

# Tehnici de Optimizare

## Curs 1

Andrei Pătrașcu  
[andrei.patrascu@fmi.unibuc.ro](mailto:andrei.patrascu@fmi.unibuc.ro)

Departament Informatică  
Universitatea din București

# Orar

## Patrascu Andrei-Mihai

Universitatea din Bucuresti, Facultatea de Matematica si Informatica, str. Academiei 14, Bucuresti

	8 8:00-9:00	9 9:00-10:00	10 10:00-11:00	11 11:00-12:00	12 12:00-13:00	13 13:00-14:00	14 14:00-15:00	15 15:00-16:00	16 16:00-17:00	17 17:00-18:00	18 18:00-19:00	19 19:00-20:00
Lu							AlgParaleli 453 10	AlgParaleli 454 L-322				
Ma			TehOptimizare 331/332/333/334 2(Pompeiu)	AlgParaleli 452 214	AlgParaleli 451 214	TehOptimizare 331 L-322						

## Oprea Teodor

Universitatea din Bucuresti, Facultatea de Matematica si Informatica, str. Academiei 14, Bucuresti

	8 8:00-9:00	9 9:00-10:00	10 10:00-11:00	11 11:00-12:00	12 12:00-13:00	13 13:00-14:00	14 14:00-15:00	15 15:00-16:00	16 16:00-17:00	17 17:00-18:00	18 18:00-19:00	19 19:00-20:00
Lu			GeomDiferent 221 10	TehOptimizare 333 L-204								
	TehOptimizare 332 L-321			TehOptimizare 334 L-204								

- ▶ 2 ore curs săptămânal
- ▶ 2 ore seminar / laborator (necesare cunoștințe MATLAB)



# Bibliografie si resurse

- ⇒ Moodle pentru comunicare și resurse
- ⇒ Documentație Matlab pentru laborator (pe moodle)
  
- ▶ I. Necoara, Metode de Optimizare Numerica, Politehnica Press. (material slide-uri)
- ▶ Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, Kluwer
- ▶ Bertsekas, Nonlinear Programming, Athena Scientific
- ▶ D. Luenberger, Linear and Nonlinear Programming, Kluwer



# Regulament Evaluare

1. Examen scris în sesiune: 70 %
2. Colocviu în săptămâna 9/10: 30 %
3. Punctaj seminar: max. 10 %

Condiții promovare:

- ▶ **nota 5 examen**
- ▶ **nota 5 colocviu**



# Obiectivele cursului si subiectele atinse

## Obiective

- ▶ recunoasterea/formularea de probleme de optimizare
- ▶ dezvoltarea de cod Matlab pentru rezolvarea problemelor de optimizare
- ▶ caracterizarea/identificarea solutiilor
- ▶ caracterizarea limitelor de performanta a algoritmilor numerici de optimizare

## Subiecte atinse

- ▶ multimi si functii convexe
- ▶ clase de probleme de optimizare
- ▶ conditii de optimalitate
- ▶ algoritmi numerici de optimizare si proprietatile lor
- ▶ exemple si aplicatii

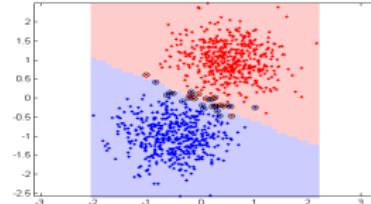
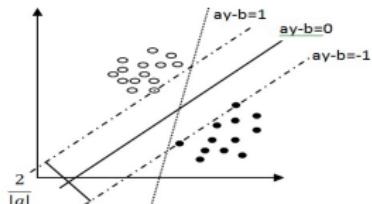
# Invatare automata - Clasificare de imagini

Tehnicile de **clasificare** ("pattern recognition") se ocupă cu identificarea clasei din care un obiect studiat face parte.

- ▶ **etapa de antrenare:** se acumulează un set de observații (sau instante) cu apartenența la categorii cunoscute;
- ▶ **etapa de clasificare:** se dezvoltă modele matematice (estimatoare) cu rolul de a clasifica un nou obiect cu apartenența necunoscută;

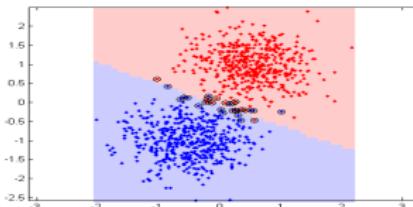
Aplicații:

1. recunoașterea email-urilor de tip spam sau malware;
2. recunoașterea vocii/fetei;
3. detectia de tipare in cadrul unei imagini;
4. recunoașterea scrisului de mana.



## Tehnica Support Vector Machine - etapa de antrenare

- Dispunem de un set de puncte recunoscute *a priori*:  $\{y_i\}_{i=1}^m$  (e.g. puncte de culori diferite). Pentru fiecare punct  $y_i$  cunoastem clasa din care face parte:  $c_i$  cu valoarea  $+1$  daca  $y_i$  este de culoare rosie sau  $c_i = -1$  daca este de culoare albastra.
- **Principiu antrenare:** Se determina un model matematic (hiperplan)  $H = \{y \in \mathbb{R}^n : a^T y = b\}$  astfel incat orice  $y_i$  ce satifice  $a^T y_i \leq b - \xi$  este de culoare rosie, iar daca  $a^T y_i \geq b + \xi$  este de culoare albastra.
- Presupunem ca datele  $\{y_i\}_{i=1}^m$  sunt liniar separabile si selectam doua hiperplane marginale descrise de:  $a^T y - b = 1$  si  $a^T y - b = -1$  ce nu contin puncte intre ele.
- **Problema:** Determinati  $(a, b)$  ce maximizeaza distanta dintre cele doua hiperplane, descrisa de  $2/\|a\|$ .



# Support Vector Machine

Etapa de antrenare se reduce la problema de optimizare:

$$\begin{aligned} & \min_{a \in \mathbb{R}^n, b \in \mathbb{R}} \|a\|^2 \\ \text{s.l. : } & c_i(a^T y_i - b) \geq 1 \quad \forall i = 1, \dots, m, \end{aligned}$$

unde  $a$  si  $b$  reprezinta parametrii hiperplanului, iar  $c_i$  indica clasa (culoarea) obiectului  $y_i$ .

