



UNIVERSITATEA BABEȘ-BOLYAI
Facultatea de Matematică și Informatică



INTELIGENȚĂ ARTIFICIALĂ

Sisteme inteligente

Sisteme care învață singure

Laura Dioșan

Sumar

A. Scurtă introducere în Inteligența Artificială (IA)

B. Sisteme inteligente

■ Sisteme care învață singure

- Metoda celor mai mici patrate, Gradient Descent, Logistic regression
- Arbori de decizie
- Rețele neuronale artificiale
- Mașini cu suport vectorial
- Algoritmi evolutivi

■ Sisteme bazate pe reguli

■ Sisteme hibride

A. Rezolvarea problemelor prin căutare

■ Definirea problemelor de căutare

■ Strategii de căutare

- Strategii de căutare neinformate
- Strategii de căutare informate
- Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
- Strategii de căutare adversarială

Materiale de citit și legături utile

- ❑ capitolul VI (18) din *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- ❑ capitolul 10 și 11 din *C. Groșan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*
- ❑ capitolul V din *D. J. C. MacKey, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003*
- ❑ capitolul 3 din *T. M. Mitchell, Machine Learning, McGraw-Hill Science, 1997*

Conținut

□ Sisteme inteligente

■ Sisteme care învață singure (SIS)

- Instruire (învățare) automata (Machine Learning - ML)
 - Problematică
 - Proiectarea unui sistem de învățare automată
 - Tipologie
 - Învățare supervizată
 - Învățare nesupervizată
 - Învățare cu întărire
 - Teoria învățării
- Exemple de sisteme

Conținut

❑ Sisteme care învață singure (SIS)



- ❑ "Field of study that gives computers the ability to learn without being explicitly programmed." -- Arthur Samuels (1959)

■ Invățare

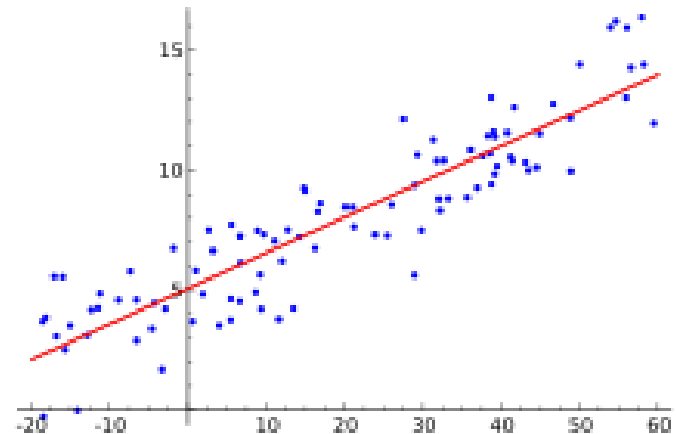
- ❑ Supervizată
- ❑ Nesupervizată
- ❑ Reinforcement

Sisteme inteligente – SIS – Învățare automată

□ Metoda celor mai mici pătrate

■ Presupunem cazul unei probleme de regresie

- Date de intrare $x \in \mathbb{R}^d$
- Date de ieșire $y \in \mathbb{R}$



- Se cere un model **liniar** f care transformă x în y
- $f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_d x_d$
- Învățare supervizată

Sisteme inteligente – SIS – Învățare automată

□ Metoda celor mai mici pătrate

■ Presupunem cazul unei probleme de regresie

- Date de intrare $x^i \in \mathbb{R}^d$, $i=1,n$
- Date de ieșire $y^i \in \mathbb{R}$
- Se cere un model **liniar** f care transformă orice x^i în y^i , $i=1,n$
- $f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_d x_d$
- Se poate defini o funcție de cost
- $\text{Loss} = \sum_{i=1,n} (y^i - f(x^i))^2$ -- minimizată \rightarrow valorile optime ale lui β

$$\mathbf{x} = (1, x) = (1, x_1, x_2, \dots, x_d)^T \in \mathbb{R}^{d+1}$$

$$\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \dots, \beta_d)^T \in \mathbb{R}^{d+1}$$

$$f(\mathbf{x}) = \mathbf{x}^T \boldsymbol{\beta}$$

$$\text{Loss}(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X} \boldsymbol{\beta}\|^2$$

$$\mathbf{X} = \begin{matrix} 1 & x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,d} \\ . & . & . & . & . & . \\ 1 & x_{n,1} & x_{n,2} & x_{n,3} & \dots & x_{n,d} \end{matrix}$$

