INTELLIGENTNI SISTEMI

OSNOVNI KONCEPTI STROJNEGA UČENJA

Osnovni koncepti strojnega učenja

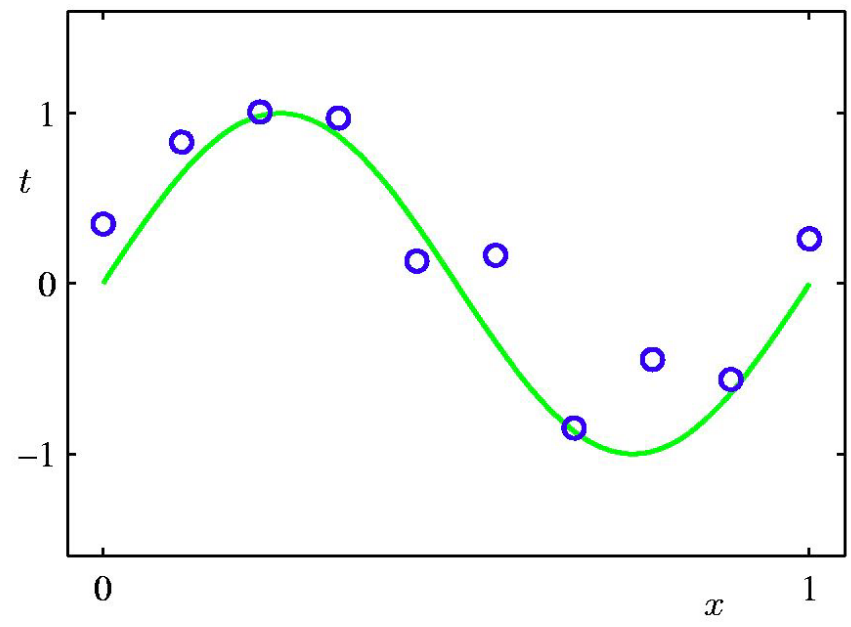
Prof. Jurij F. Tasič
Emil Plesnik

Uvod v strojno učenje: Teme

1. Polinomsko prilagajanje krivulj
2. Verjetnostna teorija več spremenljivk
3. Največja verjetnost
4. Bayesov pristop
5. Izbira modela
6. Prekletstvo dimenzionalnosti

Polinomsko prilagajanje krivulj

- Imamo realno vhodno spremenljivko *x*
- *x* uporabimo za predikcijo vrednosti ciljne spremenljivke *t*
- Sintetiziramo podatke s funkcijo *sin(2π x)*
- Pri ciljnih vrednostih je prisoten naključni šum



Polinomsko prilagajanje krivulj

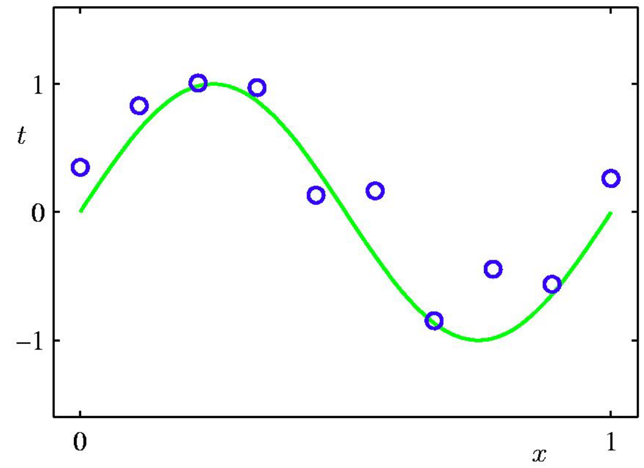
Zapišemo:

- N opazovanj *x*

x = (x1,..,xN)T

t = (t1,..,tN)T

- Cilj je izkoristiti učno množico za predikcijo vrednosti iz *x*
- Zahteven problem
- Teorija verjetnosti dovoljuje predikcijo



Sinteza podatkov:

N = 10

Enakomerna porazdeljenost na intervalu [0,1]

Ustvarjeno s funkcijo sin(2πx) z dodanim Gaussovim šumom

Polinomsko prilagajanje krivulj

- Polinomska funkcija

$$y(x,\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + .. + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

- M je red polinoma
- Ali je večja vrednost M boljša?
- Koeficienti w_0 ,…w_M sestavljajo vektor w
- Nelinearna funkcija x, linearna funkcija koeficientov w
- Klicani linearni modeli