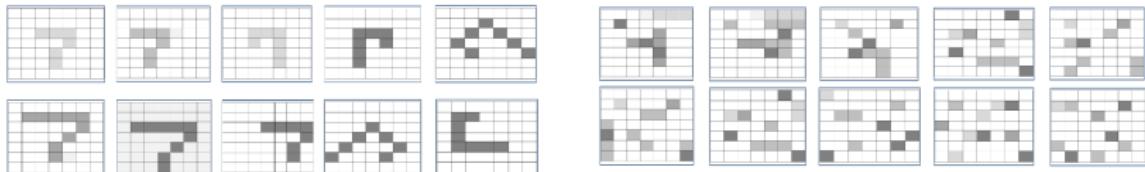
**Principiu clasificare:** Hiperplanul cu parametrii  $(a, b)$ , obtinut in etapa de antrenare, se utilizeaza pentru identificarea unui nou obiect  $z$ ;

$c(z) = \text{sgn}(a^T z - b) \Rightarrow$  daca  $a^T z \leq b$  atunci obiectul apartine clasei 1, altfel daca  $a^T z \geq b$  apartine clasei 2.



## Aplicarea tehnicii SVM in recunoasterea de imagini

- Reprezentarea numerica a imaginilor: fiecare pixel este definit de o valoare (e.g. intre 0 și 256) ce contine culoarea acestuia.
- Consideram imagini mono-colore de dim.  $7 \times 7$  ale cifrei 7 unde pixelii sunt reprezentati de nivele de gri cu valori intre 0 si 5.
- **Problema:** Determinati daca intr-o imagine data se afla cifra 7.
- **Etapa de antrenare:** acumulam un set de imagini de antrenare ale cifrei 7 in diferite pozitii (clasa I) si imagini aleatorii complet diferite de cifra 7 (clasa II).
- Imagineii i se asociaza un vector  $y_i \in \mathbb{N}^{49}$  (cu valori intre 0 și 5) si indexul  $c_i$  al clasei (dacă  $c_i = 1$  atunci continde cifra 7, daca  $c_i = -1$  atunci imaginea este aleatorie).



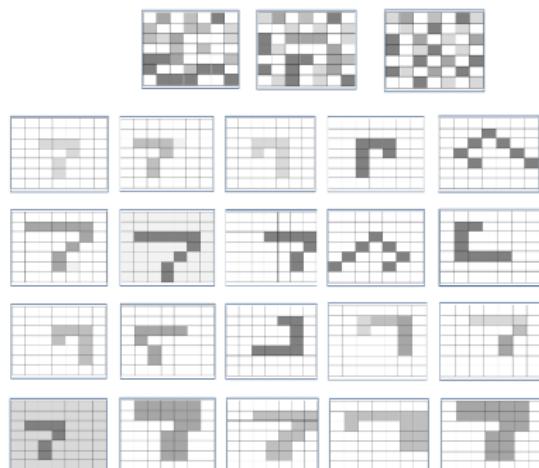
# Aplicarea tehnicii SVM in recunoasterea de imagini

- Determinam hiperplanul optim cu parametrii  $(a, b)$

## Etapa de clasificare:

- Pentru *imaginile de test* (de mai jos) calculam valoarea hiperplanului:

$$a^T y - b \begin{cases} < 0, & \text{atunci imaginea dată de } y \text{ nu conține cifra 7;} \\ \geq 0, & \text{atunci imaginea dată de } y \text{ conține cifra 7.} \end{cases}$$



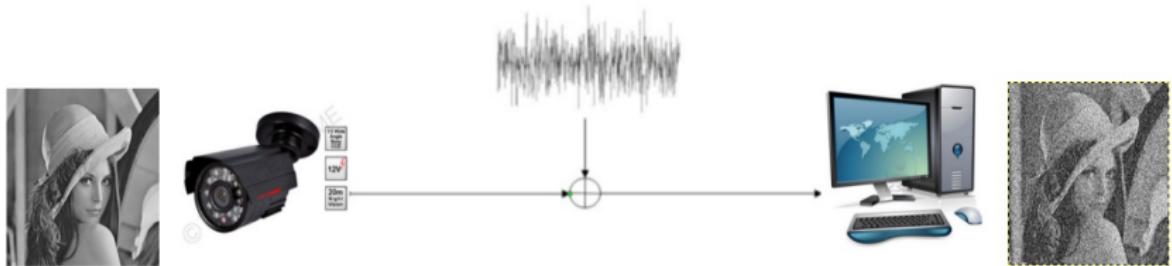
## Rezultate

- ▶ clasificarea unui set de imagini aleatorii cu densitate mare de pixeli gri si respectiv, imagini cu cifra 7 transformata in diverse moduri (translatie la stanga/dreapta, inclinare, etc.) atinge o rata de succes (recunoastere corecta) de **aproximativ 80%**;
- ▶ clasificarea unui set de imagini aleatorii cu densitate mica si respectiv, imagini cu cifra 7 transformata in diverse moduri (translatie la stanga/dreapta, inclinare), atunci rezulta o rata de succes de **aproximativ 52%**.

⇒ Obtinem o rata mai mica de succes in cel de-al doilea caz din doua motive:

- (i) similaritatea ridicata intre imaginile cu densitate mica si cele ce contin cifra 7;
- (ii) numarul relativ mic de imagini de antrenare.

# Procesare de imagini



In general, procesele ingineresti implica achizitia/ prelucrarea/ comunicatia unui numar de semnale/date.

