y - Ax$

**while** $\|r\| < \epsilon$ **do**

$\quad t = Av \qquad \alpha = v^t r / v^t t \qquad x = x + \alpha v$

$\quad r = y - Ax \qquad \beta = r^t t / v^t t \qquad v = r - \beta v$

**end while**

## Soluzione del CG

Se applicato all'equazione normale $A^t A x = A^t y$, dopo $k$ passi ritorna la soluzione del problema ai minimi quadrati ristretto al Sottospazio di Krylov $K_k(A^t A, A^t y)$

Gli ultimi tre algoritmi coincidono algebricamente, quindi dobbiamo solo confrontarli numericamente

# Gradiente Coniugato (CG)

## Algoritmo del CG

$x = 0 \qquad r = v = y - Ax$

**while** $\|r\| < \epsilon$ **do**

$\quad t = Av \qquad \alpha = v^t r / v^t t \qquad x = x + \alpha v$

$\quad r = y - Ax \qquad \beta = r^t t / v^t t \qquad v = r - \beta v$

**end while**

## Soluzione del CG

Se applicato all'equazione normale $A^t A x = A^t y$, dopo $k$ passi ritorna la soluzione del problema ai minimi quadrati ristretto al Sottospazio di Krylov $K_k(A^t A, A^t y)$

Gli ultimi tre algoritmi coincidono algebricamente, quindi dobbiamo solo confrontarli numericamente

# Esperimento di Stabilità

Testiamo la perdita di ortogonalità delle basi di PLS, LBD e Gradiente Coniugato

## Esperimento

Prese 50 matrici casuali 100 per 100 e 50 vettori noti casuali, generiamo con i tre metodi le basi ortogonali. Preso il primo vettore di base, riportiamo il suo prodotto con i primi 50 vettori di base generati.

# Esperimento di Stabilità

Testiamo la perdita di ortogonalità delle basi di PLS, LBD e Gradiente Coniugato

## Esperimento

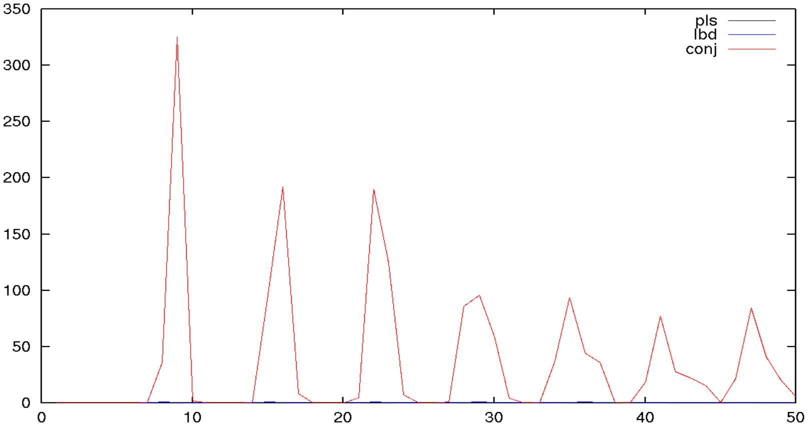
Prese 50 matrici casuali 100 per 100 e 50 vettori noti casuali, generiamo con i tre metodi le basi ortogonali. Preso il primo vettore di base, riportiamo il suo prodotto con i primi 50 vettori di base generati.

