Sisteme și algoritmi distribuiți
Curs 6

Valori și vectori proprii

Definiție. Valorile proprii (*eigenvalues*) ale matricii $A \in R^{n \times n}$ sunt date de n rădăcini ale polinomului caracteristic $p(z) = \det(zI_n - A)$. Mulțimea acestor valori se numește spectrul matricii A și este notat cu:

$$\lambda(A) = \{z : \det(zI_n - A) = 0\}.$$

Definiție. Pentru $\lambda \in \lambda(A)$ numim vectorii nenuli x $\in C^n$ care satisfac $Ax = \lambda x$ vectori proprii (*eigenvectors*). Mai exact x este vector propriu *la dreapta* dacă satisface:

$$Ax = \lambda x$$

și vector propriu *la stânga* dacă satisface:

$$x^H A = x^H \lambda.$$

Un vector propriu definește un subspațiu 1-dimensional care este invariant la premultiplicarea cu A.

Reamintim: pentru x $\in C^n$, x^H reprezintă vectorul x transpus și conjugat.

Valori și vectori proprii - exemple

În cazul matricilor diagonale (triunghiulare), valorile proprii sunt elementele diagonalei, e.g.

$$\lambda(I_n) = \{1, \cdots, 1\}.$$

Dacă $A = A^T$, atunci subspațiul v. p. la dreapta = subspațiul v. p. la stânga.

Când calculăm un vectorul propriu asociat unei valori proprii fixate, în general ne interesează doar vectorul de pe sfera unitate.

Exemplu: Calculați vectorul propriu (la stânga și dreapta) asociat valorii proprii 1 ai matricii $\mathrm{A} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{bmatrix}$.

Valori și vectori proprii (matrici stohastice)

Teorema Perron-Frobenius. Dacă matricea $A \in R^{n \times n}$ este stohastică (pe linii) și ireductibilă atunci: vectorul propriu la stânga $w \in R^n$ satisface $w \geq 0$ și

1) $\rho(A) = 1$ este simplă. (Valoarea Perron-Frobenius)

2) Vectorii proprii asociați lui $\rho(A)$ au componentele pozitive.

3) Fie w v. p. la stânga asociat lui $\rho(A)$ atunci $\lim_{t \to \infty} A^t = 1w^T$. (Proiecția Perron)

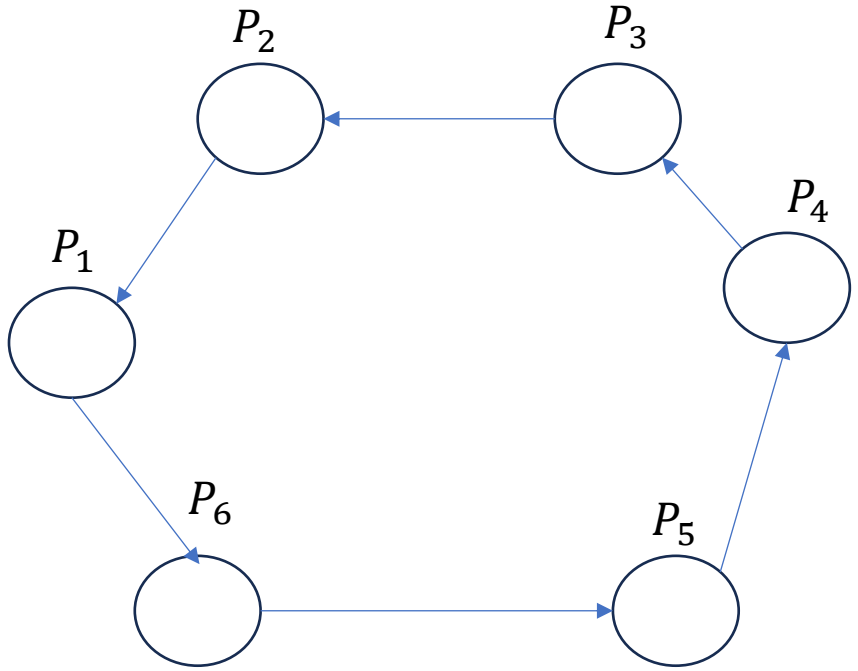
Matrice ireductibilă. Matricea A este *ireductibilă* dacă nu este similară via permutări cu o matrice bloc superior triunghiulară.