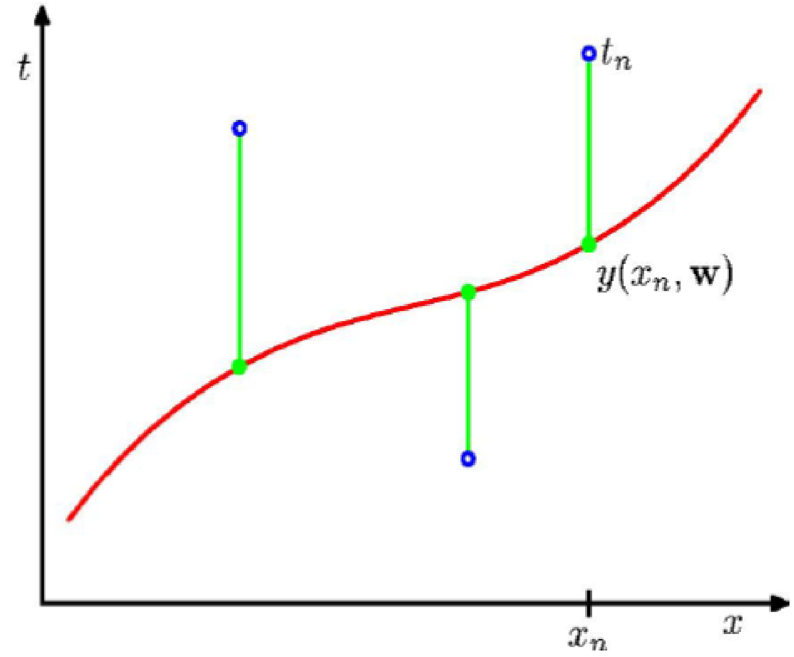
Funkcija napake

Vsota kvadratov napak med predikcijo $y(x_n,\mathbf{w})$ za Vsako točko x_n in ciljno vrednostjo t_n

$$E(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathrm{w}) - t_n\}^2$$

- Rešitev je vrednost w, pri kateri je E(w) najmanjša

Rdeča črta pomeni najboljše polinomsko ujemanje



Minimizacija funkcije napake

Funkcija napake je kvadratična glede na koeficiente w

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2$$

- Odvod po koeficientih pomeni

Linearno funkcijo glede na *w*

ker je $y(x,\mathrm{w}) = \sum_{j=0}^{M} w_j x^j$

- Funkcija napake ima torej

Minimum **w***

- Rezultat je polinom *y(x,w*)*

$$\frac{\partial E(\mathrm{w})}{\partial w_i} = \sum_{n=1}^{N}\{y(x_n,\mathrm{w}) - t_n\}x_n^i$$

$$= \sum_{n=1}^{N}\{\sum_{j=0}^{M}\mathrm{w}_j x_n^j - t_n\}x_n^i$$

postavimo izraz na nič

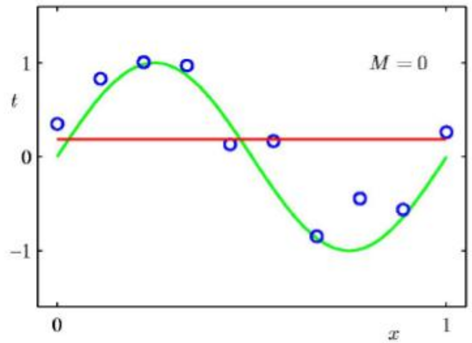
$$\sum_{n=1}^{N}\sum_{j=0}^{M} w_j x_n^{i+j} = \sum_{n=1}^{N} t_n x_n^i$$

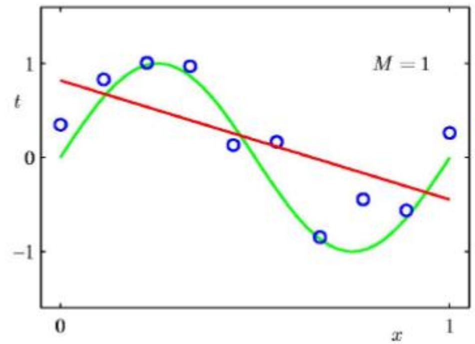
Dobimo M+1 enačb, ki jih rešimo glede na **w***