# PLS: una scelta di stabilità

- I tre algoritmi assieme

# PLS: una scelta di stabilità

- I tre algoritmi assieme

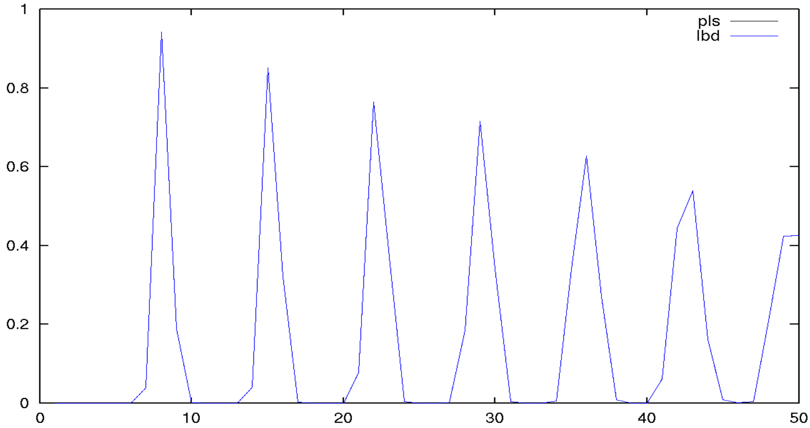


# PLS: una scelta di stabilità

- PLS e LBD
- I tre algoritmi assieme

# PLS: una scelta di stabilità

- PLS e LBD

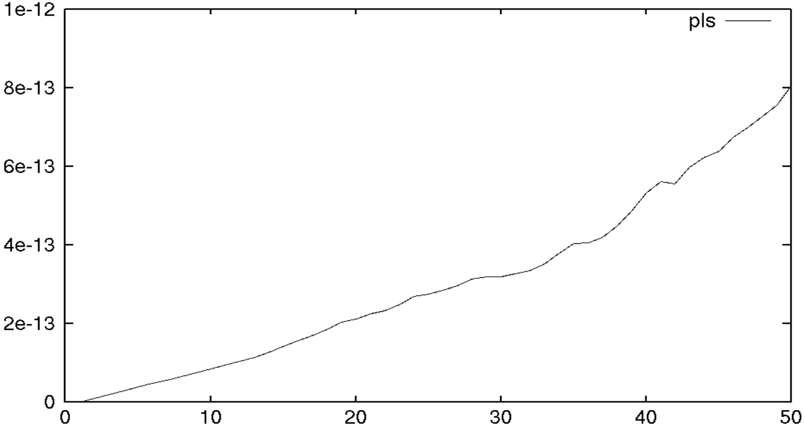


# PLS: una scelta di stabilità

- PLS
- PLS e LBD
- I tre algoritmi assieme

# PLS: una scelta di stabilità

- PLS

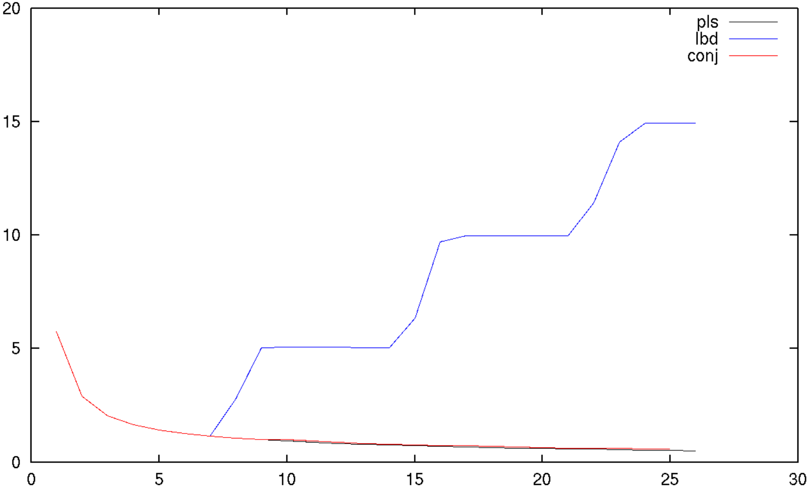


# Il break nell'LBD

Questo influenza il residuo?

# Il break nell'LBD

Questo influenza il residuo?



# Riassunto delle proprietà

- il **PLS** è stabile, ma costoso poichè non mantiene la struttura
  $\Longrightarrow$ vantaggioso in casi di taglia moderata
- L'**LBD** mantiene la struttura, offre una scomposizione, ma soffre pesantemente il breakdown
  $\Longrightarrow$ vantaggioso in casi di matrice strutturata, e analisi dei primi valori
- Il **CG** mantiene la struttura, ma è mal condizionato
  $\Longrightarrow$ vantaggioso in casi di matrice strutturata di taglia grande

# Riassunto delle proprietà

- il **PLS** è stabile, ma costoso poichè non mantiene la struttura
  $\implies$ vantaggioso in casi di taglia moderata
- L'**LBD** mantiene la struttura, offre una scomposizione, ma soffre pesantemente il breakdown
  $\implies$ vantaggioso in casi di matrice strutturata, e analisi dei primi valori
- Il **CG** mantiene la struttura, ma è mal condizionato
  $\implies$ vantaggioso in casi di matrice strutturata di taglia grande

# Riassunto delle proprietà

- il **PLS** è stabile, ma costoso poichè non mantiene la struttura
  $\Longrightarrow$ vantaggioso in casi di taglia moderata
- L'**LBD** mantiene la struttura, offre una scomposizione, ma soffre pesantemente il breakdown
  $\Longrightarrow$ vantaggioso in casi di matrice strutturata, e analisi dei primi valori
- Il **CG** mantiene la struttura, ma è mal condizionato
  $\Longrightarrow$ vantaggioso in casi di matrice strutturata di taglia grande