Dacă matricea A este matricea de adiacență asociată unui *graf (tare) conex*, atunci A este ireductibilă.

Din cursul trecut

Pentru un inel de dimensiune $n = 6$, matricea asociată este:

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



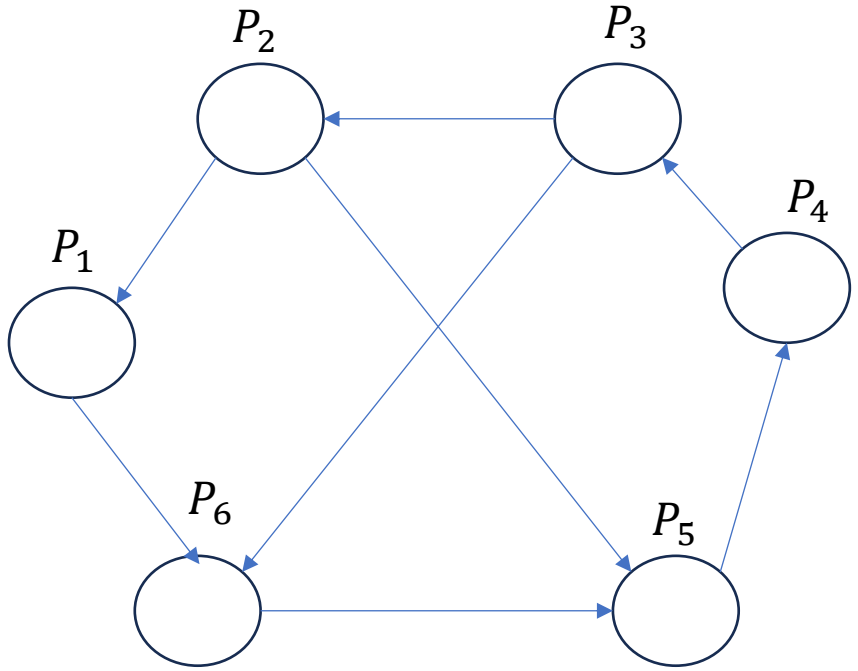

```
A^70:
    0.1667    0.1667    0.1667    0.1667    0.1667    0.1667
    0.1667    0.1667    0.1667    0.1667    0.1667    0.1667
    0.1667    0.1667    0.1667    0.1667    0.1667    0.1667
    0.1667    0.1667    0.1667    0.1667    0.1667    0.1667
    0.1667    0.1667    0.1667    0.1667    0.1667    0.1667
    0.1667    0.1667    0.1667    0.1667    0.1667    0.1667
```

$$= \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix}$$

Exemple

Pentru graful alăturat matricea asociată este:

$$A = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 1/3 & 0 & 0 & 1/3 & 1/3 \\ 1/3 & 0 & 1/3 & 0 & 0 & 1/3 \end{bmatrix}$$




```
A^30:
   0.060607   0.181818   0.242425   0.242424   0.181817   0.090909
   0.060606   0.181818   0.242424   0.242425   0.181818   0.090909
   0.060606   0.181818   0.242424   0.242424   0.181819   0.090909
   0.060606   0.181818   0.242424   0.242424   0.181818   0.090909
   0.060606   0.181818   0.242424   0.242424   0.181818   0.090909
   0.060606   0.181818   0.242424   0.242424   0.181818   0.090909
```

$$= \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} [0.061 \quad 0.182 \quad 0.243 \quad 0.243 \quad 0.182 \quad 0.091]$$

Algoritm Flooding pentru consens

Algoritm Flooding de medie prezentat anterior se exprimă recurent prin actualizarea liniară:

