Sisteme și algoritmi distribuiți
Curs 10

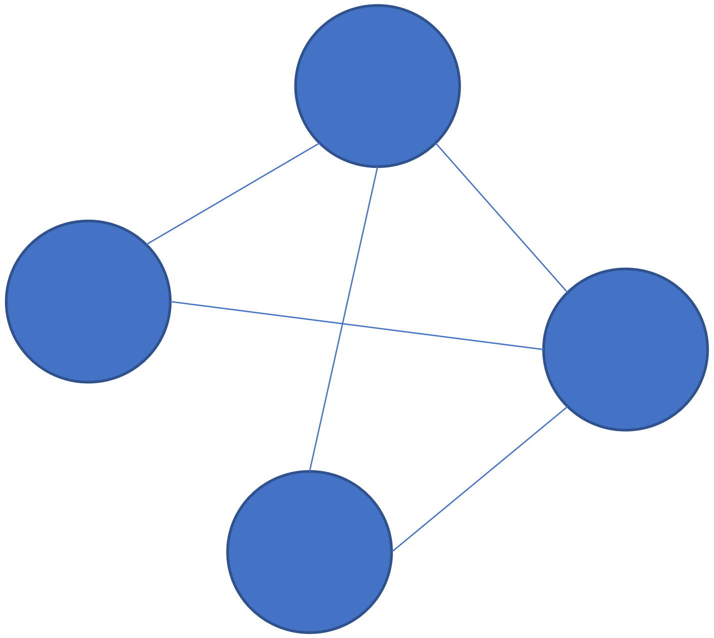
Cuprins

- Algoritmi Gossip (partea a II-a)
- Sisteme liniare. Introducere în problema intersecției de mulțimi convexe.
- Algoritmi distribuiți de consens pentru sisteme liniare.

Un model simplu

Schema Gossip de bază (pe perechi coordonate):

1. Algoritmul utilizează informația primită de la vecini $N(i)$
2. Realizează o operație de actualizare (simplă) per unitate de timp
3. **Nodul i selectează aleator un vecin și trimite actualizarea** (schema *push*)
4. Eventual, nodul i realizează o operație *pull*.



În problema de medie operația de actualizare se reduce la:

$$x_i(t+1) = \frac{x_i(t) + x_j(t)}{2}$$

Pentru push-pull, adițional:

$$x_j(t+1) = \frac{x_i(t) + x_j(t)}{2}$$

Ipoteze:

- Vecinul de intrare j al nodului i este selectat cu probabilitatea P_{ij}
- $P_{ij} > 0$ dacă $(i, j) \in E$ și $\sum_{j \in N(i)} P_{ij} = 1$

Algoritm gossip de medie

Fiecare nod $i \in V$ stochează valoarea $x_i(0)$. Notăm media $\mu = \frac{1}{n}\sum_i x_i(0)$.

Algoritm gossip *push-pull*: alege aleator (i, j) o pereche din E

$$x_i(t+1) = \frac{x_i(t) + x_j(t)}{2}, \quad x_j(t+1) = \frac{x_i(t) + x_j(t)}{2}.$$

Sau echivalent,

$$x(t+1) = W(t)x(t)$$

unde (e_i coloana i a matricii identitate)

$$W(t) = W_{ij} = I - \frac{1}{2}\left(e_i - e_j\right)\left(e_i - e_j\right)^T.$$

Matricea ponderilor este dinamică și aleatoare.

Algoritm gossip de medie

$$x(t+1) = W(t)x(t), \qquad W(t) = W_{ij} = I - \frac{1}{2}(e_i - e_j)(e_i - e_j)^T$$

Exemplu: $n = 4, i = 2, j = 3$

$$W_{23} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matricile W_{ij} sunt dublu stohastice, i.e. $W_{ij}\mathbf{1} = \mathbf{1}, \mathbf{1}^T W_{ij} = \mathbf{1}^{\boldsymbol{T}}$. Mai mult: $W(t)^T W(t) = W(t)$.

Observăm că: $x(t)$ este un proces stohastic în care
- $x(1)$ depinde de alegerea (i_0, j_0)
- $x(2)$ depinde de alegerea $\mathrm{H}(2) = \{(i_0, j_0), (i_1, j_1)\}$
- ...
- $x(t)$ depinde de alegerea $\mathrm{H(t)} = \{(i_0, j_0), (i_1, j_1), \cdots, (i_{t-1}, j_{t-1})\}$

⇒ Urmărim convergența lui $x(t)$ în medie, i.e. $\mathrm{E}\left[\sigma^2\big(x(t)\big) \middle| x(0)\right] \leq \epsilon$.

Algoritm gossip de medie

Notăm $y(t) = x(t) - \mu\mathbf{1}$, atunci varianța: $\sigma^2\big(x(t)\big) = \big||y(t)|\big|^2$ (conservarea masei).

Observăm că

$$y(t) \perp \mathbf{1} \quad (i.e.\ y(t)^T\mathbf{1} = 0)$$

și atunci

$$y(t+1) = W(t)y(t).$$

Algoritm gossip de medie

Pentru a mărgini media varianței avem:

$$\begin{aligned} E[y(t+1)^T y(t+1)|x(0)] &= E[y(t)^T W(t)^T W(t) y(t)|x(0)] \\ &= E[y(t)^T W(t) y(t)|x(0)] \\ &= E\big[\, E_{ij}[y(t)^T W(t) y(t)] \,\big|\, x(t-1), .., x(0)\big] \\ &= E\big[y(t)^T E_{ij}[W(t)] y(t) \big|\, x(t-1), .., x(0)\big] \end{aligned}$$

unde $W = E_{ij}[W(t)] = I - \frac{1}{2n} D + \frac{P+P^T}{2n}$, $D_i = \sum_j P_{ij} + P_{ji}$.

$$E[y(t+1)^T y(t+1)|x(0)] = E[y(t)^T W y(t)|\, x(0)]$$

Matricea W este dublu stohastică. Folosind Perron-Frobenius, observăm că **1** este vector propriu maximal al W și deci $y(t)$ este ortogonal pe subspațiul generat de acest v. p. ($y(t)$ se află în subspațiul generat de restul de $n-1$ v.p. ai matrici W). Această proprietate indică

$$y(t)^T W y(t) \leq \lambda_2(W) y(t)^T y(t) = \lambda_2(W) \sigma^2\big(x(t)\big).$$

Algoritm gossip de medie

Varianța convergen liniar cu factorul $\lambda_2(W)$:

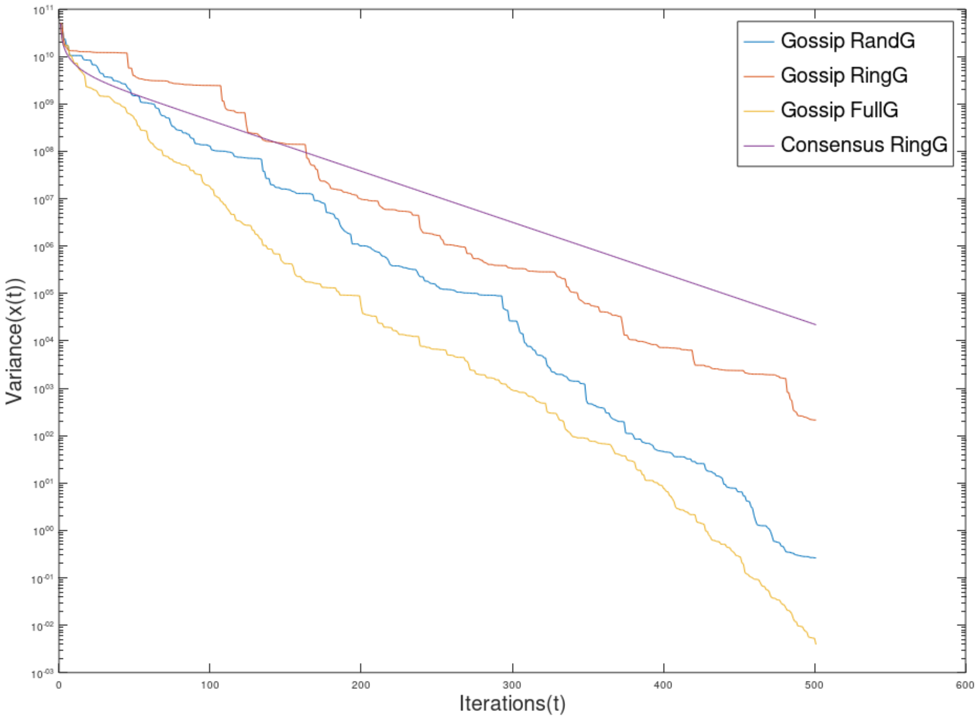
$$E\left[\sigma^2\big(x(t+1)\big)\middle|x(0)\right] \le \lambda_2(W) E\left[\sigma^2\big(x(t)\big)\middle|\, x(0)\right]$$
$$\le \lambda_2(W)^t \sigma^2\big(x(0)\big)$$

Pentru o acuratețe fixată ϵ și P dat, după $O\left(\frac{3log(\epsilon^{-1})}{log(\lambda_2(W)^{-1})}\right)$ iterații este garantat: $E\left[\sigma^2\big(x(t)\big)\right] \le \epsilon$.