# Riassunto delle proprietà

- il **PLS** è stabile, ma costoso poichè non mantiene la struttura
  $\Longrightarrow$ vantaggioso in casi di taglia moderata
- L'**LBD** mantiene la struttura, offre una scomposizione, ma soffre pesantemente il breakdown
  $\Longrightarrow$ vantaggioso in casi di matrice strutturata, e analisi dei primi valori
- Il **CG** mantiene la struttura, ma è mal condizionato
  $\Longrightarrow$ vantaggioso in casi di matrice strutturata di taglia grande

Per i confronti con il PCR, useremo il PLS

# Utilità degli algoritmi

## Problema

Se la matrice è quasi singolare, la soluzione ai minimi quadrati può avere norma elevata

## Soluzione

Risolvere il problema ristretto a sottospazi di dimensione bassa

- un sottospazio di Krylov
- un sottospazio dato dall'SVD

## Parametri Considerati

Consideriamo gli algoritmi PCR e PLS, confrontandone:

- Norma del Residuo
- Norma della Soluzione Approssimata

# Utilità degli algoritmi

## Problema

Se la matrice è quasi singolare, la soluzione ai minimi quadrati può avere norma elevata

## Soluzione

Risolvere il problema ristretto a sottospazi di dimensione bassa

- un sottospazio di Krylov
- un sottospazio dato dall'SVD

## Parametri Considerati

Consideriamo gli algoritmi PCR e PLS, confrontandone:

- Norma del Residuo
- Norma della Soluzione Approssimata

# Utilità degli algoritmi

## Problema

Se la matrice è quasi singolare, la soluzione ai minimi quadrati può avere norma elevata

## Soluzione

Risolvere il problema ristretto a sottospazi di dimensione bassa

- un sottospazio di Krylov
- un sottospazio dato dall'SVD

## Parametri Considerati

Consideriamo gli algoritmi PCR e PLS, confrontandone:

- Norma del Residuo
- Norma della Soluzione Approssimata

# PCR Modificato (MPCR)

## Residuo del PCR

$$R_k^2 = \sum_{i=k+1}^{m} (s_i^t y)^2$$

Questo risultato ci induce a formulare una modifica al PCR, per migliorare la riduzione del residuo:

## PCR Modificato

Data $\tau$ una permutazione degli indici, in modo che $s_{\tau(1)}^t y \geq s_{\tau(2)}^t y \geq \ldots s_{\tau(r)}^t y$, allora la soluzione diventa $x_{pcrm}^{(k)} = \sum_{i=1}^{k} \frac{s_{\tau(i)}^t y}{\sigma_{\tau(i)}} q_{\tau(i)}$

Confrontiamo quindi PLS, PCR, MPCR

# PCR Modificato (MPCR)

## Residuo del PCR

$$R_k^2 = \sum_{i=k+1}^{m} (s_i^t y)^2$$

Questo risultato ci induce a formulare una modifica al PCR, per migliorare la riduzione del residuo:

## PCR Modificato

Data $\tau$ una permutazione degli indici, in modo che $s_{\tau(1)}^t y \geq s_{\tau(2)}^t y \geq \dots s_{\tau(r)}^t y$, allora la soluzione diventa

$$x_{pcrm}^{(k)} = \sum_{i=1}^{k} \frac{s_{\tau(i)}^t y}{\sigma_{\tau(i)}} q_{\tau(i)}$$

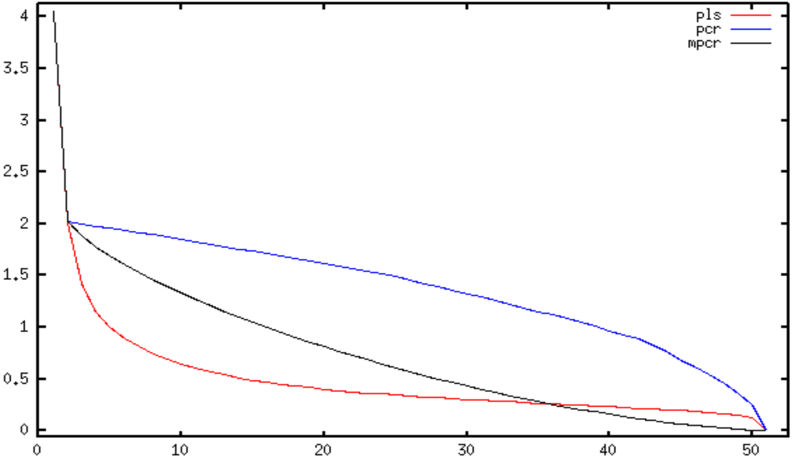
Confrontiamo quindi PLS, PCR, MPCR

# Residuo al Caso Medio

- Con lo stesso esperimento di prima, calcoliamo il residuo..

# Residuo al Caso Medio

- Con lo stesso esperimento di prima, calcoliamo il residuo..