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

Deci $x_i(t+1) = a_i^T x(t)$, iar dinamica stărilor sistemului este:

$$x(t+1) = Ax(t),$$

unde

$$A = \begin{bmatrix} a_1^T \\ \dots \\ a_n^T \end{bmatrix} \in R^{n \times n}, x(t) = \begin{bmatrix} x_1(t) \\ \dots \\ x_n(t) \end{bmatrix} \in R^n.$$

Algoritm Flooding pentru consens

Din dinamică stărilor

$$x(t+1) = Ax(t),$$

se observă ușor:

$$x(t) = A^t x(0),$$

de aceea convergența depinde total de comportamentul matricii A^t (implicit, doar de structura grafului).

Teorema. Dacă matricea A este *stohastică pe linii*, i.e. $a_{ii} + \sum_{j \in \mathcal{N}_i^-} a_{ij} = 1, a_{ij} \geq 0 \ \ \forall j \ \in \mathcal{N}_i^- \cup \{i\}$, atunci se atinge consensul asimptotic:

$$x(t) \to c\mathbf{1} \text{ când } t \to \infty.$$

În plus, dacă matricea A este *stohastică pe coloane*, i.e. $\mathbf{1}^{\mathbf{T}} A = A^T$, atunci valoarea de consens este

$$c = \frac{1}{n} \sum_{i=1}^{n} x_i(0)\,.$$

Algoritm Flooding pentru consens

Fie v vectorul propriu la stânga al matricii A asociat valorii proprii 1, și iterația

$$x(t+1) = Ax(t),$$

atunci

$$v^T x(t) = v^T A^t x(0) = v^T x(0).$$

Observație: Unghiul tuturor iterațiilor $x(t)$ față de v este constant pentru orice t.

De aceea, în cazul convergenței, la limită: $x(\infty) = \lim_{t\in\infty} A^t\, x(0) = \; c\mathbf{1}$ avem relația (din th. P-F)

$$v^T x(\infty) = c = v^T x(0),$$

concluzionând că valoarea de consens este dată de $v^T x(0)$.

Pentru a răspunde la întrebarea:
Care este valoarea de consens a algoritmului de Flooding pe o topologie particulară?
este necesară calcularea vectorul propriu la stânga al matricii A asociat valorii proprii 1.

Consens distribuit (cu procese defecte)

Ipoteze:

- Graf ~~complet~~ conex (nedirectat). Conectivitatea grafului $G, conn(G)$ = numărul minim de noduri care, o dată eliminate din graf, va rezulta un graf neconectat.
- $s < conn(G)$
- Un nod poate fi corect sau defect *Crash* (încetează funcționarea)
- Dacă un nod intră în starea de defect, rămâne în ea până la finalul algoritmului

Urmărim atingerea consensului distribuit sub maxim $s \geq 0$ defecte *Crash*.

Algoritm Flooding pentru consens

Algoritm Flooding de medie prezentat anterior se exprimă recurent prin actualizarea liniară:

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

Deci $x_i(t+1) = a_i^T x(t)$, iar dinamica stărilor sistemului este:

$$x(t+1) = Ax(t),$$

unde

$$A = \begin{bmatrix} a_1^T \\ \dots \\ a_n^T \end{bmatrix} \in R^{n \times n}, x(t) = \begin{bmatrix} x_1(t) \\ \dots \\ x_n(t) \end{bmatrix} \in R^n.$$

Problemă

Fie graful $G = (V, E)$ cu matricea de adiacență ponderată $A \in R^{5\times5}$.

a) Care este valoarea de consens asimptotic a algoritmului Flooding de medie?