Exemplu:

$$n = 20$$

Inel: $\lambda_2(W)$ = 0.9988
Random: $\lambda_2(W)$ = 0.9636
Full: $\lambda_2(W)$ = 0.9500



Algoritm gossip de medie

Comparatie cu Algoritmul Flooding (vezi Consens)

Exemplu:

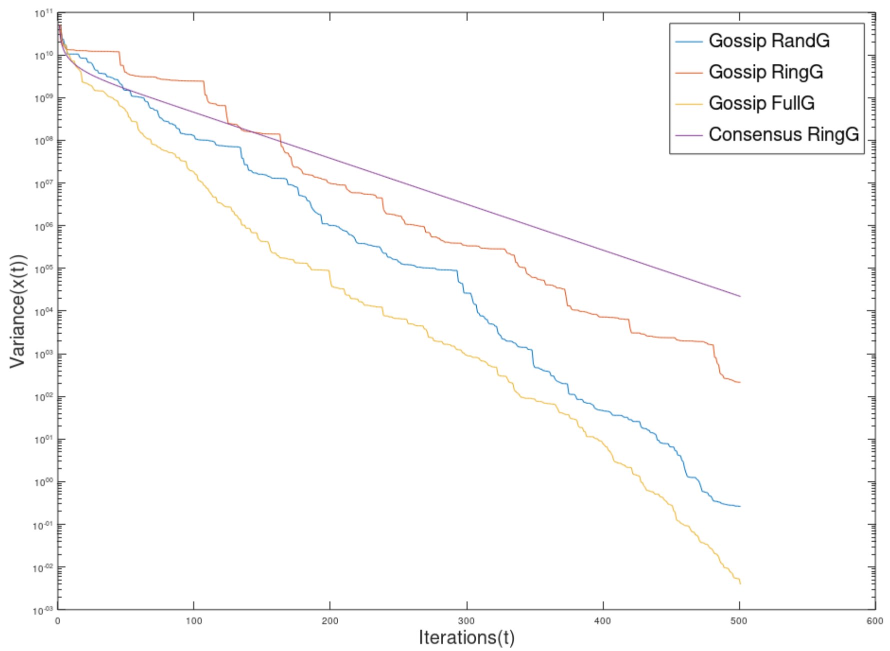
$$n = 20$$

Inel: $\lambda_2(W)$ = 0.9988

Random: $\lambda_2(W)$= 0.9636 (p = 0.8)

Full: $\lambda_2(W)$ = 0.9500

Vs. Alg. Flooding (inel)



Algoritm gossip de medie

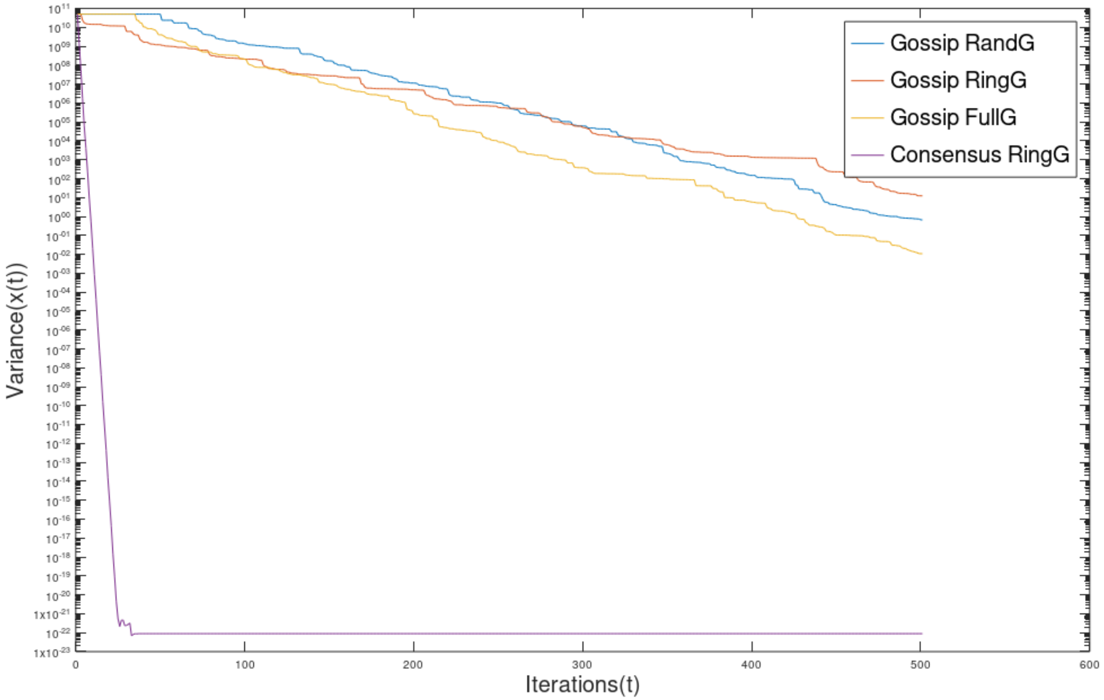
Comparatie cu Algoritmul Flooding

Exemplu:

$$n = 20$$

Inel: $\lambda_2(W)$ = 0.9988
Random: $\lambda_2(W)$= 0.9636 (p = 0.8)
Full: $\lambda_2(W)$ = 0.9500

Vs. Alg. Flooding (graf random, p = 0.8)



Sisteme liniare

Sisteme liniare

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

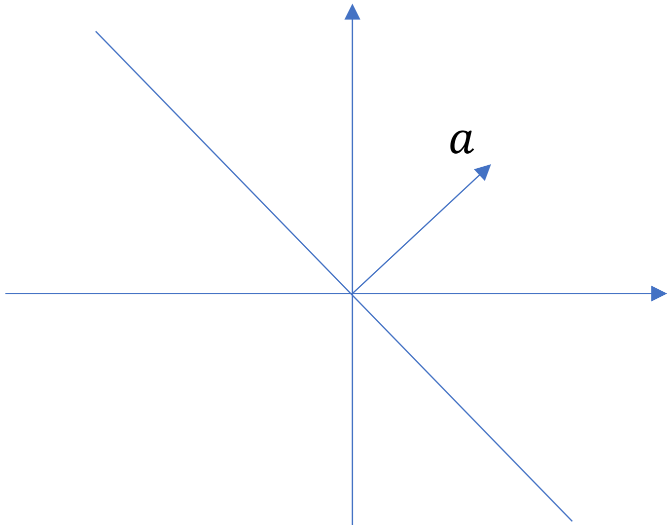
$$Ax = b$$

Hiperplan

În R^n, un hiperplan reprezintă un subspațiu liniar de dimensiune $n-1$ parametrizat de $a \in R^n, b \in R$
$H = \{x \in R^n : a^T x = b\}$.