Izbira stopnje M

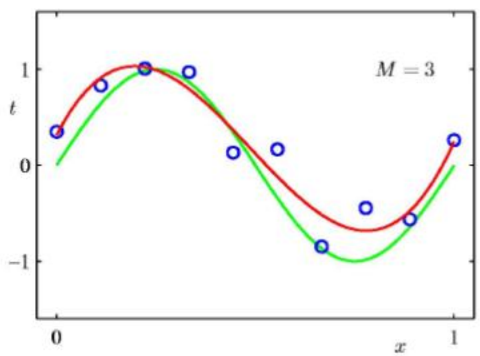
Primerjava modelov ali izbira modelov

- Rdeče črte so najboljše prilagoditve za
  - *M = 0,1,3,9 in N=10*

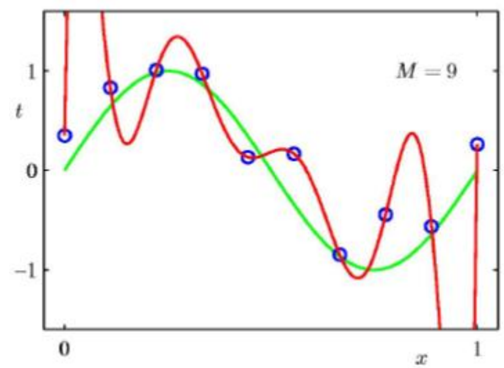




Slaba predstavitev sin(2πx)



dobra predstavitev (2πx)



Najmanjši kvadrat je primer maksimalne verjetnosti

- Omejitev števila parametrov na velikost učne množice ni primerna
- Bolje je izbrati kompleksnost modela glede na kmpleksnost problema
- Pristop najmanjših kvadratov je poseben primer maksimalne verjetnosti
    - Lastnost maksimalne verjetnosti je prevelika prilagoditev (Over-fitting)
- Z Bayesovim pristopom se izognemo preveliki prilagoditvi
    - Število parametrov lahko močno preseže število podatkovnih točk
    - Efektivno število parametrov se avtomatsko prilagodi velikosti podatkovne množice

Ureditev metode najmanjših kvadratov

- uporaba kompleksnih modelov z omejenimi velikostmi podatkovnih množic
- dodajanje omejitve v funkcijo napake, ki preprečuje koeficientom doseganje velikih vrednosti

- λ določa relativno pomembnost ureditvenega izraza za funkcijo napake
- Možna je natančna minimizacija v zaprti obliki
- Poznano kot skrčitev v statistiki

Osnove adaptivnih sistemov

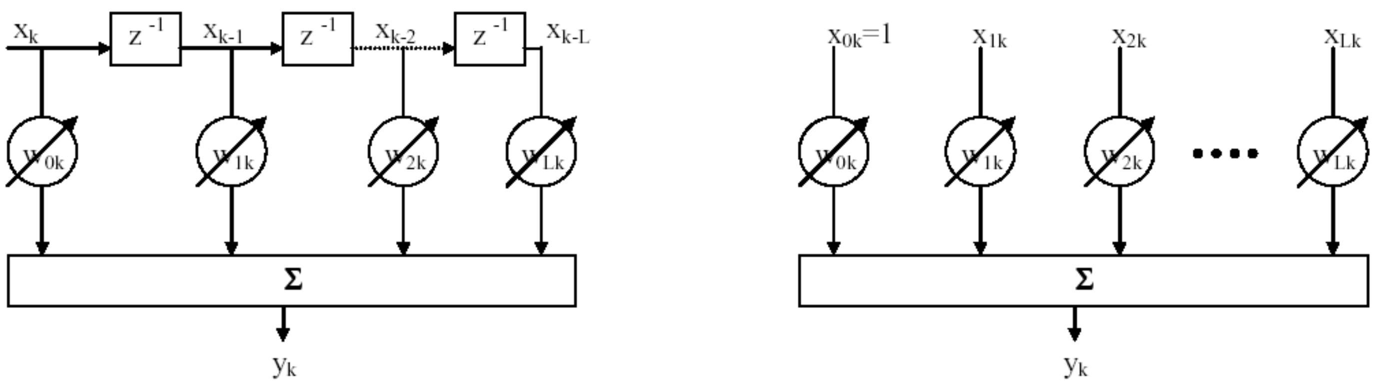
- Linearno adaptivno sito ali nerekurzivno adaptivno sito je eden osnovnih elementov v adaptivnem procesiranju diskretnih signalov.
- Predpostavimo torej diskreten vhodni signal, ki ima vec komponent. Tak vhodni signal lahko zapišemo v obliki vektorja **X**. Poleg tega predpostavimo tudi vektor uteži **W**, s katerim dejansko vplivamo na komponente vhodnega signala,