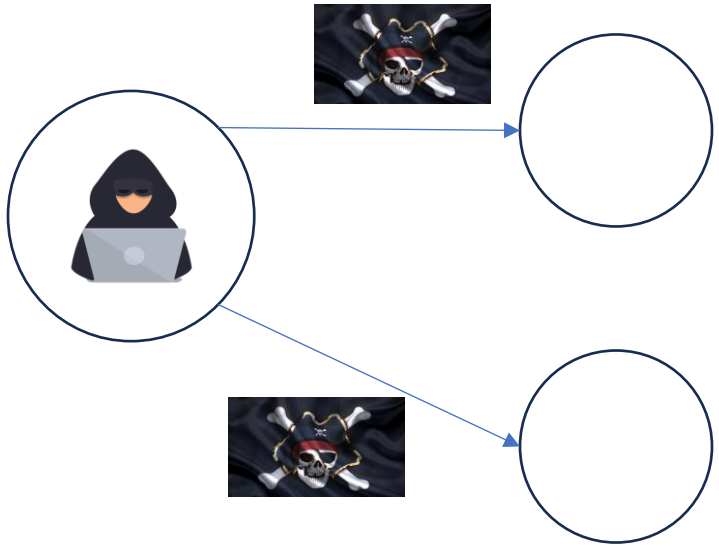
b) Presupunem că la momentul de timp $t = 3$, nodurile 2 și 3 suferă defect de tip *Crash*. Cum se schimbă valoarea de consens asimptotic a algoritmului Flooding de medie în acest caz?

Defecte bizantine

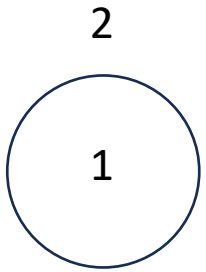
Defect Bizantin

Sub defect bizantin nodurile se comportă malițios, perturbând activitatea întregului sistem (e.g. comportament arbitrar):

- Livrează mesaje atipice execuției algoritmului local
- Actualizează starea după reguli atipice execuției algoritmului local