## Problema majora:

modificari nedorite ale semnalelor (*zgomot*) rezultate din:

- ▶ Achizitia semnalelor (e.g. imagine, video, audio)
- ▶ Comunicatii imperfecte

Exemplu: semnale cu detalii excesive au *variatie totala* mare (i.e. integrala gradientului absolut al semnalului este mare)  $\Rightarrow$  reducerea zgomotului echivalenta cu reducerea variatiei totale

# Procesare de imagini

- ▶ Recuperarea perfecta a semnalului original este **imposibila**
- ▶ Concluzie: cautam “cea mai buna” aproximare a imaginii corupte
- ▶ Aproximarea se determina utilizand *algoritmi de optimizare* (pentru niveluri rezonabile de zgomot avem recuperare “aproape” perfecta)

Modelare pentru semnale 2D (tip imagine):

- ▶ Asociem imaginii o matrice  $X \in \mathbb{R}^{m \times n}$  sau un vector  $x \in \mathbb{R}^{mn}$
- ▶ Dimensiunile  $(m, n)$  reprezinta numarul de pixeli pe linii si coloane.
- ▶ Ex.: Pentru o imagine cu dimensiuni  $640 \times 480$  pixeli asociem  $X \in \mathbb{R}^{640 \times 480}$ .

## Procesare de imagini - total variation denoising

Semnale cu detalii excesive au *variatie totală* mare (i.e. integrala gradientului absolut al semnalului este mare)  $\implies$  reducerea zgomotului echivalentă cu reducerea variatiei totale:

$$\min_{Y \in \mathbb{R}^{m \times n}} \frac{1}{2} \|Y - X\|^2 + TV(Y)$$

- ▶  $TV =$  “total variation” funcție cuantificare variatiei totale
- ▶  $TV(y) = \sum_{i,j} |y_{i+1,j} - y_{i,j}| + |y_{i,j+1} - y_{i,j}|;$

Rezultate (noisy image/denoised image):



## Procesare de imagine - compresia imaginilor

- ▶ O imagine se reprezinta numeric printr-o matrice !

### Aproximare matrice cu matrice de rang mai mic via DVS:

- ▶ fie  $A \in \mathbb{R}^{m \times n}$  cu  $\text{rang}(A) = r$  si  $A = U\Sigma V^T = \sum_{i=1}^r \sigma_i u_i v_i^T$
- ▶ cautam matricea  $\hat{A}$  avand rangul  $p < r$  a.i.  $\hat{A} \approx A$ , i.e.

$$\min_{\hat{A}: \text{rang } \hat{A} \leq p} \|A - \hat{A}\|$$

- ▶ solutie:  $\hat{A} = \sum_{i=1}^p \sigma_i u_i v_i^T$ , deci  $\|A - \hat{A}\| = \sum_{i=p+1}^r \sigma_i u_i v_i^T$

**Demonstratie:** fie  $\text{rang}(B) \leq p$ , deci  $\dim \text{Null}(B) \geq n - p$ . Apoi:  $\dim \text{Span} \{v_1, \dots, v_{p+1}\} = p + 1$ . Cele 2 spatii se intersecteaza:

$$\exists z : \|z\| = 1, Bz = 0, z \in \text{Span}\{v_1, \dots, v_{p+1}\}$$

$$(A - B)z = Az = \sum_{i=1}^{p+1} \sigma_i u_i v_i^T z = \sum_{i=1}^{p+1} (\sigma_i v_i^T z) u_i \quad (z \perp v_{p+2})$$

Cum  $u_i^T u_j = \delta_{ij}$  avem  $\|(A - B)z\|^2 = \sum_{i=1}^{p+1} \sigma_i^2 (v_i^T z)^2 \|u_i\|^2 \geq \sigma_{p+1}^2 \|z\|^2$ .

$$\|A - B\| = \max_{\|x\|=1} \|(A - B)x\| \geq \|(A - B)z\| \stackrel{\|z\|=1}{\geq} \sigma_{p+1} = \|A - \hat{A}\|$$

## Procesare de imagine - compresia imaginilor

**Interpretare:** pentru  $A = U\Sigma V^T = \sum_{i=1}^r \sigma_i u_i v_i^T$  componentele principale de rang 1,  $u_i v_i^T$ , sunt ordonate in functie de importanta lor data de  $\sigma_i$ .

Fie imaginea originala  $359 \times 371$  pixeli (Melancolia de Durer, 1514):



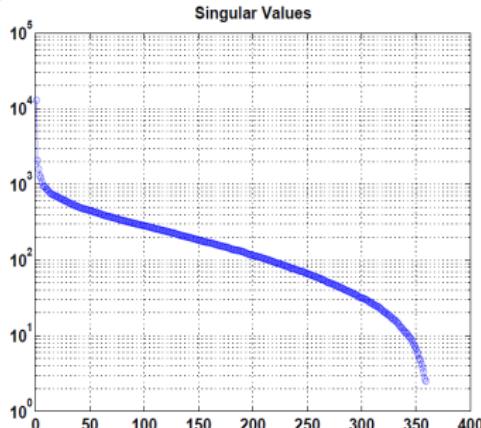
pe care o putem scrie ca o matrice  $A$  de dimensiune  $359 \times 371$  (fiecare intrare  $(i,j)$  reprezinta intensitatea de gri a pixelului  $(i,j)$  cu valori intre 0 (negru) si 255 (alb)), ce poate fi descompusa via DVS:  $A = U\Sigma V^T$ , unde  $U$  este  $359 \times 359$ ,  $\Sigma$  este  $359 \times 371$  si  $V$  este  $371 \times 371$ .

## Compresia imaginilor

Imaginea originală are  $359 \times 371 = 133.189$  pixeli. Dar matricea  $A$  poate fi scrisă ca suma de componente principale:

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^T,$$

unde fiecare matrice  $u_i v_i^T$  (componenta principală) de rang 1 este de marimea matricei originale  $A$ . Dar pentru ca valorile singulare  $\sigma_i$  sunt ordonate  $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ , este posibila o compresie semnificativă a imaginii, atât timp cat spectrul valorilor singulare are doar câteva intrări mari și restul sunt mici!