Enovhodni primer:

$$X_k = [x_k, x_{k-1} \ldots x_{k-L}]^T \tag{2.1}$$

Večvhodni primer:

$$X_k = [x_{0k}, x_{1k} \ldots x_{Lk}]^T \tag{2.2}$$



Osnove adaptivnih sistemov

Vektor uteži lahko podobno kot vhodna vektorja zapišemo z izrazom

$$W_{k=[0k,w_{1k}\dots w_{Lk}]^T}$$

Odnos med vhodom in izhodom lahko torej zapišemo za oba primera izrazimo z

Enovhodni primer:

$$y_k = \sum_{l=1}^{L} W_{1k} \cdot x_{k-1}$$

Večvhodni primer:

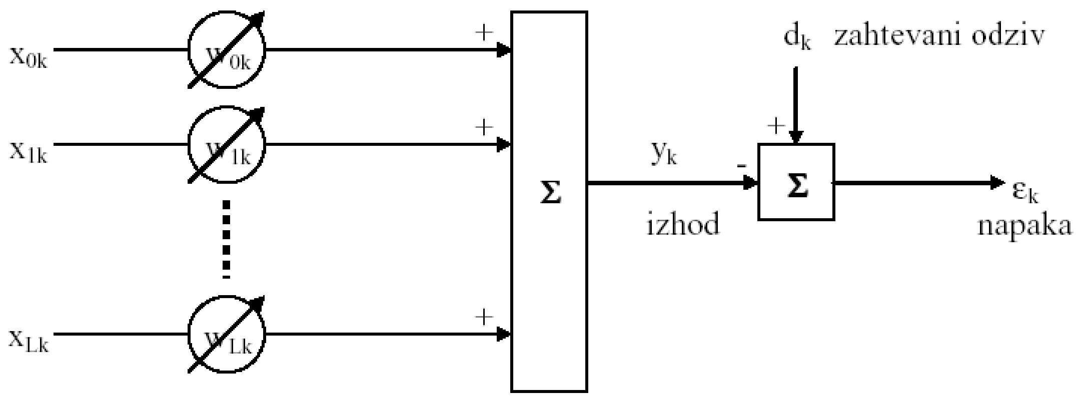
$$y_k = \sum_{l=1}^{L} W_{1k} \cdot x_{1k}$$

Za oba primera pa lahko odnos med vhodom in izhodom zapišemo tudi z izrazom v vektorski obliki, ki je enaka za oba primera:

$$y_k = X_k^T W_k = W_k^T X_k$$

Večvhodno adaptivno linearno sito

Zapišimo z izrazom napako, ki jo dobimo s principielnim vezjem:



$$\varepsilon_k = d_k - -y_k = d_k - X_k^T W = d_k - W^T X_k$$

$$\varepsilon_k^2 = d_k^2 + W^T X_k X_k^T W - 2 d_k X_k^T W$$

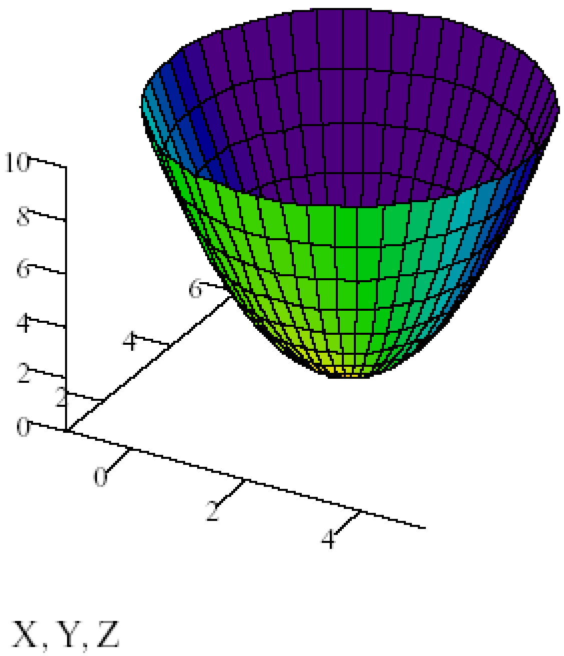
$$R = E\left[X_k X_k\right]^T = E\begin{bmatrix} X_{0k}^2 & X_{0k}X_{1k} & X_{0k}X_{2k}\cdots & X_{0k}X_{Lk} \\ X_{1k}X_{0k} & X_{1k}^2 & X_{1k}X_{2k}\cdots & X_{0k}X_{Lk} \\ \vdots & & & \\ X_{Lk}X_{0k} & X_{Lk}X_{1k} & X_{Lk}X_{2k} & X_{Lk}^2 \end{bmatrix}$$

$$P = E[d_k X_k] = [d_k x_{0k}, d_k x_{1k} \ldots d_k x_{Lk}]^T$$

KVADRATIČNA NAPAKA

Srednjo kvadratic:no napako izrazimo kot

$$MSE = \xi = E[\varepsilon^2] = E[d_k^2] + \mathbf{W}^T\mathbf{R}\mathbf{W} - 2\mathbf{P}^T\mathbf{W}$$