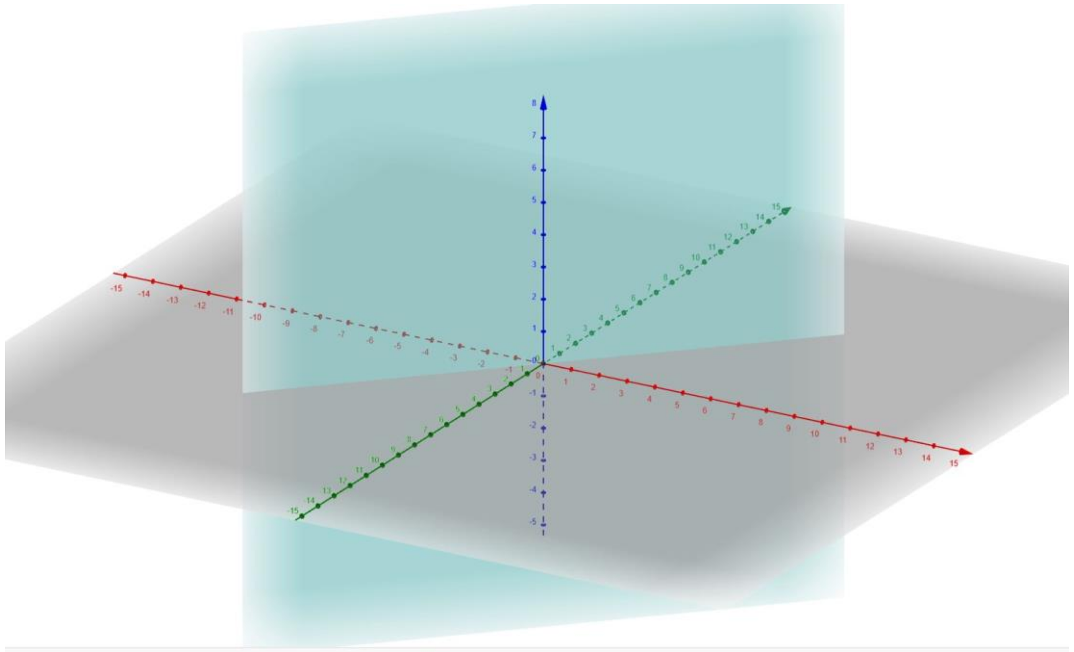
Exemple:

- în R^2 : $H = \{x \in R^2 : x_1 + x_2 = 0\}$



Exemplu R^3:

$$H = \{x \in R^3 : x_1 - x_2 = 0\}$$



Sistem liniar subdeterminat

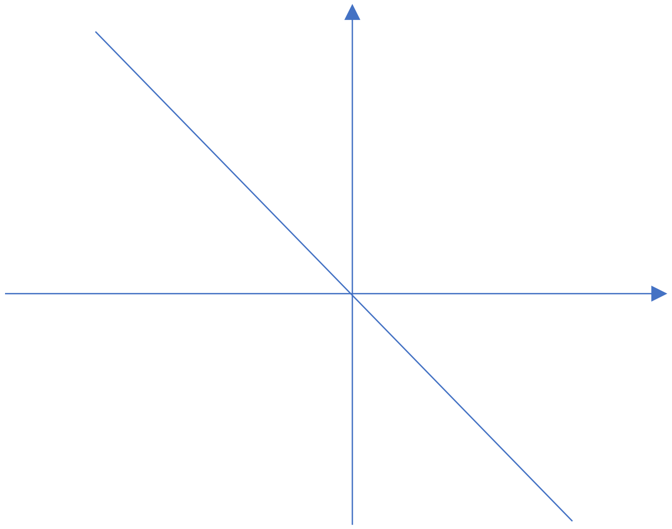
Un sistem liniar se numește subdeterminat

$$Ax = b, \qquad A \in R^{m \times n}, b \in R^m$$

dacă $m < n$.

De exemplu, determinarea unui punct din H, presupune rezolvare unui sistem subdeterminat. În acest caz, întregul H este spațiul soluțiilor.

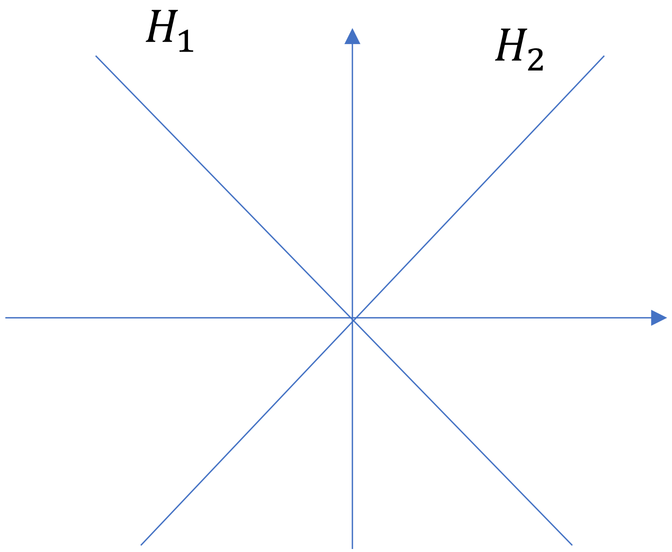
$H = \{x \in R^2 : x_1 + x_2 = 0\}$



Sistem liniar determinat

Un sistem liniar se numește determinat $Ax = b, \quad A \in R^{m \times n}, b \in R^m$, dacă $m = n$.

De exemplu, avem sistemul cu două ecuații: $\begin{cases} x_1 + x_2 = 0 \\ x_1 - x_2 = 0 \end{cases}$.



Fiecare dintre ecuații se exprimă printr-un hiperplan:

$$H_1 = \{x \in R^2 : x_1 + x_2 = 0\}$$
$$H_2 = \{x \in R^2 : x_1 - x_2 = 0\}$$

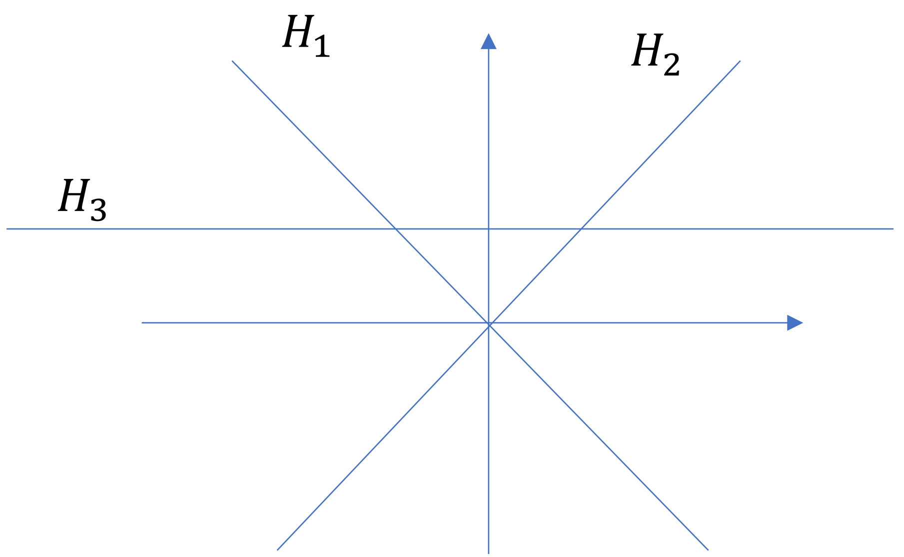
Observăm că

$$\{0\} = H_1 \cap H_2.$$

Sistem liniar supradeterminat

Un sistem liniar se numește supradeterminat $Ax = b, A \in R^{m \times n}, b \in R^m$ dacă $m > n$.
De exemplu, adăugând o ecuație sistemului precedent, obținem unul supradeterminat:

$$\begin{cases} x_1 + x_2 = 0 \\ x_1 - x_2 = 0 \\ \quad x_2 = 1 \end{cases}.$$



Sistemul este echivalent cu intersecția $\mathrm{H}_1 \cap \mathrm{H}_2 \cap \mathrm{H}_3$, unde