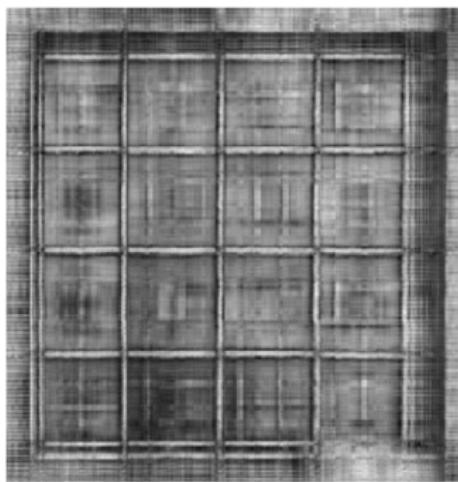
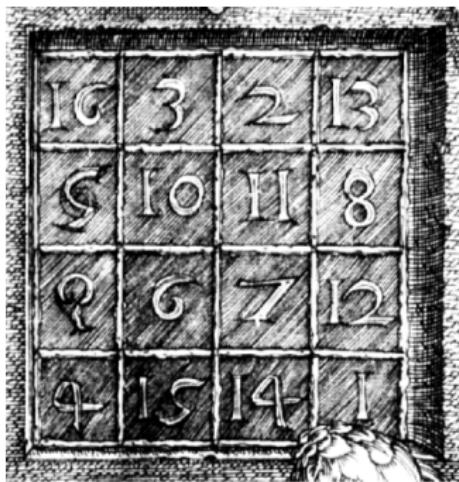


Spectrul lui  $A$  conține doar 100 – 200 componente principale max.!

## Compresia imaginilor

Putem deci reconstrui fidel imaginea folosind doar o submultime de componente principale. De exemplu, putem recupera o imagine asemenatoare cu doar **20** de componente principale folosind comenziile din Matlab:

$$[U, S, V] = \text{svd}(A) \quad \text{si} \quad B = U(:, 1 : 20) S(1 : 20, 1 : 20) V(:, 1 : 20)$$



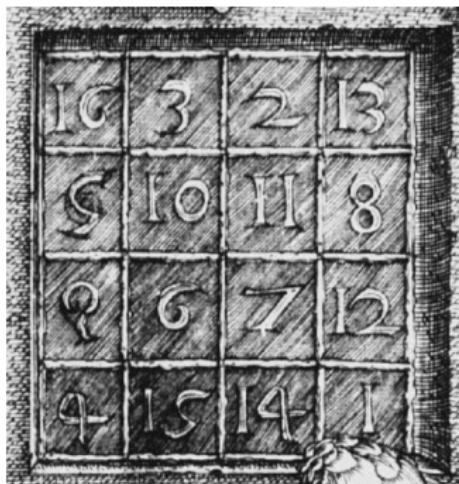
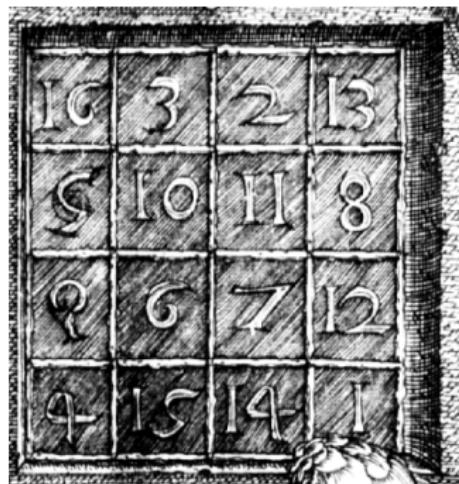
Imaginea  $B$  foloseste doar **10%** din memoria imaginii  $A$ :

$$20 \times 359 + 20 \times 371 + 20 = 14.620 \quad \text{vs.} \quad 359 \times 371 = 133.189 \text{ pixeli}$$

## Compresia imaginilor

Putem recupera o imagine fidela cu doar **200** de componente principale, in loc de 359 cat are imaginea originala (i.e. 50% informatie din imaginea originala):

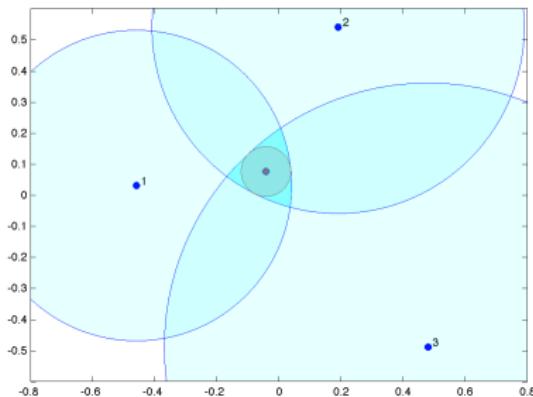
$$[U, S, V] = \text{svd}(A) \text{ si } B = U(:, 1 : 200) S(1 : 200, 1 : 200) V(:, 1 : 200)$$



## Localizarea

- se dau 3 ancore (antene) a caror pozitie este cunoscută:  
 $x_1, x_2, x_3 \in \mathbb{R}^2$  (sau  $\mathbb{R}^3$ )
- se cunosc distanțele de la un obiect (e.g. avion) necunoscut la cele 3 ancore:  $R_1, R_2, R_3 > 0$

**Formularea problemei:** estimarea pozitiei obiectului  $x$  si estimarea incertitudinii (zona)



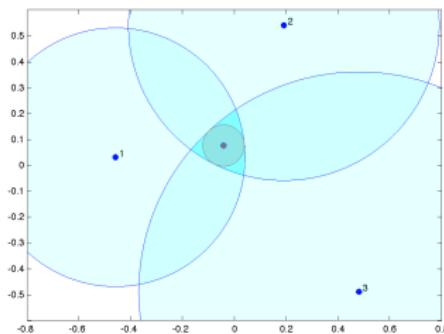
## Localizarea

- pozitiile ancorelor  $x_1, x_2, x_3$  cunoscute, dar distantele la obiectul necunoscut  $R_1, R_2, R_3$  nu se stiu exact:

$$\min_{x \in \mathbb{R}^3, \delta \in \mathbb{R}^3} \|\delta\|_p$$

$$\text{s.t.: } \|x - x_i\| \leq R_i + \delta_i \quad \forall i = 1, 2, 3$$

unde norma  $p = 1$  sau  $p = 2$  sau  $p = \infty$ .