□ Metoda celor mai mici pătrate

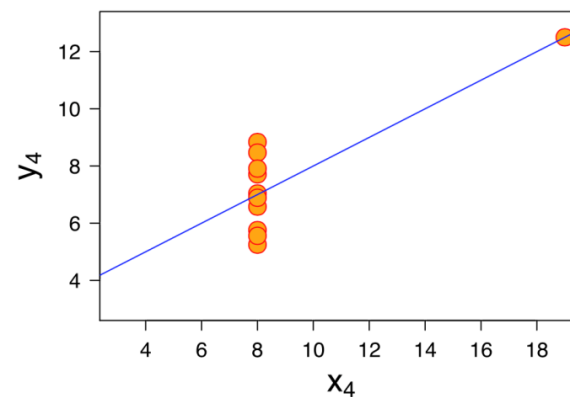
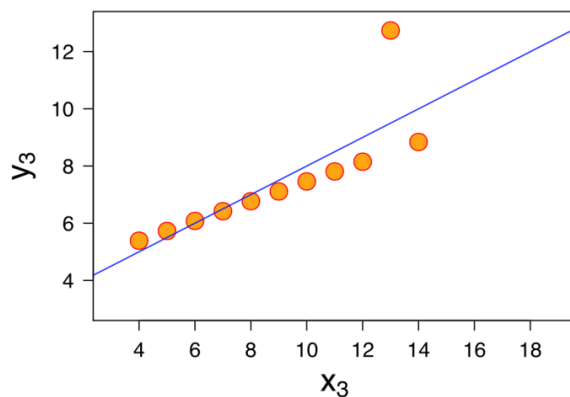
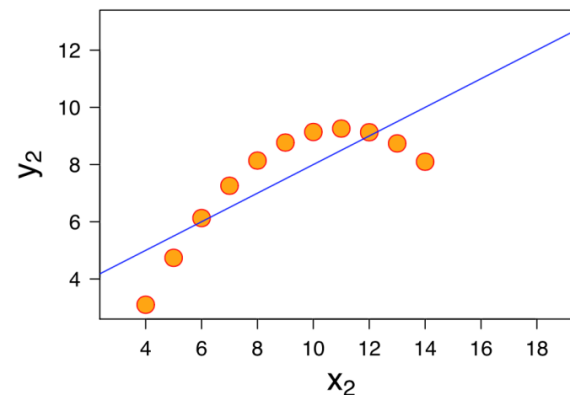
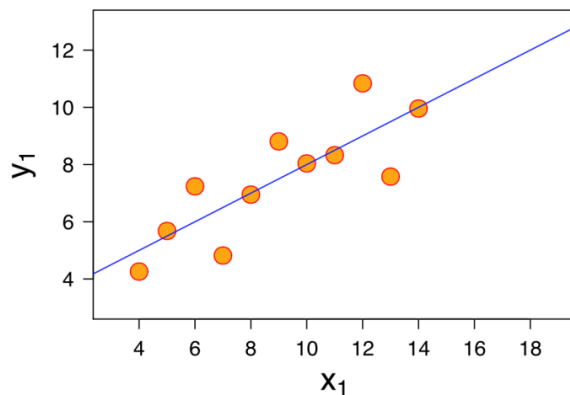
■ Presupunem cazul unei probleme de regresie

- Date de intrare $x^i \in \mathbb{R}^d$, $i=1,n$
- Date de ieșire $y^i \in \mathbb{R}$
- Se cere un model **liniar** f care transformă orice x^i în y^i , $i=1,n$
- $f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_d x_d$
- Se poate defini o funcție de cost
- $\text{Loss} = \sum_{i=1,n} (y^i - f(x^i))^2$ -- minimizată \rightarrow valorile optime ale lui β
- $\beta = (X^T X)^{-1} X^T y$
- Dacă $d = 1$, $\beta_1 = \text{cov}(x,y)/\text{var}(x)$, $\beta_0 = y - \beta_1 x$