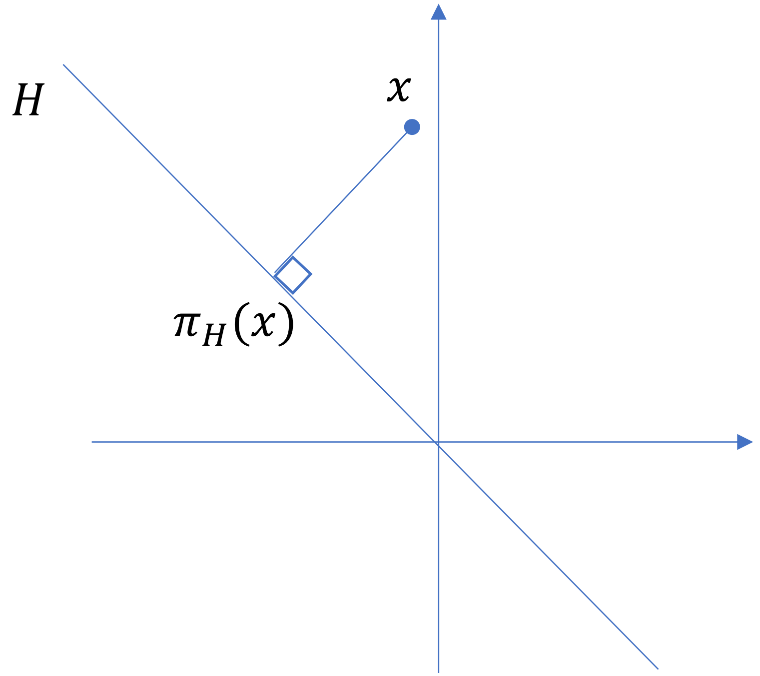
$$H_1 = \{x \in R^2 : x_1 + x_2 = 0\}$$
$$H_2 = \{x \in R^2 : x_1 - x_2 = 0\}$$
$$H_3 = \{x \in R^2 : x_2 = 1\}.$$

Sistemul nu are soluție

$$\{\emptyset\} = H_1 \cap H_2 \cap H_3.$$

Proiecție ortogonală

În R^n, proiecția ortogonală a punctului x pe hiperplanul $H = \{x \in R^n : a^T x = b\}$ este punctul $\pi_H(x)$ din H cel mai „apropiat” (în norma euclidiană) de x.

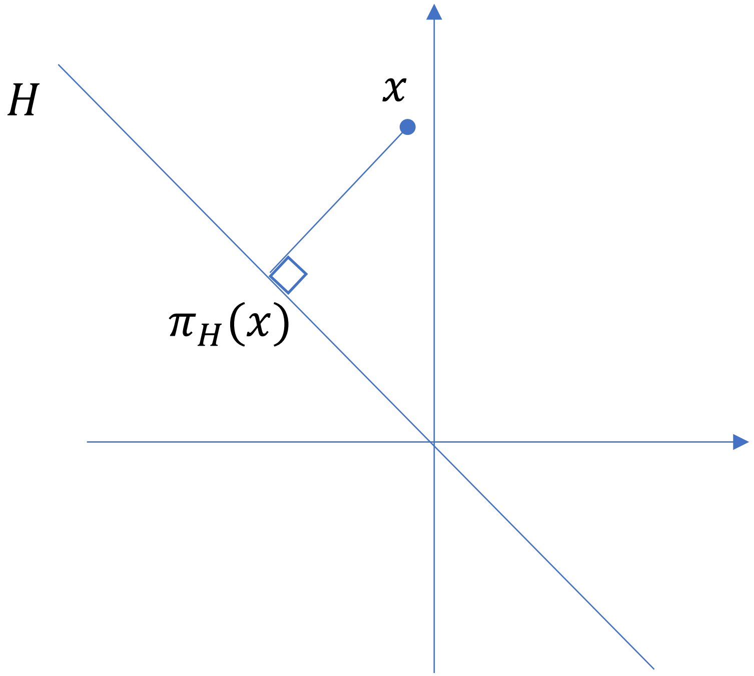


Proprietăți:

- $\|\pi_H(x) - x\| \leq \|z - x\|, \ \forall z \in H$
- $\pi_H(x) = x, \ \forall x \in H$
- $\pi_H(x)$ unică (H mulțime convexă)
- Formă explicită: $\pi_H(x) = x - \frac{a^T x - b}{\|a\|^2} a$

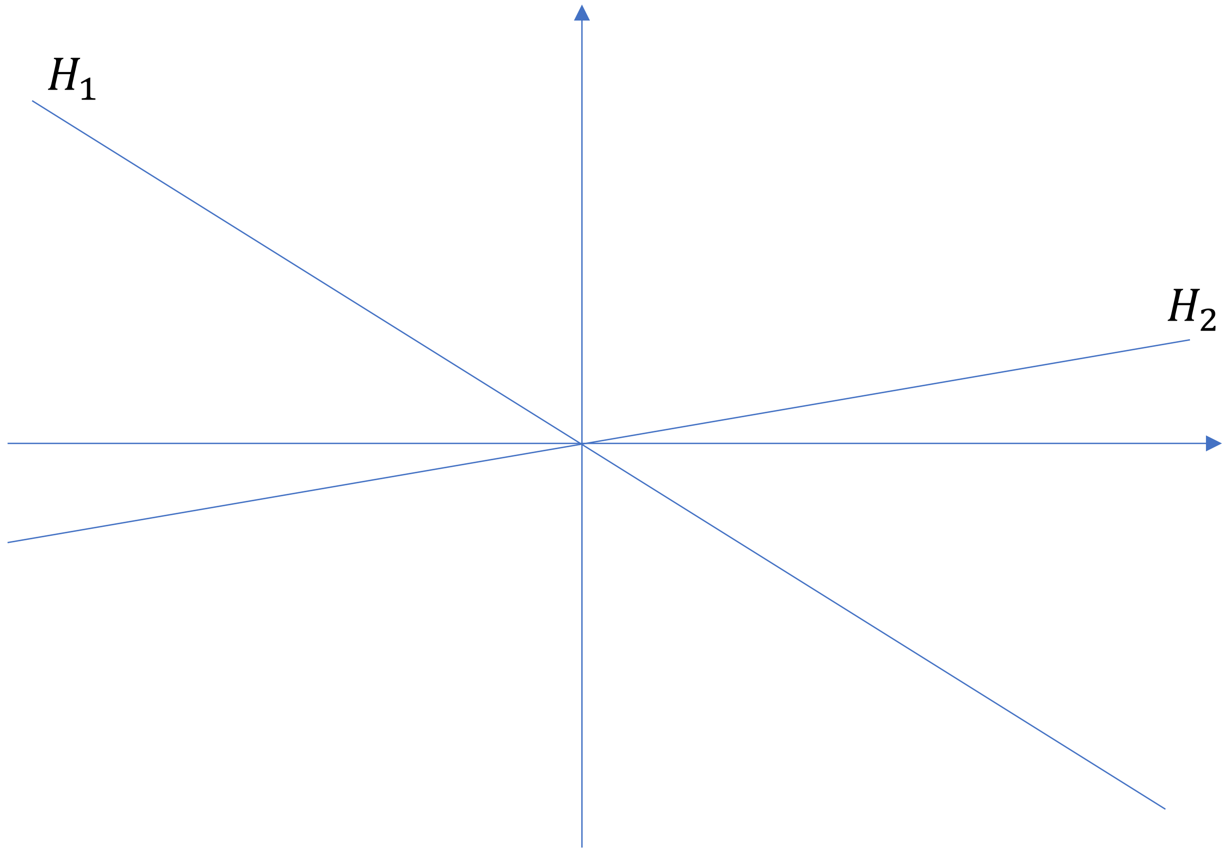
Proiecție ortogonală

Fie $H = \{x \in R^2 : x_1 + x_2 = 0\}$ și $z = \begin{bmatrix} 3 \\ 2 \end{bmatrix}, u = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$. Calculați $\pi_H(z), \pi_H(u)$și distanțele până la H.