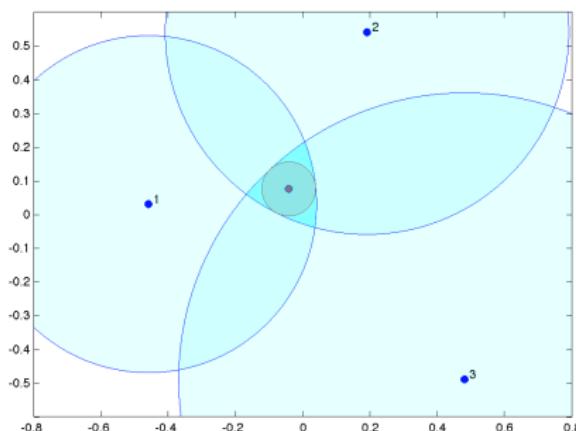


## Localizarea

- pozitiile ancorelor  $x_1, x_2, x_3$  si distantele la obiectul necunoscut  $R_1, R_2, R_3$  cunoscute  $\Rightarrow$  estimarea zonei unde se gaseste obiectul  
*o sfera de centru  $x$  si raza  $R$  se gaseste intr-o sfera de centru  $x_i$  si raza  $R_i$  daca diferențele intre raze depaseste distantele dintre centre:  $\|x - x_i\| \leq R_i - R$*

$$\max_{x,R} R$$

$$\text{s.t.: } \|x - x_i\| + R \leq R_i \quad \forall i = 1, 2, 3$$

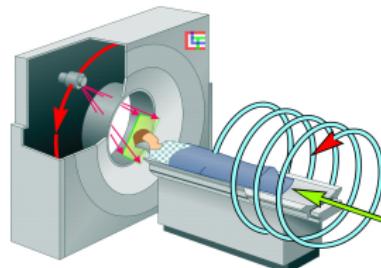


# Reconstructie tomografica

**Tomografie computerizata** = tehnica **noninvaziva** ce foloseste raze X (sau alte tipuri de radiatii) pentru a produce imagini 2D/3D ale interiorului obiectului scanat.

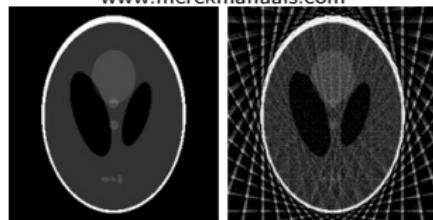
Procedura de functionare consta in:

1. Se achizitioneaza o serie de **proiectii**, din diferite unghiuri, ale obiectului scanat;



[www.merckmanuals.com](http://www.merckmanuals.com)

2. Prin intermediul proiectiilor obtinute, se reconstruiese interiorul obiectului cu ajutorul unui algoritm iterativ;



[www.mathworks.com](http://www.mathworks.com)

In majoritatea cazurilor, radiatiile folosite sunt daunatoare; de aceea se urmareste achizitionarea unui numar minim de proiectii.

## Reconstructie tomografica

Formularea problemei:

- Fie  $x \in \mathbb{R}^n$  imaginea interiorului de reconstructuit.
- Pentru reconstructie, disponem de diferite masuratori liniare (proiectii) ale imaginii  $x$ :  $b_i = A_i x$ ,  $i = 1, \dots, m$ .
- Notam vectorul proiectiilor  $b \in \mathbb{R}^m$  si  $A = [A_1^T \cdots A_m^T]^T \in \mathbb{R}^{m \times n}$  matricea de achizitie.
- Imaginea interiorului reprezinta solutia sistemului liniar (subdeterminat deoarece sunt mai putine masuratori  $m$  decat dimensiunea imaginii  $n$ ):  $\mathbf{Ax} = \mathbf{b}$ .
- Reformulare in termeni de problema CMMP:

$$\min_{x \in \mathbb{R}^n: Ax=b} \|x\|_\alpha$$

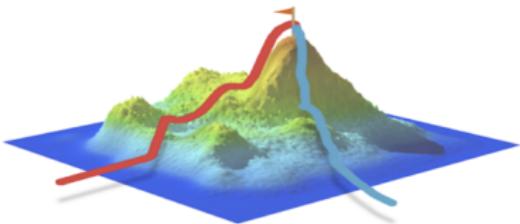
unde de obicei se alege  $\alpha = 2$  sau  $\alpha = 0$  sau  $\alpha = 1$ . Alegem  $\alpha = 0 \vee 1$  pentru a induce o reprezentare rara a imaginii (vectorul solutie). Se doreste o reprezentare rara a imaginii deoarece aceasta permite: compresie usoara; algoritmi rapizi pt. procesare; memorie de stocare mica; eliminarea usoara a zgromotului;...

# Optimizarea matematica

$$f^* = \min_{x \in \mathbb{R}^n} f(x)$$

$$\text{s.l.: } g_1(x) \leq 0, \dots, g_m(x) \leq 0$$

$$h_1(x) = 0, \dots, h_p(x) = 0.$$



- ▶  $x$  variabila de decizie
- ▶ functia obiectiv  $f: \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$
- ▶ constrangeri de inegalitate si egalitate  $g_i, h_j: \mathbb{R}^n \rightarrow \mathbb{R}$

In **solutia optima**  $x^*$  functia obiectiv ia valoarea cea mai mica (**valoarea optima**  $f^*$ ) in raport cu toti vectorii ce satisfac constrangerile (restrictiile)

## Scurta istorie:

- ▶ Antichitate: abordare geometrica (aplicatii izoperimetrice - Problema Dido sec. IX i.H.)
- ▶ Evul mediu: abordare algebraica (folosind e.g. inegalitatea mediilor)
- ▶ Dupa sec XIX: abordare pe baza calculului diferential

## Scurta istorie

- **teorie:** conditii de optimalitate, analiza convexa (1900-1970)
- **algoritmi:**
  - ▶ 1947: algoritmul simplex pentru programare liniara (Dantzig)
  - ▶ 1950: programarea dinamica (Bellman '52)
  - ▶ 1960: metoda gradientilor conjugati (Fletcher & Reeves '64); metode quasi-Newton (David-Fletcher-Powel '59)
  - ▶ 1970: metode de penalitate si bariera (Fiacco & McCormick); programare patratica secentiala; probleme de mari dimensiuni (Lasdon '70); metoda subgradient (scoala ruseasca)
  - ▶ 1980-2000: metode de punct interior cu complexitate polinomiala (Karmakar '84; Nesterov & Nemirovski '94)
- **aplicatii:**
  - ▶ inainte de 1990: cercetari operationale (finante, logistica, manufacturing, scheduling)
  - ▶ dupa 1990: aplicatii ingineresti (control, procesare de semnal, motoare de cautare, retele); medicina (tomografie); biologie (recunoasterea de gene); fizica (predictia vremii)