Sisteme inteligente – SIS – Învățare automată

□ Metoda *celor mai mici pătrate*

■ Anscombe Quartet

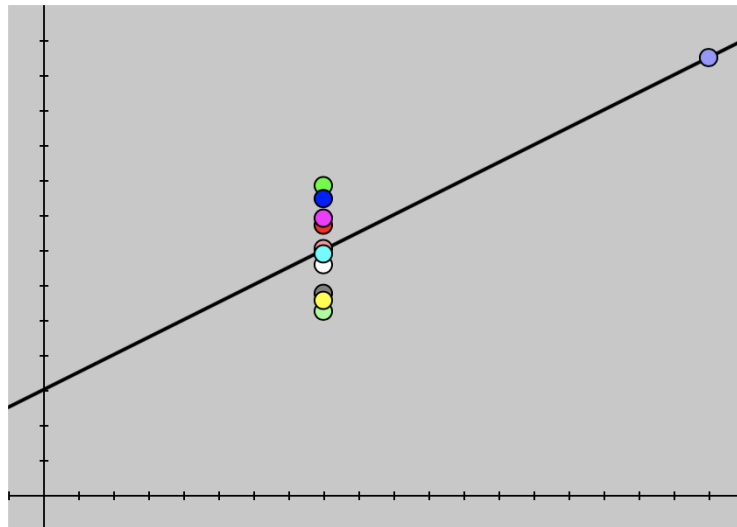


[Ref. imagine](#)

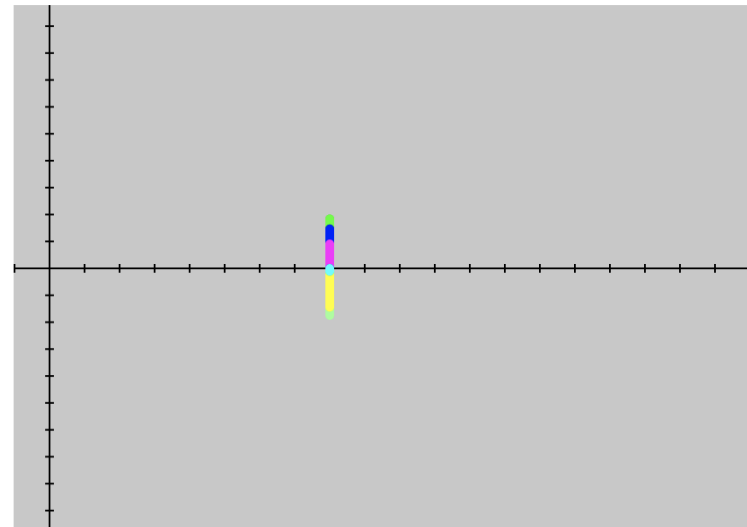
Sisteme inteligente – SIS – Învățare automată

- Metoda *celor mai mici pătrate*
 - Residual plot

Scatter plot



Residual plot

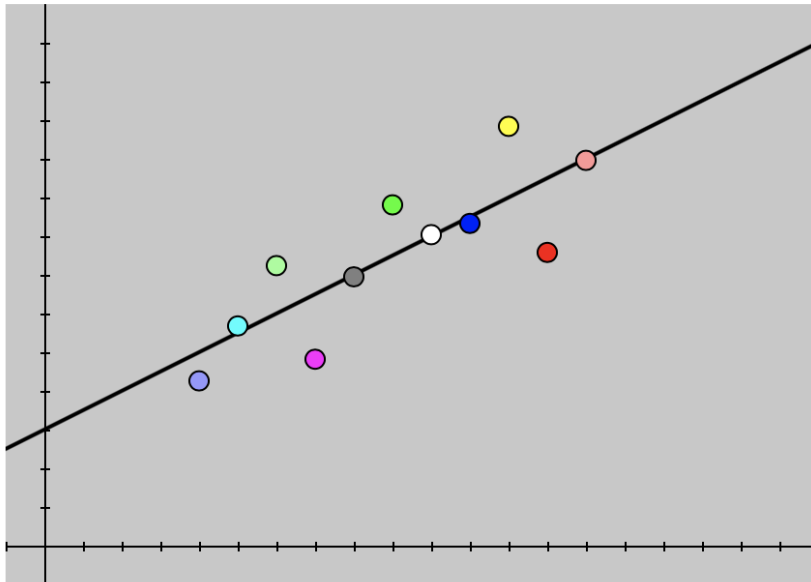


[Ref. imagine](#)