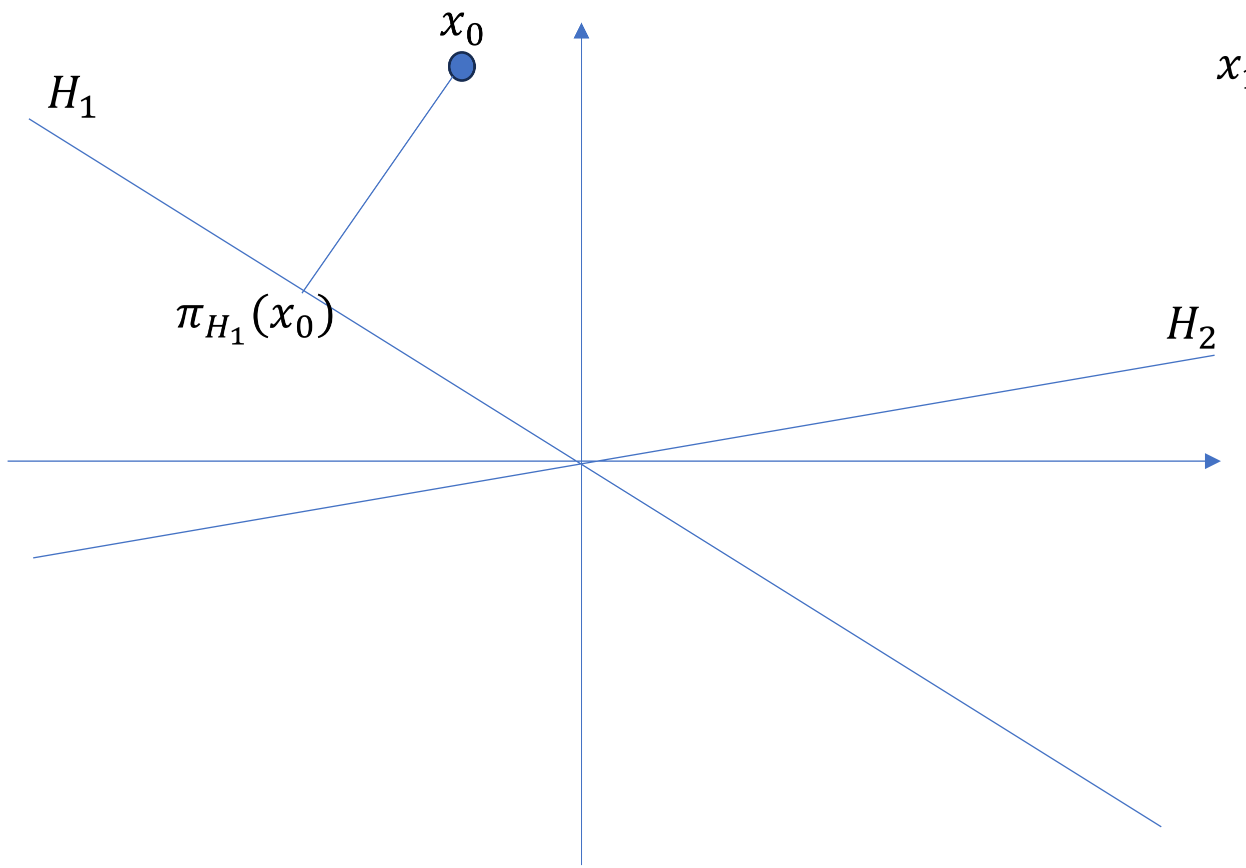
Algoritmul Proiecțiilor Alternative

Problemă: Fie $a_1, a_2 \in R^n, a_1 \neq a_2$. Rezolvați sistemul $\begin{cases} a_1^T x = b_1 \\ a_2^T x = b_2 \end{cases}$ folosind doar proiecții ortogonale.



Algoritmul Proiecțiilor Alternative

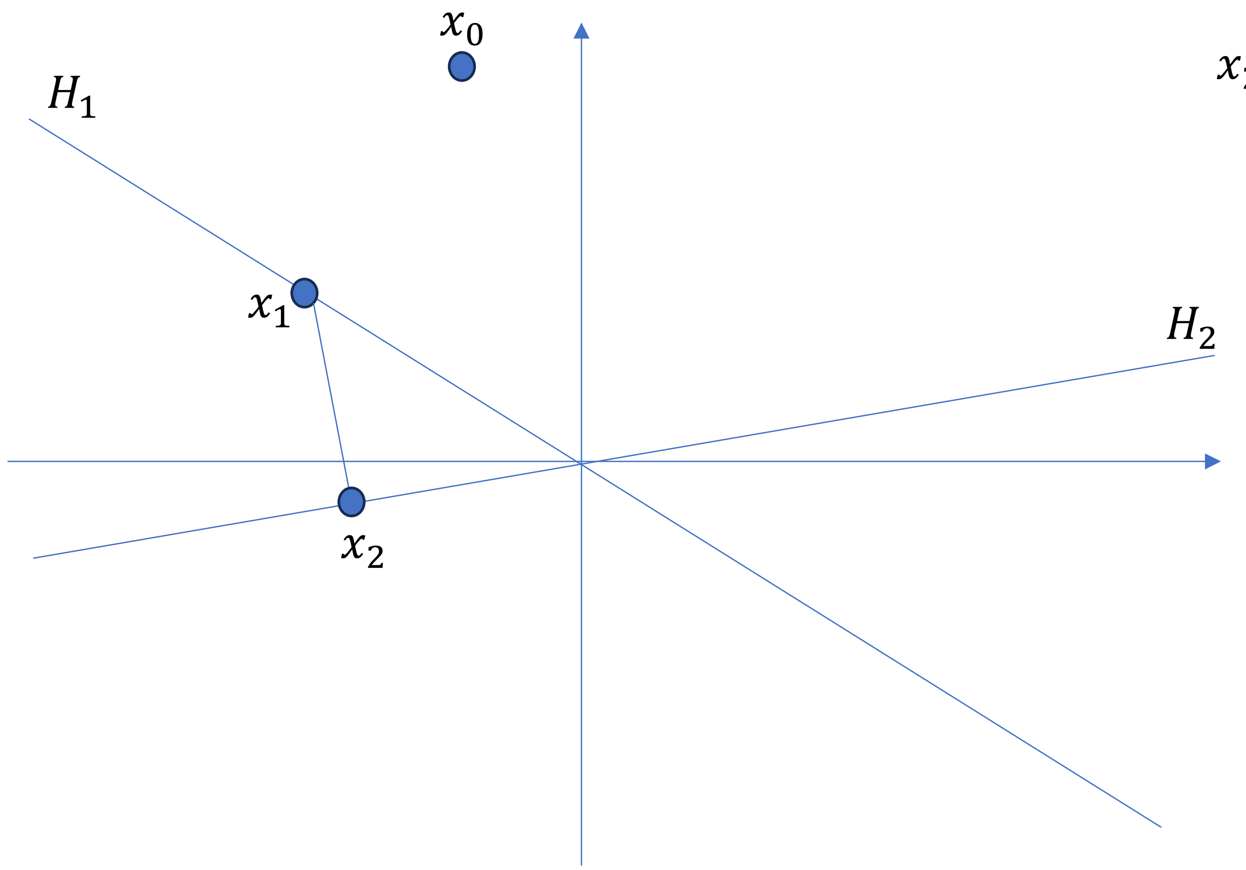
Alegem un x_0 arbitrar, alegem un hiperplan (H_1 sau H_2) și realizăm șirul iterativ $x_{t+1} = \pi_{H_{i_t}}(x_t)$.