# Notatii

In cadrul cursului vom utiliza urmatoarele notatii:

- Vectori (considerati intotdeauna vector coloana) cu litere mici,

i.e.  $x \in \mathbb{R}^n$ ,  $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

- Produs scalar in spatiul Euclidian:  $\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i$
- Norma Euclidiana standard  $\|x\| = \sqrt{\langle x, x \rangle} = \sqrt{x_1^2 + \dots + x_n^2}$
- Multimi cu litere mari:  $S, Q, U \subseteq \mathbb{R}^n$  ( $\mathbb{R}_+^n$  - orthantul nenegativ,  $S_+^n$  - multimea matricelor pozitiv semidefinite)
- Matrice cu litere mari:  $A, B, C, H \in \mathbb{R}^{n \times m}$
- Norma spectrala a unei matrici  $\|A\| = \sqrt{\lambda_{max}(A^T A)}$
- Matrice pozitiv definita:  $A \succ 0$ , si pozitiv semidefinita  $A \succeq 0$

## Notatii

In cadrul cursului vom utiliza urmatoarele notatii:

- Functii cu litere mici:  $f, g, h: \mathbb{R}^n \rightarrow \mathbb{R}$ .
- Gradientul, respectiv matricea Hessiana a functiei continuu differentiabile  $f: \mathbb{R}^n \rightarrow \mathbb{R}$ :  $\nabla f(x) \in \mathbb{R}^n$ ,  $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$ .

Hessiana este matrice simetrica (i.e. in  $S^n$ )!

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}, \quad \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial^2 x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial^2 x_n} \end{bmatrix},$$

**Teorema de medie:** fie functia  $g: \mathbb{R} \rightarrow \mathbb{R}$  atunci exista  $\theta \in [a, b]$  a.i.  $g(b) - g(a) = g'(\theta)(b - a) = \int_a^b g'(\tau) d\tau \implies$  pentru  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  luam  $g(t) = f(x + t(y - x))$ :

$$f(y) = f(x) + \langle \nabla f(x + \theta(y - x)), y - x \rangle \quad \theta \in [0, 1]$$

$$f(y) = f(x) + \int_0^1 \langle \nabla f(x + \tau(y - x)), y - x \rangle d\tau$$

# Probleme de optimizare

Problema de optimizare neliniara:

$$f^* = \min_{x \in \mathbb{R}^n} f(x)$$

$$\text{s.l.: } g_1(x) \leq 0, \dots, g_m(x) \leq 0$$

$$h_1(x) = 0, \dots, h_p(x) = 0.$$

**O problema de optimizare** contine urmatoarele trei componente:

- (i) o functie obiectiv,  $f(x)$ , ce va fi minimizata/maximizata;
- (ii) variabile de decizie,  $x \in \mathbb{R}^n$ , care se pot alege dintr-o anumita multime;
- (iii) constrangeri (numite si restrictii) ce vor fi respectate, de forma  $g(x) \leq 0$  (constrangeri de inegalitate) si/sau  $h(x) = 0$  (constrangeri de egalitate).

## Probleme de optimizare

**Formularea standard** a unei probleme de optimizare:

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$\text{s.l.: } g_1(x) \leq 0, \dots, g_m(x) \leq 0$$

$$h_1(x) = 0, \dots, h_p(x) = 0.$$

Functia obiectiv  $f: \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$  si variabila de decizie  $x \in \mathbb{R}^n$

Notatii compacte:

$$g: \mathbb{R}^n \rightarrow \mathbb{R}^m, g(x) = [g_1(x) \dots g_m(x)]^T$$

$$h: \mathbb{R}^n \rightarrow \mathbb{R}^p, [h_1(x) \dots h_p(x)]^T$$

$\Rightarrow$  forma compacta a problemei de *optimizare neliniara* ((NLP) - nonlinear programming)

$$(NLP): \quad \min_{x \in \mathbb{R}^n} f(x)$$

$$\text{s.l.: } g(x) \leq 0, h(x) = 0.$$

## Probleme de optimizare

**Remarca:** o problema de maximizare poate fi transformata intr-o de minimizare,  $\max_x f(x) = -\min_x -f(x)$ .

Exemplu problema de optimizare:

$$\begin{aligned} & \min_{x \in \mathbb{R}^2} 2x_1^3 + 3x_2^2 + 4x_1 \\ \text{s.l.: } & x \geq 0, \quad x_1^2 + x_2^2 - 1 \leq 0 \\ & x_1 x_2 - 1 = 0. \end{aligned}$$

In acest exemplu avem:

- ▶ variabila de decizie:  $x = [x_1 \ x_2]^T \in \mathbb{R}^2$  si functia obiectiv  $f(x) = 2x_1^3 + 3x_2^2 + 4x_1$
- ▶ trei constrangeri de inegalitate:  $g : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ , unde  $g_1(x) = -x_1$ ,  $g_2(x) = -x_2$  si  $g_3(x) = x_1^2 + x_2^2 - 1$ .
- ▶ o singura constrangere de egalitate definita de functia  $h(x) = x_1 x_2 - 1$ .