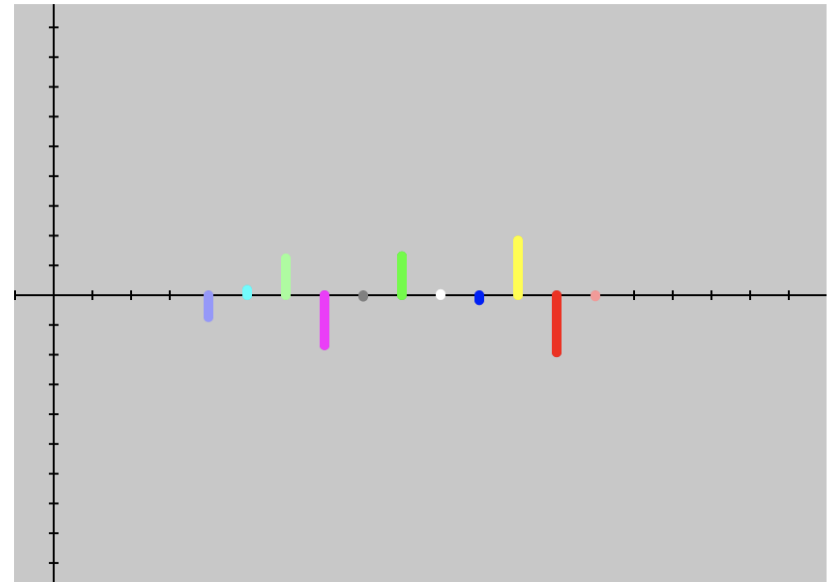
Sisteme inteligente – SIS – Învățare automată

- Metoda *celor mai mici pătrate*
 - Residual plot

Scatter plot



Residual plot



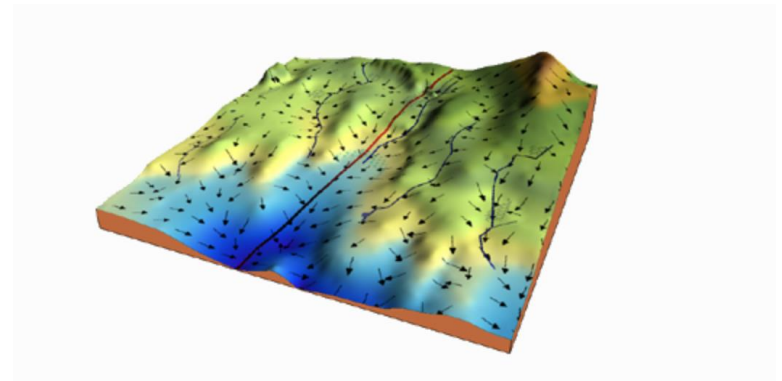
[Ref. imagine](#)

Sisteme inteligente – SIS – Învățare automată

□ Metoda *gradient descent*

■ Presupunem cazul unei probleme de regresie

- Date de intrare $x \in \mathbb{R}^d$
- Date de ieșire $y \in \mathbb{R}$



- Se cere un model **liniar** f care transformă x în y
- $f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_d x_d$
- Învățare supervizată – aflarea valorilor β pentru care eroarea este optima (minima) \rightarrow Derivarea loss-ului după β

[Ref. imagine](#)

□ Metoda *gradient descent*

■ Modelarea coeficienților β :

□ la iterația 0: valori random (sau 0)

□ la iterația $t + 1$ ($t = 0, 1, 2, \dots$)

$$\beta_k(t+1) = \beta_k(t) - \text{learning_rate} * \text{error}(t) * x_k, \quad k=1,2,\dots,d$$

$$\beta_0(t+1) = \beta_0(t) - \text{learning_rate} * \text{error}(t)$$

■ Unde

■ $\text{error}(t) = \text{computed} - \text{realOutput}$

■ $\text{error}(t) = \beta_0(t) + \beta_1(t)*x_1 + \beta_2(t)*x_2 + \dots + \beta_d(t)*x_d - y$