$$x_1 = x_0 - \frac{a_{(1)}^T x_0 - b_1}{\left\|a_{(1)}\right\|^2} a_{(1)}.$$

Algoritmul Proiecțiilor Alternative

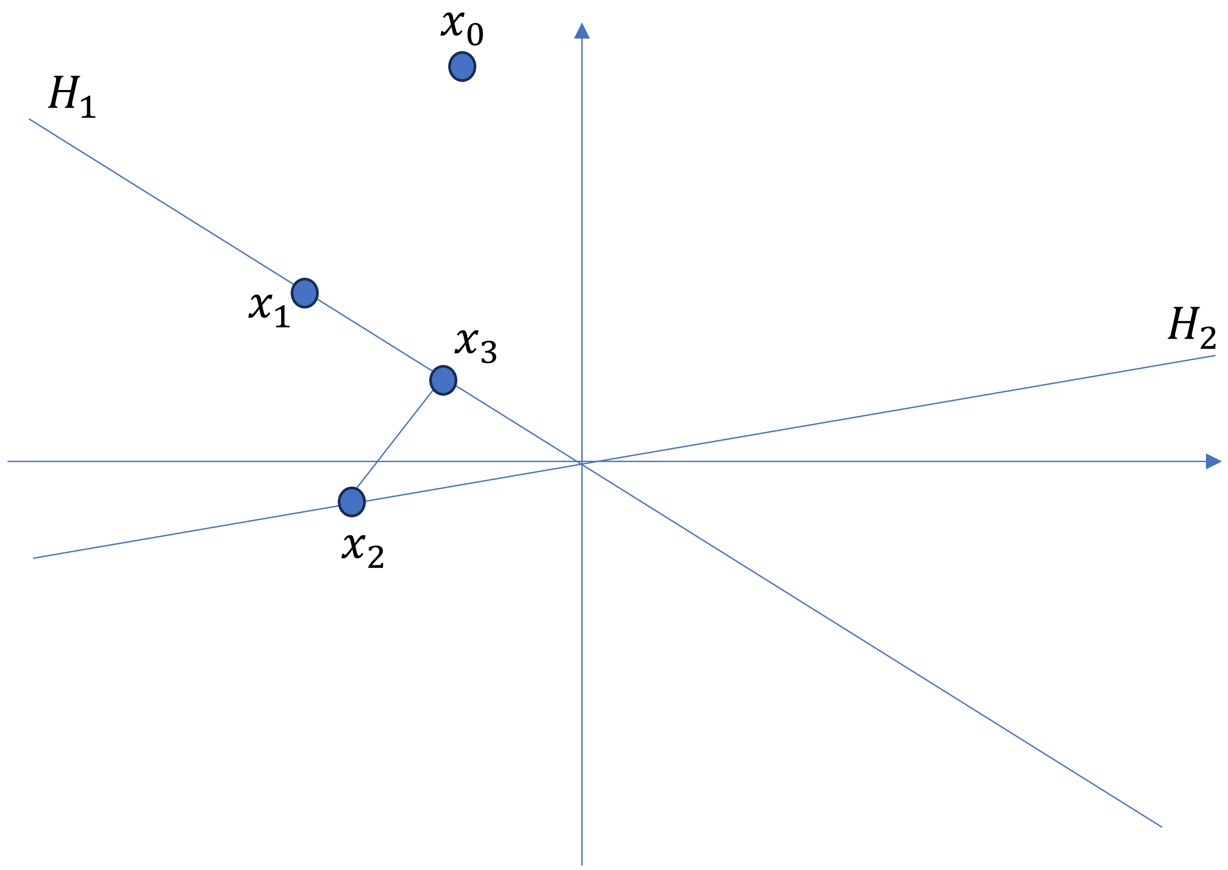
Alegem un x_0 arbitrar, alegem un hiperplan (H_1 sau H_2) și realizăm șirul iterativ $x_{t+1} = \pi_{H_{i_t}}(x_t)$.



$$x_2 = x_1 - \frac{a_{(2)}^T x_1 - b_2}{\left|\left|a_{(2)}\right|\right|^2} a_{(2)}.$$

Algoritmul Proiecțiilor Alternative

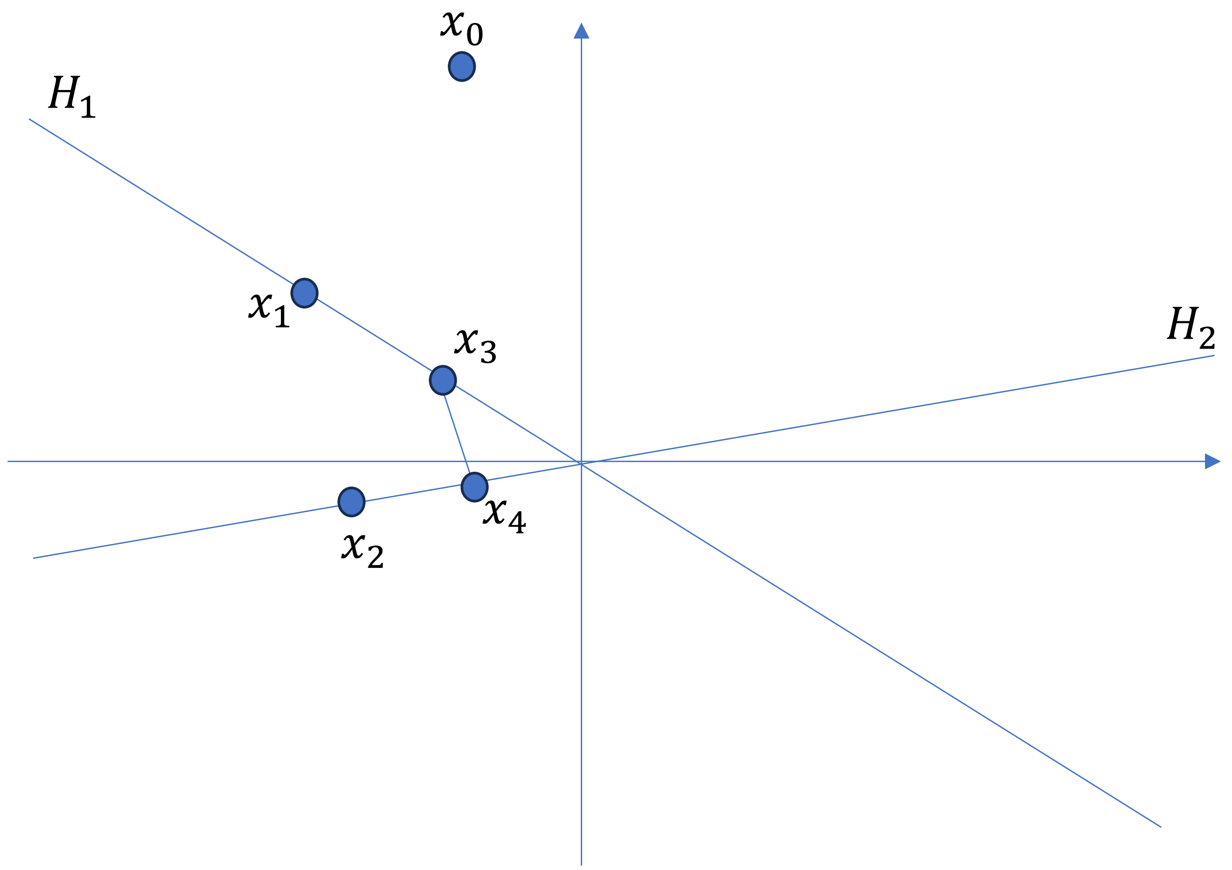
Alegem un x_0 arbitrar, alegem un hiperplan (H_1 sau H_2) și realizăm șirul iterativ $x_{t+1} = \pi_{H_{i_t}}(x_t)$.



$$x_3 = x_2 - \frac{a_{(1)}^T x_2 - b_1}{\left\|a_{(1)}\right\|^2} a_{(1)}.$$

Algoritmul Proiecțiilor Alternative

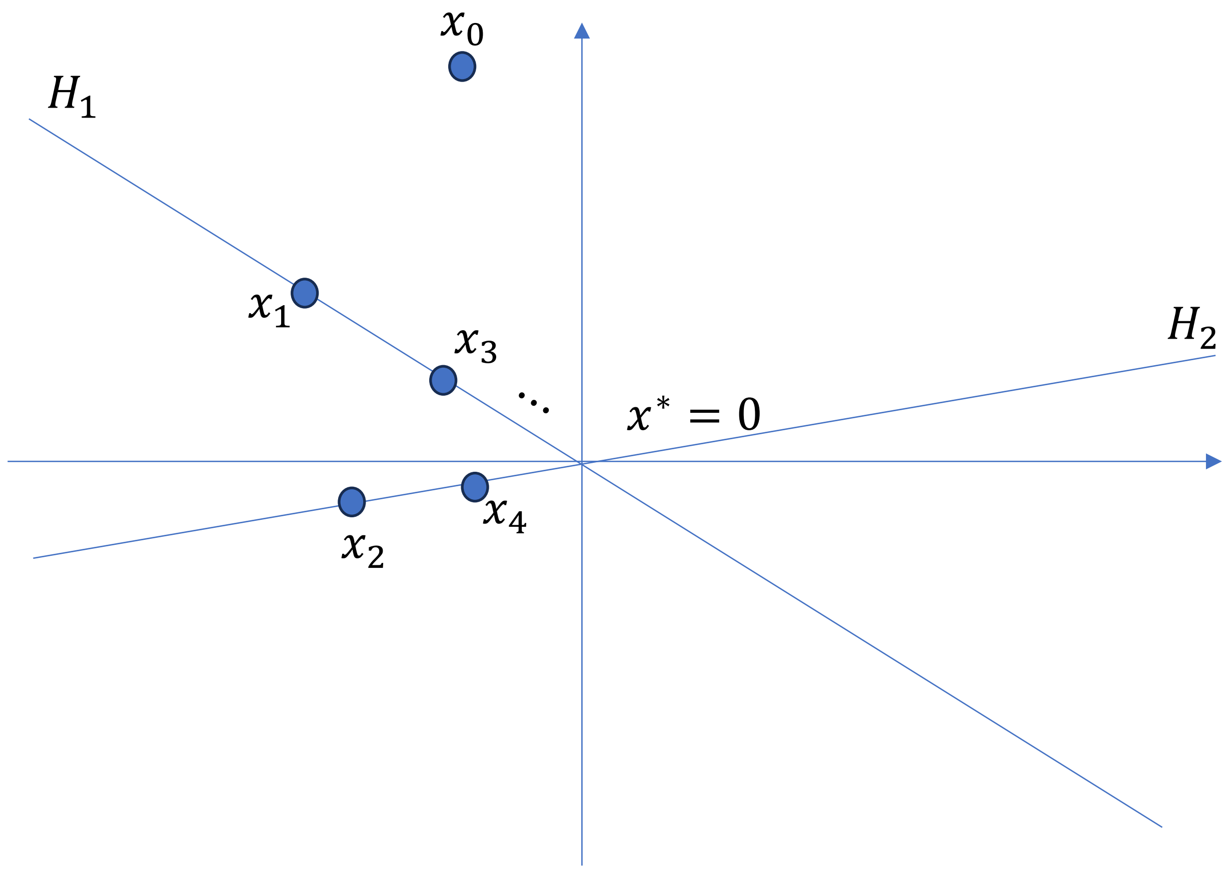
Alegem un x_0 arbitrar, alegem un hiperplan (H_1 sau H_2) și realizăm șirul iterativ $x_{t+1} = \pi_{H_{i_t}}(x_t)$.