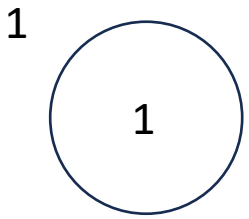


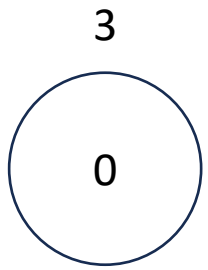
Exemplu



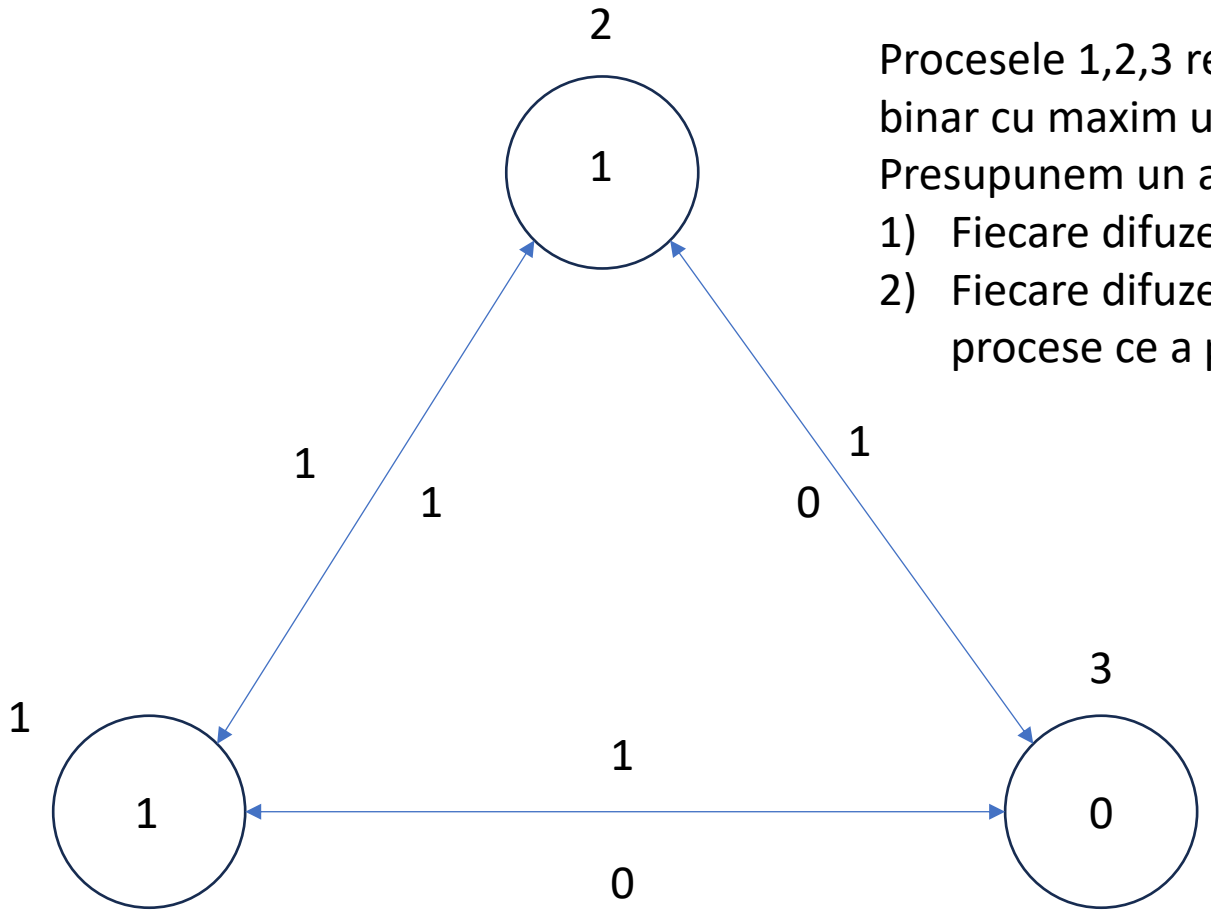
Procesele 1,2,3 rezolva problema de consens bizantin binar cu maxim un defect.
Presupunem un algoritm cu 2 runde:

1) Fiecare difuzeaza propria valoare initiala
2) Fiecare difuzeaza catre fiecare din celelalte procese ce a primit de la al treilea proces





Exemplu

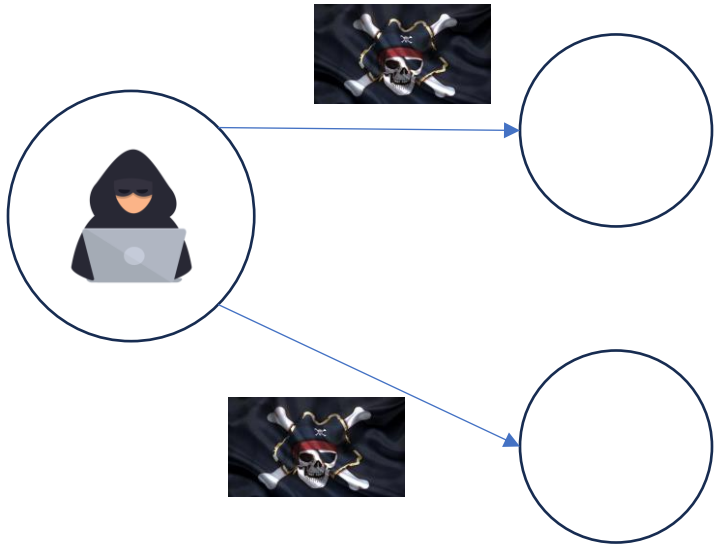


Procesele 1,2,3 rezolva problema de consens bizantin binar cu maxim un defect.
Presupunem un algoritm cu 2 runde:

1) Fiecare difuzeaza propria valoare initiala
2) Fiecare difuzeaza catre fiecare din celelalte procese ce a primit de la al treilea proces