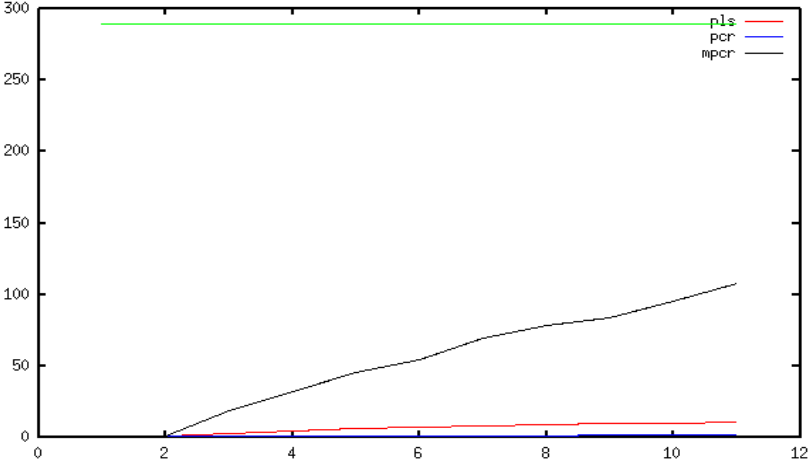


# Residuo al Caso Medio

- .. e la norma della soluzione

# Residuo al Caso Medio

- .. e la norma della soluzione



# Esperimenti Effettuati

In media, il PLS riduce il residuo più velocemente dell'MPCR, e mantiene la norma della soluzione bassa

Questo non succede sempre, quindi proviamo a verificare alcuni casi

**Esperimenti**

Concentrandoci sui primi pochi passi, analizziamo:

- y uniforme
- y e A sbilanciati nello stesso modo
- y e A sbilanciati in maniera opposta
- y concentrato su poche componenti

# Esperimenti Effettuati

In media, il PLS riduce il residuo più velocemente dell'MPCR, e mantiene la norma della soluzione bassa

Questo non succede sempre, quindi proviamo a verificare alcuni casi

**Esperimenti**

Concentrandoci sui primi pochi passi, analizziamo:

- y uniforme
- y e A sbilanciati nello stesso modo
- y e A sbilanciati in maniera opposta
- y concentrato su poche componenti

# Esperimenti Effettuati

In media, il PLS riduce il residuo più velocemente dell'MPCR, e mantiene la norma della soluzione bassa

Questo non succede sempre, quindi proviamo a verificare alcuni casi

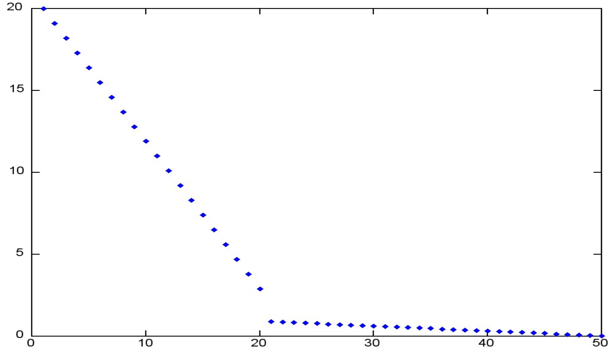
## Esperimenti

Concentrandoci sui primi pochi passi, analizziamo:

- y uniforme
- y e A sbilanciati nello stesso modo
- y e A sbilanciati in maniera opposta
- y concentrato su poche componenti

# $y$ Uniforme

Scegliamo $y$ con componenti uniformi, e prendiamo $A$ diagonale, con elementi riportati di seguito:

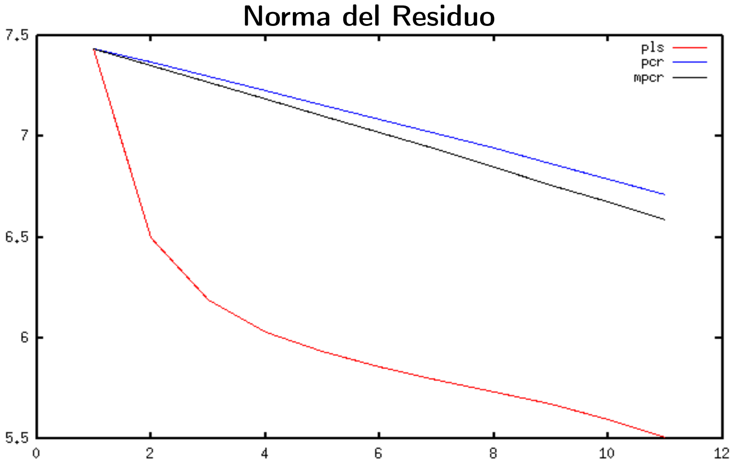


# $y$ Uniforme

- $y$ uniforme a circa 1

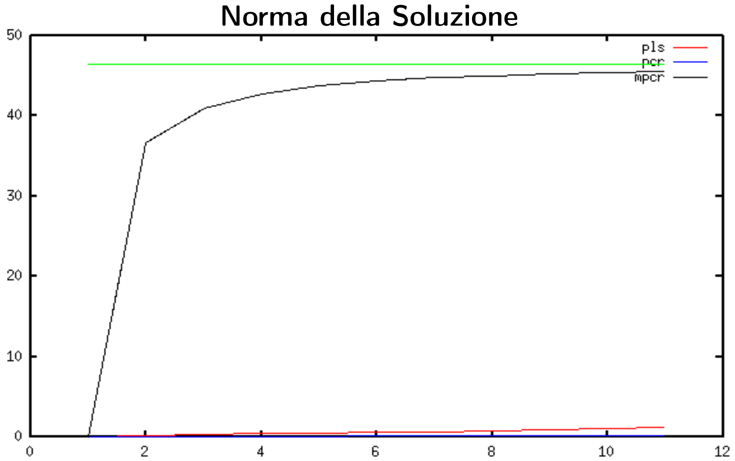
# $y$ Uniforme

- $y$ uniforme a circa 1



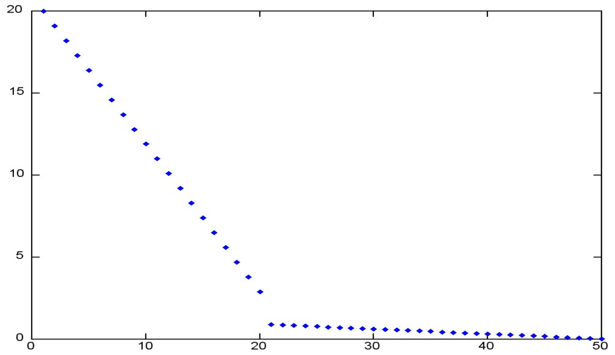
# $y$ Uniforme

- $y$ uniforme a circa 1



# $y$ Concentrata

Presa $A$ dell'esempio precedente, scegliamo $y$ concentrata su poche componenti, corrispondenti a valori singolari alti, medi e bassi di $A$.