Sisteme inteligente – SIS – Învățare automată

□ Metoda *gradient descent* – versiuni:

■ Stochastic GD

- Eroarea se calculează pentru fiecare exemplu de antrenament
- Modelul se updatează pentru fiecare exemplu de antrenament (*online learning*)

■ Batch GD

- Eroarea se calculează pentru fiecare exemplu de antrenament
- Modelul se updatează după ce toate exemplele de antrenament au fost evaluate (la finalul unei epoci)

■ Mini-batch GD

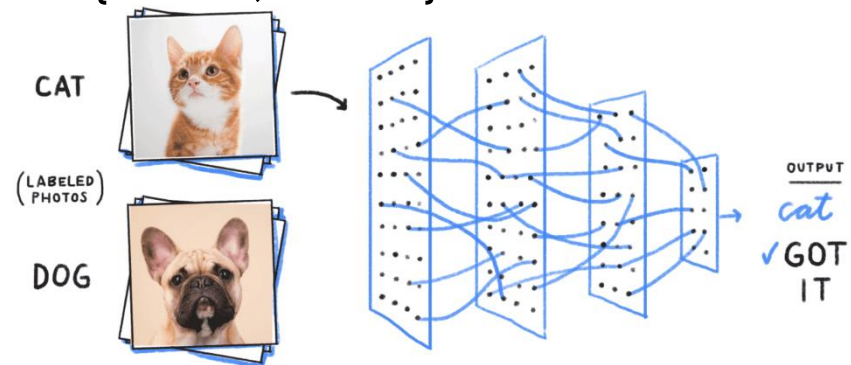
- Combinare a precedentelor două
- Setul de date se împarte în mai multe părți (mini-batch-uri)
- Eroarea se calculează pentru fiecare exemplu de antrenament dintr-un mini-batch
- Modelul se updatează pentru fiecare exemplu de antrenament dintr-un mini-batch

Sisteme inteligente – SIS – Învățare automată

□ Regresie Logistică (clasificare)

■ Presupunem cazul unei probleme de clasificare

- Date de intrare $x^i \in \mathbb{R}^d$, $i=1,n$
- Date de ieșire $y^i \in \{0,1\}$ sau {label1, label2}



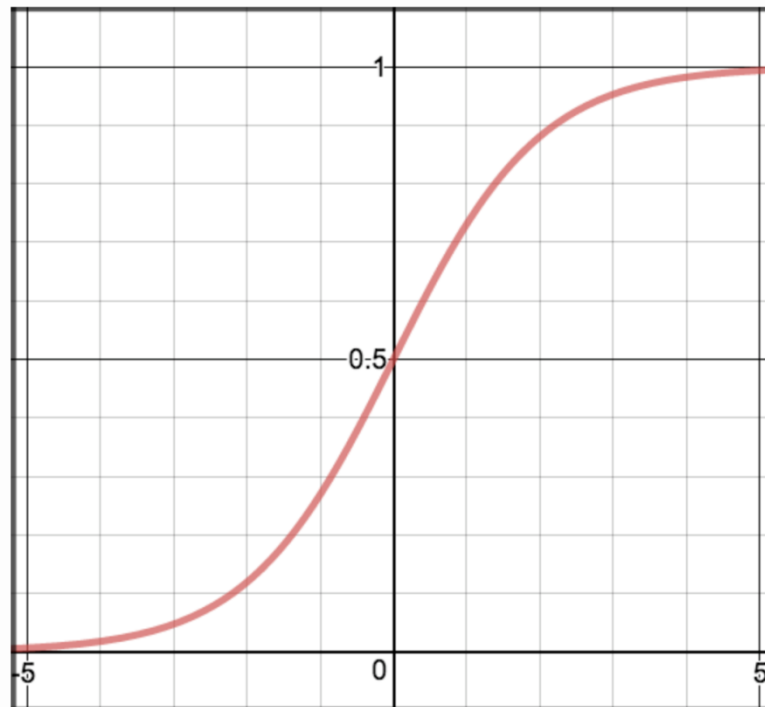
- Se cere un model **liniar** f care separa orice x^i în 2 clase (etichetate cu 0 și 1)
- $f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_d x_d$
- Învățare supervizată