$$x_4 = x_3 - \frac{a_{(2)}^T x_3 - b_2}{\left\|a_{(2)}\right\|^2} a_{(2)}.$$

Algoritmul Proiecțiilor Alternative

Alegem un x_0 arbitrar, alegem un hiperplan (H_1 sau H_2) și realizăm șirul iterativ $x_{t+1} = \pi_{H_{i_t}}(x_t)$.



Algoritmul Proiecțiilor Alternative

Convergenta pt 2 hiperplane:

$$x_{t+1} = \pi_{H_1}(x_t) = \pi_{H_1}\left(\pi_{H_2}(x_{t-1})\right)$$

Observăm: $\mathrm{x}_{t+1}, x_{t-1} \in H_1$ și $\mathrm{x}_t, x_{t-2} \in H_2$, de aceea $a_{(1)}^T x_{t-1} = b_1$.

Reziduul $||Ax_t - b||_2$ măsoară distanța la $H_1 \cap H_2$, deci

$$a_{(2)}^T x_{t+1} - b_2 = \left(a_{(2)}^T x_{t-1} - b_2\right) \frac{\left(a_{(1)}^T a_{(2)}\right)^2}{||a_{(1)}||^2\,||a_{(2)}||^2} - \left(a_{(1)}^T x_{t-1} - b_1\right) \frac{a_{(1)}^T a_{(2)}}{||a_{(1)}||^2}$$

$$a_{(2)}^T x_{t+1} - b_2 = \left(a_{(2)}^T x_{t-1} - b_2\right) \frac{\left(a_{(1)}^T a_{(2)}\right)^2}{||a_{(1)}||^2\,||a_{(2)}||^2}$$

$$||Ax_{t+1} - b||_2 = \frac{\left(a_{(1)}^T a_{(2)}\right)^2}{||a_{(1)}||^2\,||a_{(2)}||^2} ||Ax_{t-1} - b||_2$$

Factorul de convergență $\frac{|a_{(1)}^T a_{(2)}|}{||a_{(1)}||\,||a_{(2)}||}$ = cosinusul unghiului dintre H_1 și H_2.

Algoritmul Proiecțiilor Alternative

Rezolvarea sistemului

$$Ax = b, \qquad A \in R^{m \times n}, b \in R^m$$

este echivalentă cu determinarea

$$x \in H_1 \cap H_2 \cap \cdots \cap H_m,$$

unde $H_i = \left\{x: a_{(i)}^T x = b_i\right\}$. Schema generală APA selectează la iterația t un hiperplan $H_{i(t)}$ și actualizează iterația:

$$x_{t+1} \coloneqq \pi_{H_{i(t)}}(x_t)$$

Alegerea $H_{i(t)}$ poate urma regula *ciclică*, *aleatoare*, *„greedy”* etc.

Algoritmul Proiecțiilor Alternative

Formatul distribuit al problemei găsirii $x \in H_1 \cap H_2 \cap \cdots \cap H_m$ presupune determinarea z $\in H_i = \left\{x : a_{(i)}^T x = b_i\right\}$ o sarcină individuală asociată nodului P_i.

Deci presupunem că P_i poate rezolva propria sarcină locală (i.e. determinarea unui punct din H_i, respectiv proiecția ortogonală pe H_i).

Starea $x_i(t) \in R^n$ reprezintă estimarea locală a soluției la pasul t. Dacă starea nodului P_i se află pe hiperplanul H_i, i.e. $x_i \in H_i$, atunci fiecare P_i va satisface ecuația $a_{(i)}^T x_i = b_i$, dar nu avem o soluție a întregului sistem.

Algoritmul de Consens Proiectat (ACP)

Algoritmul de Consens Proiectat compune pasul de consens cu cel de proiecție ortogonală:

$$x_i(t+1) = \pi_{H_i}\left(\sum_{j \in \mathcal{N}_i^- \cup i} w_{ij} x_j(t)\right) \qquad \forall i,$$

unde ponderile $w_{ij} \geq 0, \sum_{j \in \mathcal{N}_i^- \cup i} w_{ij} = 1$ (medie).

Dacă $H_i = R^n$, atunci ACP se reduce la Algoritmul Flooding de medie (din cursul 6):

$$x_i(t+1) = w_{ii} x_i(t) + \sum_{j \in \mathcal{N}_i^-} w_{ij} x_j(t) \qquad \forall i.$$

Algoritmul de Consens Proiectat (ACP)

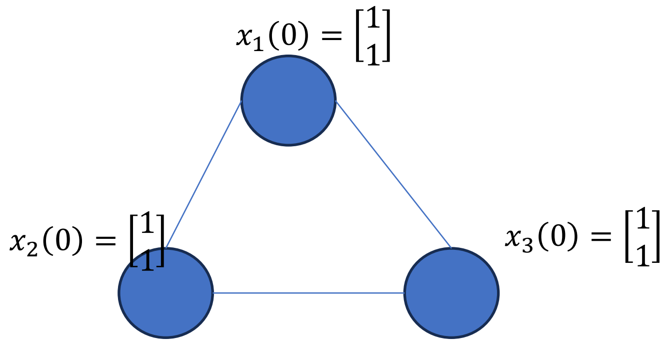
Algoritmul de Consens Proiectat compune pasul de consens cu cel de proiecție ortogonală:

$$x_i(t+1) = \pi_{H_i}\left(\sum_{j\,\in \mathcal{N}_i^- \cup i} w_{ij} x_j(t)\right) \qquad \forall i.$$

- La fiecare iterație P_i își menține starea pe H_i (păstrează activă ecuația $a_{(i)}^T x = b_i$) prin operația de proiecție.
- P_i se apropie de consens prin operația de medie $w_i^T x(t)$
- Este suficient consensul pentru a garanta rezolvarea sistemului.

Algoritmul de Consens Proiectat (ACP)

- Considerăm $n = 3, w_{ij} = \frac{1}{3}$ (uniforme)
- Rezolvăm sistemul: $\begin{cases} x_1 + x_2 = 0 \\ x_1 - x_2 = 0 \\ 2x_1 + 3x_2 = 0 \end{cases}$, pornind din $x_i(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, i = 1,2,3.$



Algoritmul de Consens Proiectat (ACP)

$$x_1(1) = \pi_{H_1}\left(\frac{1}{3}x_1(0) + \frac{1}{3}x_2(0) + \frac{1}{3}x_3(0)\right)$$

$$x_1(1) = \pi_{H_1}\left(\begin{bmatrix}1\\1\end{bmatrix}\right)$$

$$= \begin{bmatrix}1\\1\end{bmatrix} - \begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}$$

$$x_2(1) = \pi_{H_2}\left(\frac{1}{3}x_1(0) + \frac{1}{3}x_2(0) + \frac{1}{3}x_3(0)\right)$$