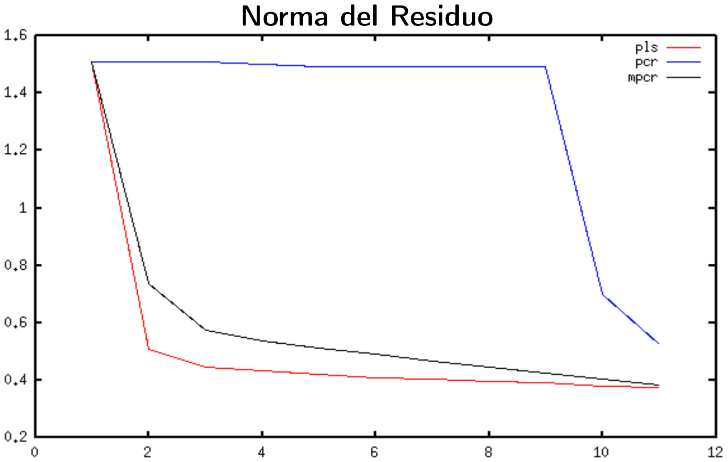


# $y$ Concentrata

- $y$ concentrata sul terzo valore singolare

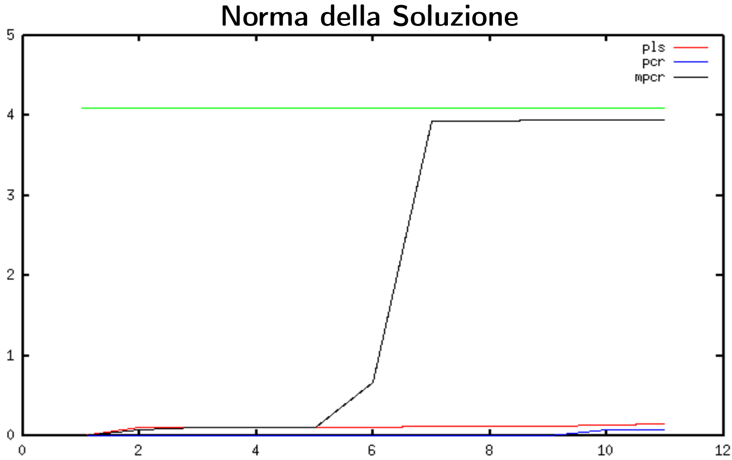
# $y$ Concentrata

- $y$ concentrata sul terzo valore singolare



# $y$ Concentrata

- $y$ concentrata sul terzo valore singolare

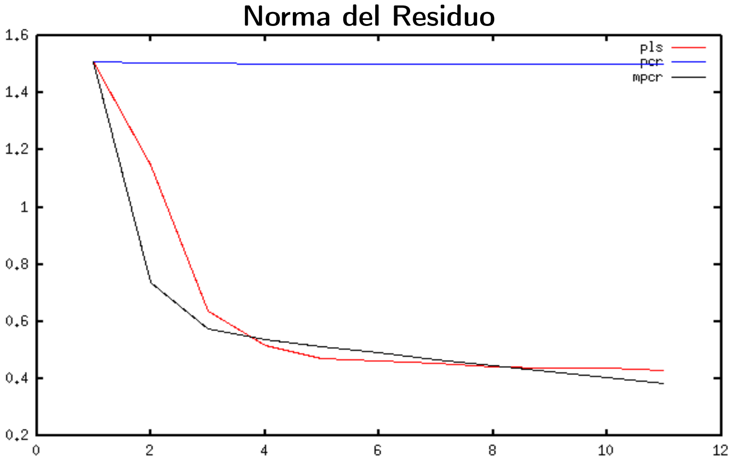


# $y$ Concentrata

- $y$ concentrata sull'ottavo valore singolare

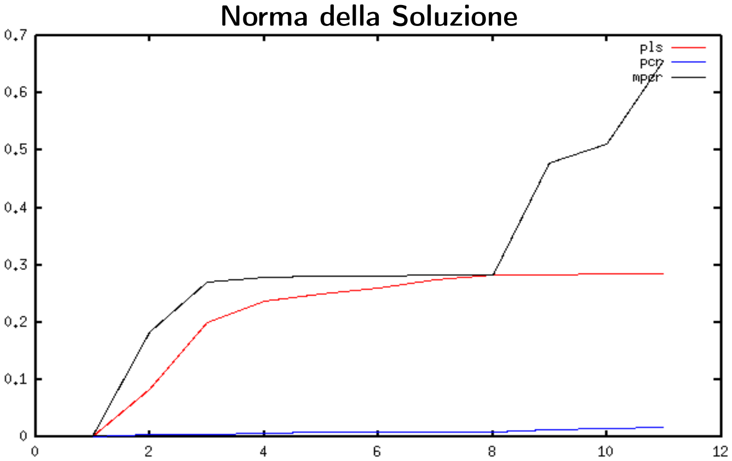
# $y$ Concentrata

- $y$ concentrata sull'ottavo valore singolare



# $y$ Concentrata

- $y$ concentrata sull'ottavo valore singolare

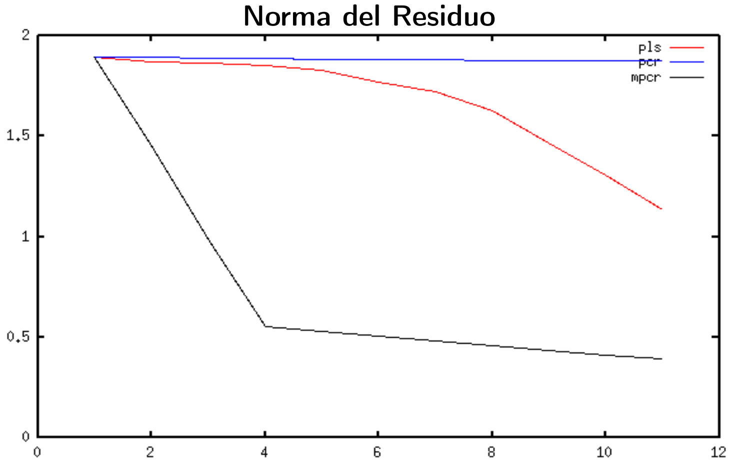


# $y$ Concentrata

- $y$ concentrata sul ventottesimo valore singolare

# $y$ Concentrata

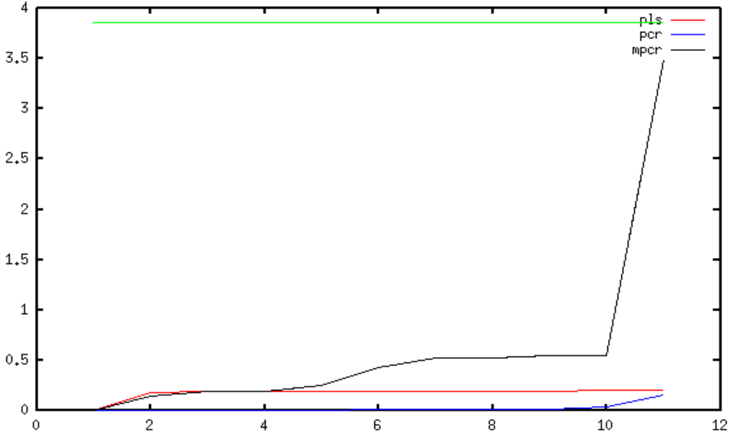
- $y$ concentrata sul ventottesimo valore singolare