# Probleme de optimizare

Alte definitii esentiale:

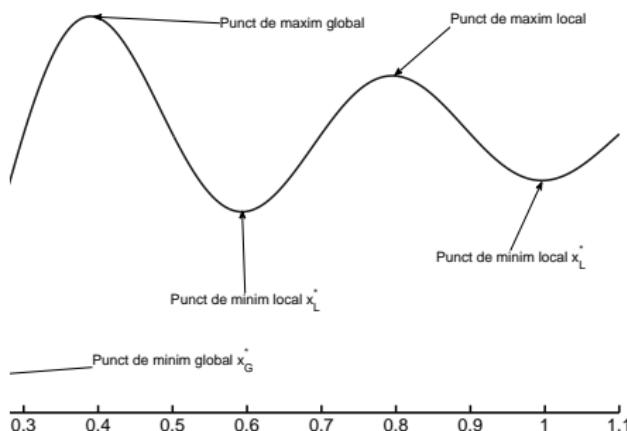
- ▶ Multimea  $\{x \in \mathbb{R}^n : f(x) = c\}$  este *multimea nivel* (conturul) a functiei  $f$  pentru valoarea  $c \in \mathbb{R}$ .
- ▶ *Multimea fezabila* a problemei de optimizare (NLP) este:

$$X = \{x \in \mathbb{R}^n : g(x) \leq 0, h(x) = 0\}.$$

- ▶ Punctul  $x^* \in \mathbb{R}^n$  este un *punct de minim global* (adeseara denumit *minim global*) daca si numai daca  $x^* \in X$  si  $f(x^*) \leq f(x)$  oricare ar fi  $x \in X$ .
- ▶ Punctul  $x^* \in \mathbb{R}^n$  este *minim local* daca si numai daca  $x^* \in X$ . si exista o vecinatate  $\mathcal{N}$  a lui  $x^*$  (e.g. o bila deschisa cu centrul in  $x^*$ ) astfel incat  $f(x^*) \leq f(x)$  oricare ar fi  $x \in X \cap \mathcal{N}$ .

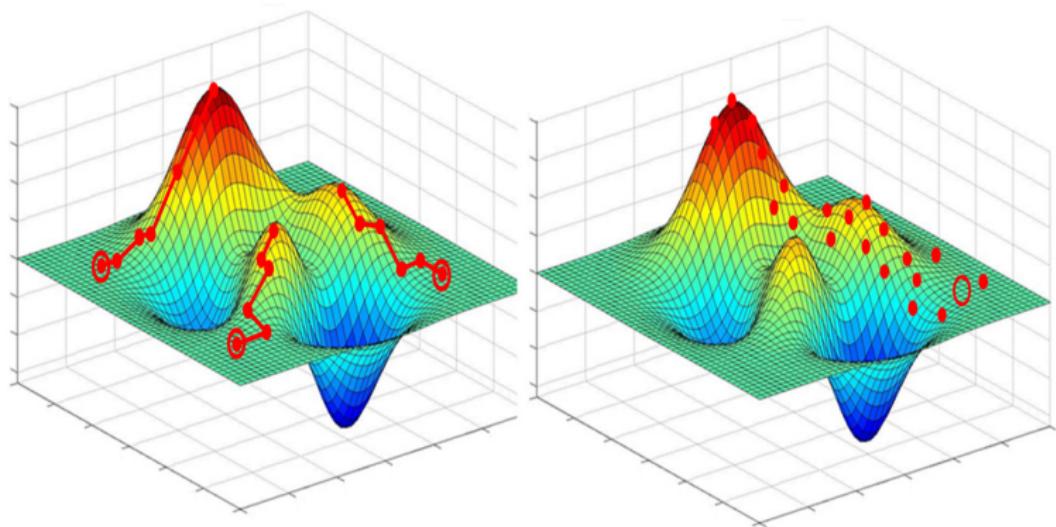
## Minime-maxime locale/globale

Exemplu: puncte de minim local ( $x_L^*$ ) și punctul de minim global ( $x_G^*$ ) pentru  $f(x) = \cos(5\pi x)/x$  în intervalul  $[0.1, 1.1]$ .



- ▶ Punctul  $x^* \in \mathbb{R}^n$  este un *punct strict de minim global* daca si numai daca  $x^* \in X$  si  $f(x^*) < f(x)$  oricare ar fi  $x \in X \setminus \{x^*\}$ .
- ▶ Punctul  $x^* \in \mathbb{R}^n$  este un *punct strict de minim local* daca si numai daca  $x^* \in X$  si exista o vecinatate  $\mathcal{N}$  a lui  $x^*$  astfel încât  $f(x^*) < f(x)$  oricare ar fi  $x \in (X \cap \mathcal{N}) \setminus \{x^*\}$ .

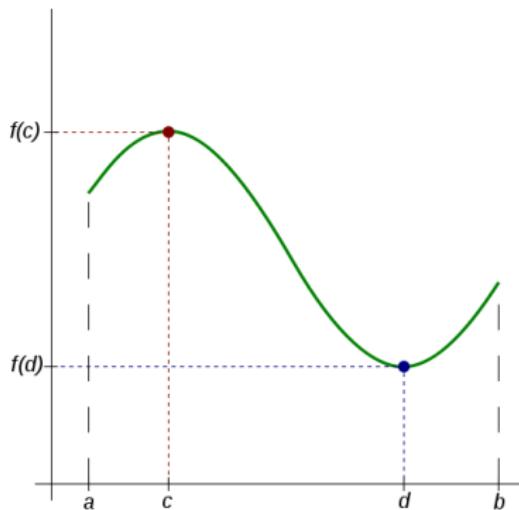
## Minime-maxime locale/globale



## Minime-maxime locale/globale

**Existenta punctelor de minim** (teorema lui Weierstrass): daca multimea fezabila  $X \subset \mathbb{R}^n$  este compacta (adica marginita si inchisa) si  $f : X \rightarrow \mathbb{R}$  este continua atunci exista un punct de minim global pentru problema de minimizare  $\min_{x \in X} f(x)$ .

Exemplu: pentru functia  $f(x)$  din figura consideram multimea fezabila compacta  $X = \{x : a \leq x \leq b\}$  si observam ca are punct de minim global in  $d$ .



## Probleme de optimizare - NLP

**Categorii** diferite de probleme de optimizare si exemple.

Reamintim categoria cea mai generală, **problema de optimizare neliniară** (NLP):

$$(NLP) : \min_{x \in \mathbb{R}^n} f(x)$$

$$\text{s.l.: } g(x) \leq 0, \quad h(x) = 0,$$

In mod obisnuit,  $f(x)$ ,  $g(x)$  si  $h(x)$  se presupun a fi continuu differentiabile cel putin o data. NLP rezolvate cu algoritmi predati in acest curs!