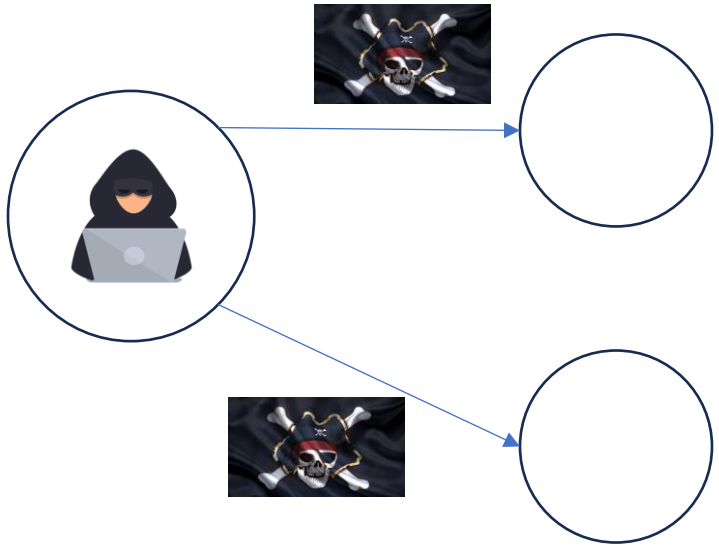
Defect Bizantin

Exemplul precedent stă la baza demonstrației faptul că $\boldsymbol{n > 3f}$ procese sunt necesare pentru toleranța a f defecte bizantine.



Defect Bizantin

Algoritmii robuști la defecte bizantine se bazează pe operații de broadcast speciale pentru a difuza informația corectă în sistem.



Algoritmul ByzFlood

Sistem cu n procese si maxim s defecte (bizantine)

Urmărim consensul binar majoritar (distribuit): $x(0) \in \{0,1\}^n$

$$x_i^* = Maj\big(x(0)\big) = \begin{cases} 1, dacă\ |\{i|x_i(0) = 1\}| \geq \frac{n}{2} + 1 \\ 0, dacă\ |\{i|x_i(0) = 1\}| < \frac{n}{2} + 1 \end{cases}$$

Pentru început, considerăm graful sistemului *complet*.

Presupunem că numărul denoduri din sistem satisface: $n > 4s$.

Fiecare nod își cunoaște indexul în sistem.

Algoritmul ByzFlood

Algoritm **ByzFlood**(Maj()):

M_i: - int $x(0)$ (starea inițială, inițial egal cu v_i)
- int s, integer (număr maxim de defecte)
- int *t*, integer, inițial 0

Funcție transformare nod i ():
% Runda 1
1. **Bcast**($x(0)$) % difuzează $x(0)$
2. Fie U mulțimea mesajelor $v_j = x_j(t)$ primite restul nodurilor
3. $Majority(t) = Maj(U)$
4. $mult(t) = numărul\ de\ apariții\ al\ Majority(t)$

% Runda 2
1. **If** (i==t): % nodul leader/king
    1. **Bcast**($Majority(t)$)
2. **Else**: recv(Tie, P_t)
3. **If** (mult(t) > n/2 + s):
    1. $x(t) := Majority(t)$)
4. **Else**: $x(t) :=$ Tie
5. **If** (t>s+1):
    1. **Return** $x(t)$
6. $t := t+ 1$

1. Între cele s+1 iterații există cel puțin una (să zicem k) în care nodul king este nod corect.
2. La iterația k, două noduri P_i și P_j se pot afla în situațiile:

- P_i și P_j actualizează x_i și x_j pe baza majorității (dacă valoarea majorității este b, atunci mult> n/2 + s; de aceea majoritatea proceselor adoptă valoare b)
- P_i și P_j actualizează x_i și x_j pe baza Tie
- P_i act. pe baza majorității și P_j actualizează pe baza Tie. P_i are mult> n/2 + s. De asemenea, și P_t are primit cel puțin n/2 voturi pentru aceeași valoare.

În cele 3 situații P_i și P_j ajung la consens.

Algoritmul ByzFlood