# $y$ Concentrata

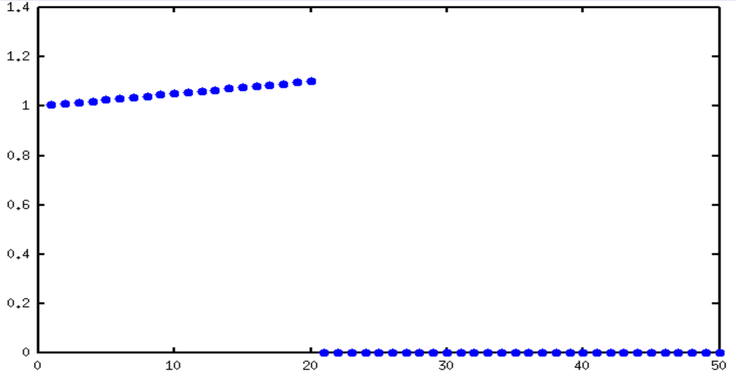
- $y$ concentrata sul ventottesimo valore singolare

**Norma della Soluzione**



# $A$ e $y$ sbilanciate

$A$ uniforme su 20 componenti, e le altre a $10^{-3}$ come mostrato sotto. $y$ con elementi uguali a quelli di $A$, e poi al contrario: uniforme sulle ultime 30, e piccolo sulle prime.