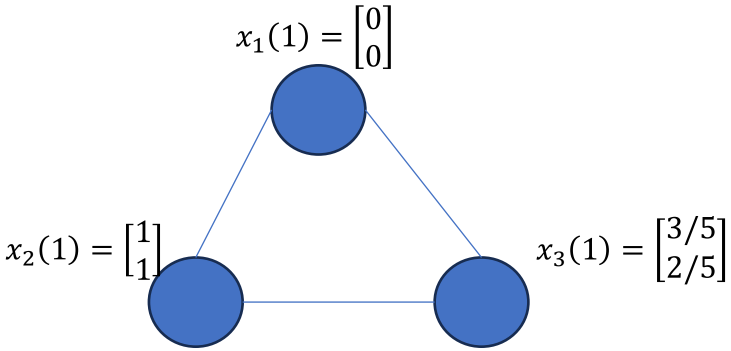
$$x_2(1) = \pi_{H_2}\left(\begin{bmatrix}1\\1\end{bmatrix}\right)$$

$$= \begin{bmatrix}1\\1\end{bmatrix}$$

$$x_3(1) = \pi_{H_3}\left(\frac{1}{3}x_1(0) + \frac{1}{3}x_2(0) + \frac{1}{3}x_3(0)\right)$$

$$x_3(1) = \pi_{H_3}\left(\begin{bmatrix}1\\1\end{bmatrix}\right)$$

$$= \begin{bmatrix}1\\1\end{bmatrix} - \frac{5}{25}\begin{bmatrix}2\\3\end{bmatrix} = \begin{bmatrix}3/5\\2/5\end{bmatrix}$$



Algoritmul de Consens Proiectat (ACP)

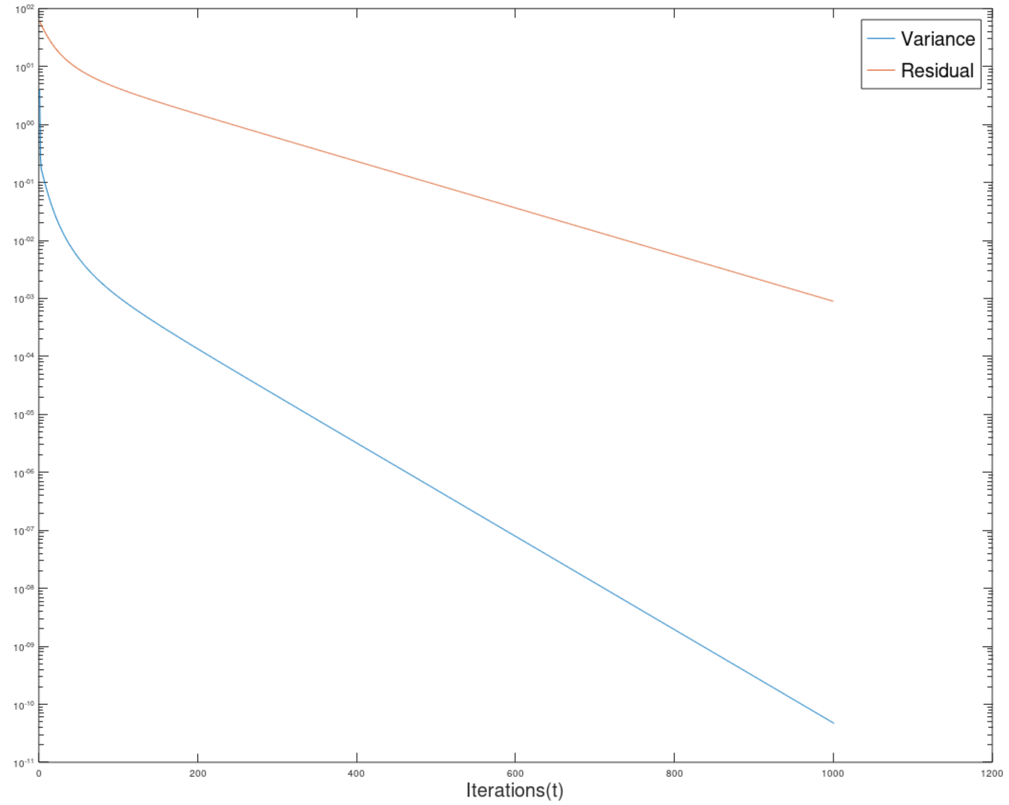
Algoritmul de Consens Proiectat compune pasul de consens cu cel de proiecție ortogonală:

$$x_i(t+1) = \pi_{H_i}\left(\sum_{j\,\in\mathcal{N}_i^{-}\cup i} w_{ij}x_j(t)\right) \qquad \forall i$$

Teorema. Fie matricea ponderilor W dublu stohastică. Presupunem că există constanta $\eta > 0$ astfel încât toate ponderile $w_{ij} > 0$ satisfac $w_{ij} \geq \eta$ ($w_{ii} \geq \eta$). Dacă sistemul $Ax = b$ are soluție, atunci șirul generat de ACP atinge consensul asimptotic (într-una dintre soluțiile sistemului).

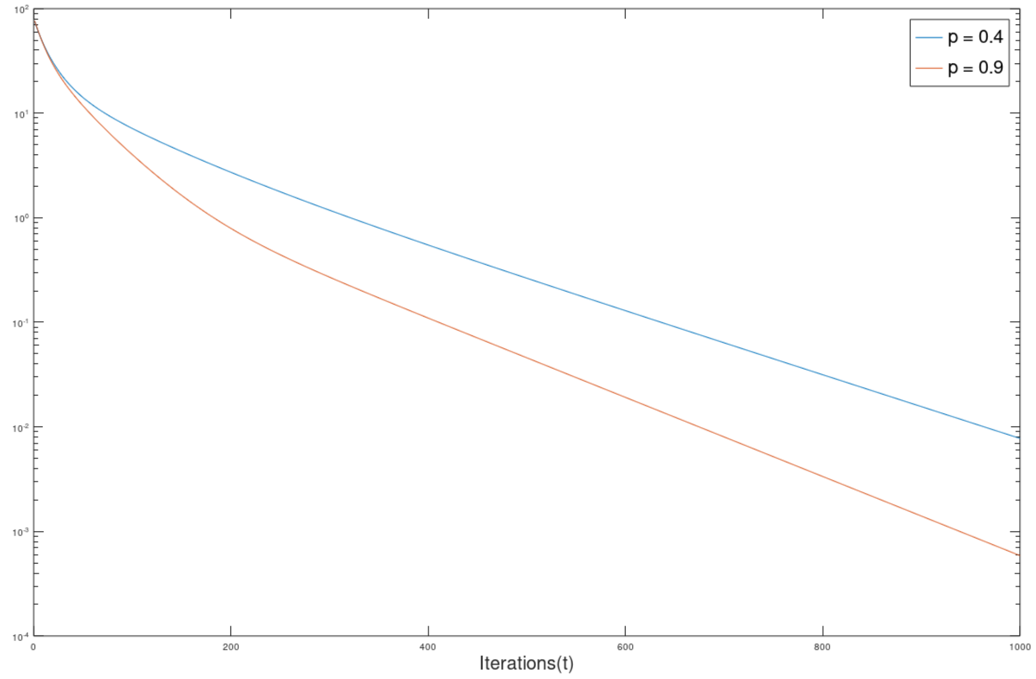
Algoritmul de Consens Proiectat (ACP)

- Date aleatoare (functia Matlab randn)
  $$A \in R^{20\times 50}, W \in R^{20\times 20}$$
- Graf random, probabilitatea de apariție a unei muchii $p = 0.9$
- Curbele urmăresc: $\sum_i var(x_i(t))$ $(blue), \sum_i ||Ax_i(t) - b||$ (red)



Algoritmul de Consens Proiectat (ACP)

- Date aleatoare (functia Matlab randn)
  $$A \in R^{20\times 50}, W \in R^{20\times 20}$$
- Grafuri aleatoare, probabilitatea de apariție a unei muchii $p = 0.4$, respectiv $p = 0.9$
- Curbele urmăresc: $\sum_i ||Ax_i(t) - b||$



References

[1] Nedic, Angelia, Asuman Ozdaglar, and Pablo A. Parrilo. "Constrained consensus and optimization in multi-agent networks." *IEEE Transactions on Automatic Control* 55.4 (2010): 922-938.

[2] S. Boyd, A. Gosh, B. Prabhakar, D. Shah, *Gossip Algorithms: Design, Analysis and Applications, Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*.. Vol. 3. IEEE, 2005.

[3] Márk JELASITY. *Gossip-based protocols for large-scale distributed systems*. PhD Thesis, 2013.