- ❑ Regresie Logistică (clasificare)
 - Mapează datele într-un set discret de clase (label-uri)
 - Tipuri:
 - ❑ Binar (Pass/Fail, True/False)
 - ❑ Multi (Cat,Dog,Panda)
 - ❑ Ordinal (Low,Medium,High)
 - Folosește funcția sigmoid pentru a decide clasa de apartenență
 - Putem folosi gradient descent pentru minimizarea erorii

Sisteme inteligente – SIS – Învățare automată

- Regresie Logistică – sigmoid:
 - Mapează orice număr real in intervalul (0,1)

$$S(z) = \frac{1}{1 + e^{-z}}$$



[Ref. imagine](#)

□ Regresie Logistica

■ Modelarea coeficienților β :

- la iterația 0: valori random (sau 0)

- la iterația $t + 1$ ($t = 0, 1, 2, \dots$)

$$\beta_k(t+1) = \beta_k(t) - \text{learning_rate} * \text{error}(t) * x_k, \quad k=1,2,\dots,d$$

$$\beta_0(t+1) = \beta_0(t) - \text{learning_rate} * \text{error}(t)$$

- Unde

- $\text{error}(t) = \text{Sigmoid}(\text{computed}) - \text{realOutput}$

- $\text{error}(t) = \text{Sigmoid}(\beta_0(t) + \beta_1(t)*x_1 + \beta_2(t)*x_2 + \dots + \beta_d(t)*x_d) - y$

■ Clasificarea rezultatelor

- $(0,1) \rightarrow [\text{label}_0, \text{label}_1, \dots, \text{label}_n]$

Recapitulare

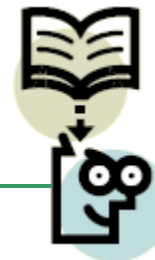


□ Sisteme care învață singure (SIS)

■ Instruire (învățare) automata (Machine Learning - ML)

- Învățare supervizată → datele de antrenament sunt deja etichetate cu elemente din E , iar datele de test trebuie etichetate cu una dintre etichetele din E pe baza unui model (învățat pe datele de antrenament) care face corespondența date-etichete
- Învățare nesupervizată → datele de antrenament NU sunt etichetate, trebuie învățat un model de etichetare, iar apoi datele de test trebuie etichetate cu una dintre etichetele identificate de model

Recapitulare



□ Sisteme care învață singure (SIS)

■ Metoda celor mai mici patrate

- Supervizată
- Output continu (vânzări, preț, etc.)
- Panta constantă

■ Gradient descent

- Supervizată
- Output continu (vânzări, preț, etc.)
- Optimizare

■ Regresie Logistică

- CLASIFICARE
- Supervizată
- Output – label-uri (clase) – set discret
- Folosește Gradient descent

Cursul următor

A. Scurtă introducere în Inteligența Artificială (IA)

B. Sisteme inteligente

■ Sisteme care învață singure

- Arbori de decizie
- Rețele neuronale artificiale
- Mașini cu suport vectorial
- Algoritmi evolutivi

■ Sisteme bazate pe reguli

■ Sisteme hibride

C. Rezolvarea problemelor prin căutare

■ Definirea problemelor de căutare

■ Strategii de căutare

- Strategii de căutare neinformate
- Strategii de căutare informate
- Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
- Strategii de căutare adversială

Cursul următor – Materiale de citit și legături utile

- ❑ Capitolul VI (19) din *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- ❑ capitolul 8 din *Adrian A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001*
- ❑ capitolul 12 și 13 din *C. Groșan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*
- ❑ Capitolul V din *D. J. C. MacKey, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003*
- ❑ Capitolul 4 din *T. M. Mitchell, Machine Learning, McGraw-Hill Science, 1997*

□ Informațiile prezentate au fost colectate din diferite surse de pe internet, precum și din cursurile de inteligență artificială ținute în anii anteriori de către:

■ Conf. Dr. Mihai Oltean –
www.cs.ubbcluj.ro/~moltean

■ Lect. Dr. Crina Groșan -
www.cs.ubbcluj.ro/~cgrosan

■ Prof. Dr. Horia F. Pop -
www.cs.ubbcluj.ro/~hfpop