Exemplu:

$$\min_{x \in \mathbb{R}^3} x_1^6 + x_2^4 + x_3$$

$$\text{s.l.: } 5x_1 + 6x_2 \leq 0, \quad x_1 + x_3 = 4, \quad x_1 + x_2 = -2$$

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}, \quad f(x) = x_1^6 + x_2^4 + x_3$$

$$g : \mathbb{R}^3 \rightarrow \mathbb{R}, \quad g(x) = 5x_1 + 6x_2$$

$$h : \mathbb{R}^3 \rightarrow \mathbb{R}^2, \quad h(x) = \begin{bmatrix} x_1 + x_3 - 4 \\ x_1 + x_2 + 2 \end{bmatrix}$$

## Probleme de optimizare LP

Daca  $f$ ,  $g$  si  $h$  din problema NLP sunt afine:

$$f(x) = c^T x,$$

$$g(x) = Cx - d, h(x) = Ax - b,$$

atunci problema NLP devine o problema de programare liniara (LP sau linear programming):

$$(LP) : \min_{x \in \mathbb{R}^n} c^T x$$

$$\text{s.l.: } Cx - d \leq 0, Ax - b = 0,$$

LP - urile pot fi rezolvate eficient, cu teorie dezvoltata incepand cu anii 1940 (metoda simplex). Astazi se pot rezolva probleme LP cu  $10^9$  variabile!

Aplicatii in : **economie** (optimizare de portofoliu); **assignment** (companie aeriana numeste echipajul fiecarui zbor a.i.: fiecare zbor este acoperit)



## Probleme de optimizare - LP

Exemplu de problema LP:

$$\min_{x \in \mathbb{R}^3} 3x_1 + 2x_2 + 4x_3 \quad (1)$$

$$\text{s.l.: } 0 \leq x_1 \leq 10, x_2 \geq 0, x_3 \geq 0$$

$$x_1 + x_2 = 2$$

$$x_2 + x_3 = 1$$

Observam:  $f(x) = 3x_1 + 2x_2 + 4x_3 = \underbrace{\begin{bmatrix} 3 & 2 & 4 \end{bmatrix}}_{c^T} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_x = c^T x$

Constrangeri de egalitate:  $h(x) : \mathbb{R}^3 \rightarrow \mathbb{R}^2 \quad h(x) = Ax - b = 0$

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_x - \underbrace{\begin{bmatrix} 2 \\ 1 \end{bmatrix}}_b = 0$$

## Probleme de optimizare - LP

Constrangeri de inegalitate:  $g(x) : \mathbb{R}^3 \rightarrow \mathbb{R}^4$ ,  $g(x) = Cx - d \leq 0$

Inegalitatile  $0 \leq x_1 \leq 10, x_2 \geq 0, x_3 \geq 0$  pot fi scrise drept:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}}_C \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_x - \underbrace{\begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_d \leq 0$$

Astfel, problema (1) poate fi scrisa ca problema LP standard.

## Probleme de optimizare - QP

Daca  $g(x)$  si  $h(x)$  din problema NLP sunt afine iar functia obiectiv  $f(x)$  este o functie patratica, atunci problema NLP devine o problema de **programare patratica** (QP sau quadratic programming):

$$(QP) : \min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + q^T x + r$$
$$\text{s.l.: } Cx - d \leq 0, \quad Ax - b = 0.$$

Conditii de convexitate pentru functii de doua ori continuu differentiabile: o functie este convexa daca  $\nabla^2 f(x) \succeq 0$ . Pentru o problema QP, daca  $Q = Q^T$ , atunci  $\nabla^2 f(x) = Q$ . In acest caz, functia obiectiv este convexa daca  $Q \succeq 0$ .

QP-urile se rezolva cu metode punct interior, "active set" sau de tip gradient: astazi se pot rezolva probleme de dimensiuni foarte mari ( $10^9$  variabile) ce apar in multe aplicatii: **optimizarea motoarelor de cautare, procesarea de imagine, predictie**

## Probleme de optimizare - QP

Exemplu de problema QP:

$$\min_{x \in \mathbb{R}^n} x_1^2 + x_2^2 + x_1 x_2 + 3x_1 + 2x_2$$

$$\text{s.l.: } x \geq 0, x_1 + x_2 = 5$$

Observam

$$\begin{aligned} f(x) &= x_1^2 + x_2^2 + x_1 x_2 + 3x_1 + 2x_2 \\ &= \frac{1}{2} [x_1 \quad x_2] \underbrace{\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}}_Q \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 3 & 2 \end{bmatrix}}_{q^T} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned}$$

Constrangeri de inegalitate:  $-I_2 x \leq 0$

Constrangeri de egalitate:  $[1 \quad 1]^T x = 5$

Calcul simple:  $\Lambda(Q) = \{3, 1\}$ , deci  $Q \succ 0$  iar  $f(x)$  este convexă

- Ierarhizarea în motoarele de căutare se pune ca o problemă QP

$$\min_{x \in \mathbb{R}^n: e^T x = 1, x \geq 0} \|Ex - x\|^2$$

$E$  matrice de adiacență

## Probleme de optimizare - CP

### Probleme de optimizare convexa:

$$(CP) : \min_{x \in \mathbb{R}^n} f(x)$$
$$\text{s.l.: } g(x) \leq 0, \quad Ax - b = 0,$$

O problema de optimizare este convexa daca functia obiectiv  $f(x)$  si multimea fezabila sunt convexe.

Daca  $g(x)$  convexa si  $h(x)$  afina (i.e.  $h(x) = Ax - b$ ), atunci multimea fezabila  $X = \{x : g(x) \leq 0, Ax = b\}$  este convexa.

- Observatie: problemele LP si problemele QP (cu  $Q \succeq 0$ ) sunt probleme de optimizare convexe. Problemele convexe sunt, in general, mai usor de rezolvat decat (NLP)-urile. Foarte multe aplicatii: **procesare de semnal, control optimal, design optim**
- Observatie: o problema de optimizare convexa unde  $f(x) = \frac{1}{2}x^T Qx + q^T x$  si  $g_i(x) = \frac{1}{2}x^T Q_i x + q_i^T x + r_i$ , i.e.  $f$  si functiile din constrangerile de inegalitate sunt patratice, se numeste QCQP(*quadratically constrained quadratic program*)