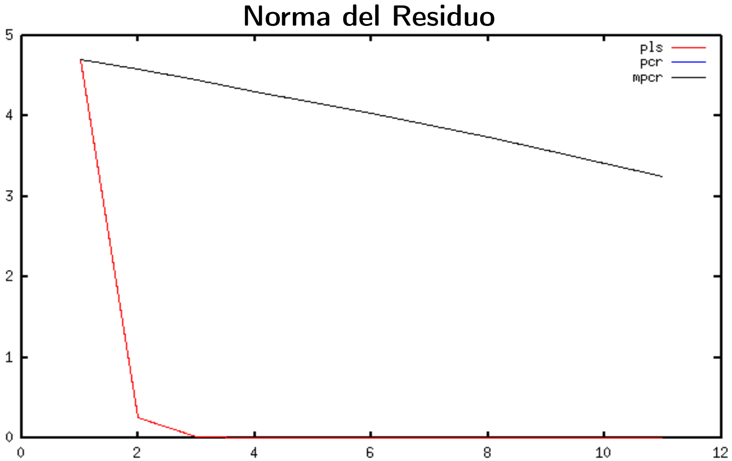


# $A$ e $y$ sbilanciate

- $y$ e $A$ sbilanciate uguali

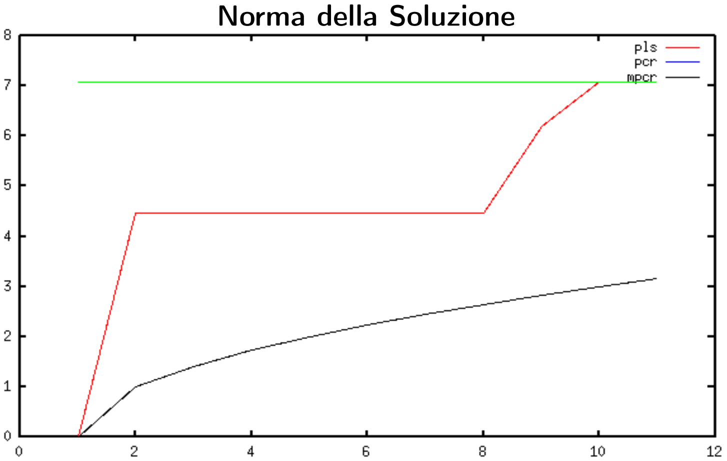
# $A$ e $y$ sbilanciate

- $y$ e $A$ sbilanciate uguali



# $A$ e $y$ sbilanciate

- $y$ e $A$ sbilanciate uguali

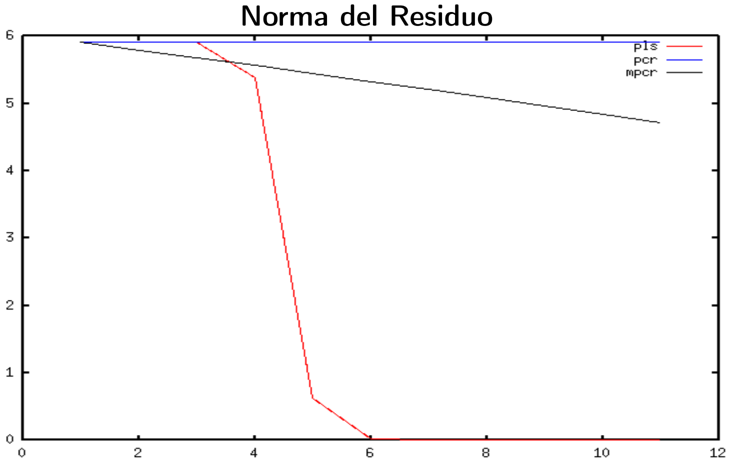


# $A$ e $y$ sbilanciate

- $y$ e $A$ sbilanciate in maniera opposta

# $A$ e $y$ sbilanciate

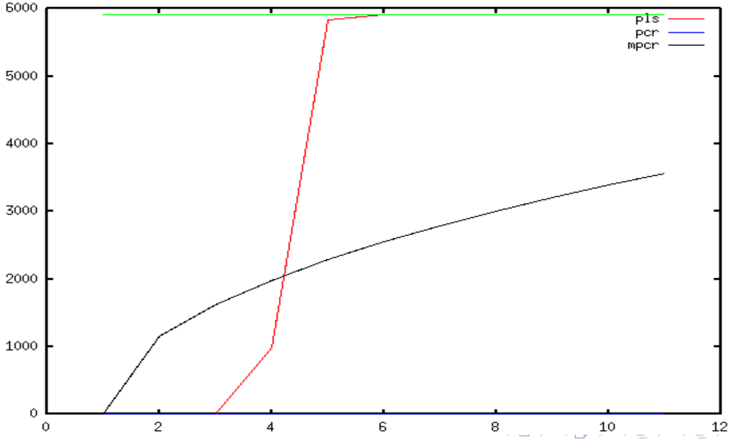
- $y$ e $A$ sbilanciate in maniera opposta



# $A$ e $y$ sbilanciate

- $y$ e $A$ sbilanciate in maniera opposta

Norma della Soluzione



# Conclusioni

- Il PLS è un algoritmo molto stabile, ma con un costo computazionale di $O(n^2)$ per passo, e il cui comportamento non è semplice da analizzare
- Il PLS batte il PCR al caso medio, sia in relazione al residuo che alla norma della soluzione generata
- Dato che un SVD costa $O(n^3)$ *(Golub & Van Loab)*, il PLS batte il PCR anche in tempo
- Bisogna comunque usare il PLS con cura, poiché su particolari matrici non siamo in grado di predirne l'andamento

## Conclusioni

- Il PLS è un algoritmo molto stabile, ma con un costo computazionale di $O(n^2)$ per passo, e il cui comportamento non è semplice da analizzare
- Il PLS batte il PCR al caso medio, sia in relazione al residuo che alla norma della soluzione generata
- Dato che un SVD costa $O(n^3)$ *(Golub & Van Loab)*, il PLS batte il PCR anche in tempo
- Bisogna comunque usare il PLS con cura, poiché su particolari matrici non siamo in grado di predirne l'andamento

# Conclusioni

- Il PLS è un algoritmo molto stabile, ma con un costo computazionale di $O(n^2)$ per passo, e il cui comportamento non è semplice da analizzare
- Il PLS batte il PCR al caso medio, sia in relazione al residuo che alla norma della soluzione generata
- Dato che un SVD costa $O(n^3)$ *(Golub & Van Loab)*, il PLS batte il PCR anche in tempo
- Bisogna comunque usare il PLS con cura, poiché su particolari matrici non siamo in grado di predirne l'andamento

# Conclusioni

- Il PLS è un algoritmo molto stabile, ma con un costo computazionale di $O(n^2)$ per passo, e il cui comportamento non è semplice da analizzare
- Il PLS batte il PCR al caso medio, sia in relazione al residuo che alla norma della soluzione generata
- Dato che un SVD costa $O(n^3)$ *(Golub & Van Loab)*, il PLS batte il PCR anche in tempo
- Bisogna comunque usare il PLS con cura, poiché su particolari matrici non siamo in grado di predirne l'andamento

# Conclusioni

- Il PLS è un algoritmo molto stabile, ma con un costo computazionale di $O(n^2)$ per passo, e il cui comportamento non è semplice da analizzare
- Il PLS batte il PCR al caso medio, sia in relazione al residuo che alla norma della soluzione generata
- Dato che un SVD costa $O(n^3)$ *(Golub & Van Loab)*, il PLS batte il PCR anche in tempo
- Bisogna comunque usare il PLS con cura, poiché su particolari matrici non siamo in grado di predirne l'andamento