Površina paraboloida kvadrata napake

UVOD V OPTIMIZACIJSKE METODE

- terminologija in predpostavke
- metoda manjšanja gradienta
- metoda največje strmine gradienta
- Newton-ova metoda
- samousklajene funkcije
- implementacija

Optimizacija je matematična disciplina, ki se ukvarja z iskanjem minimumov in maksimumov funkcij glede na dane omejitve.

SESTAVNI DELI

- Kriterijska funkcija
- Spremenljivke
- Omejitve

Cilj je poiskati takšne vrednosti spremenljivk, ki minimizirajo ali maksimizirajo kriterijsko funkcijo, upoštevajoč omejitve.

Odločanje

Primeri:

- Odločanje, katere sestavine in v kakšnih količinah dodati, da bo dobljena zmes ustrezala specifikacijam sestave
- Dodeliti razpoložljiva sredstva med različne prejemnike/agencije
- Odločanje, katero pot ubrati do neke nove lokacije

- Odločanje vedno zajema odločitev med različnimi razpoložljivimi možnostmi.

Optimizacijski modeli

- En sam x Večobjektni modeli
- Statični x Dinamični modeli
- Deterministični x Stohastični modeli

Neomejena minimizacija

- terminologija in predpostavke
- metoda manjšanja gradienta
- metoda največje strmine gradienta
- Newton-ova metoda
- implementacija

Recimo, da imamo stroškovno funkcijo (objective function)

$$f(\mathbf{x}) : \mathbb{R}^N \longrightarrow \mathbb{R}$$

Cilj je poiskati vrednosti parametrov (odločitvenih spremenljivk) $\mathbf{x}$, ki minimizirajo to funkcijo

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} f(\mathbf{x})$$

Upoštevamo sledeče omejitve:

enakost: $c_i(\mathbf{x}) = 0$

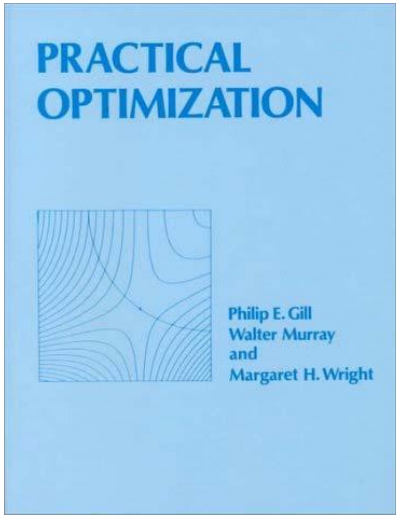
neenakost : $c_j(\mathbf{x}) \geq 0$

Če iščemo minimum funkcije $f(\mathbf{x})$ (profitna funkcija), je to ekvivalentno iskanju minimuma funkcije $-f(\mathbf{x})$

Practical Optimization

Philip E. Gill, Walter Murray, and Margaret H. Wright, Academic Press,

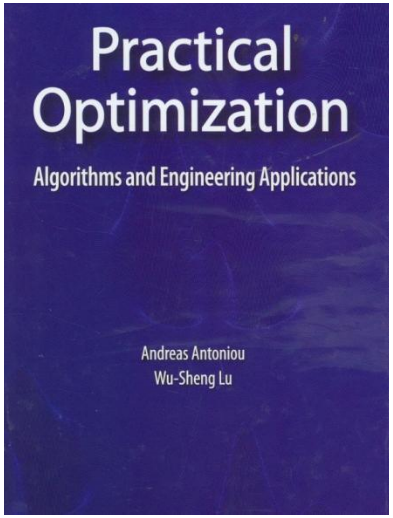
1981



Practical Optimization: Algorithms and Engineering Applications

Andreas Antoniou and Wu-Sheng Lu

2007



Numerical Recipes in C (or C++) : The Art of Scientific Computing

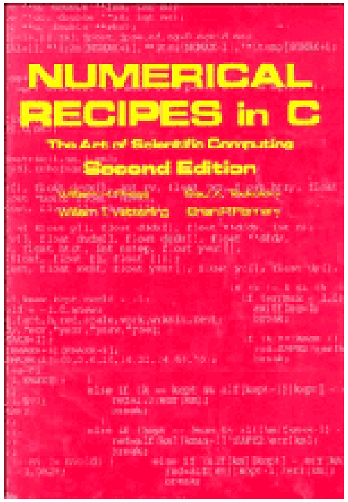
William H. Press, Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling

Good chapter on optimization

Available on line at

(1992 ed.) www.nrbook.com/a/bookcpdf.php

(2007 ed.) www.nrbook.com

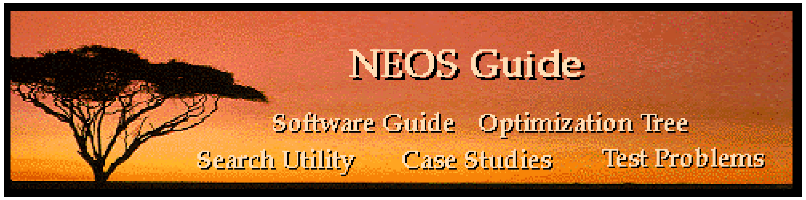


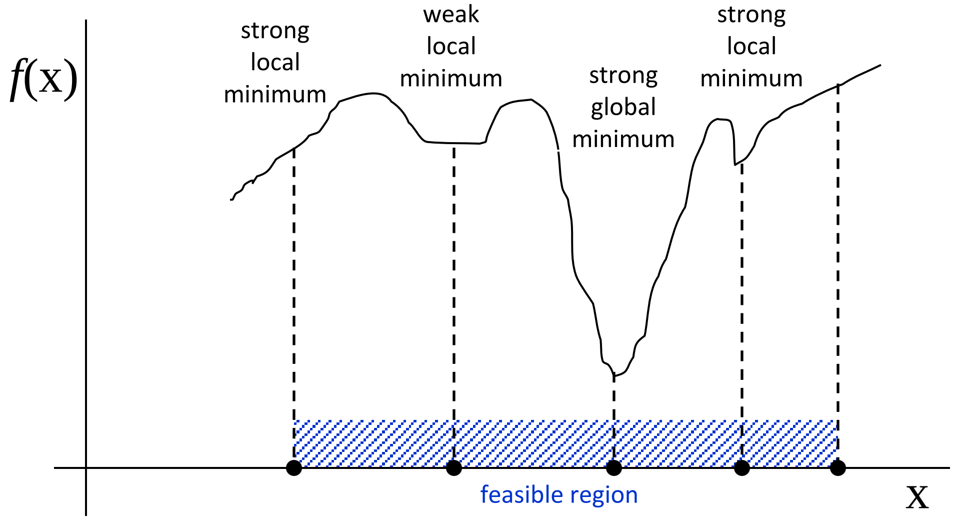
NEOS Guide

www-fp.mcs.anl.gov/OTC/Guide/

This powerpoint presentation

www.utia.cas.cz



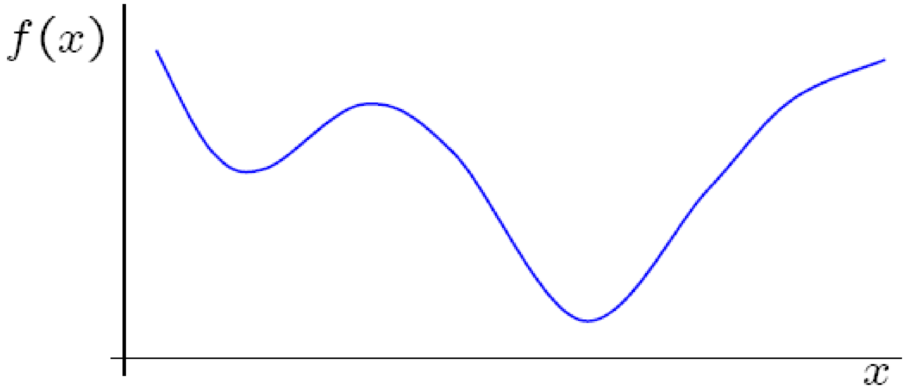


Od začetne točke je odvisno, katerega od minimumov bomo našlli

Takšen potek je pogost v realnih funkcijah

Predpostavimo, da je začnetna točka v bližini globalnega minimuma

$$\min_x f(x)$$

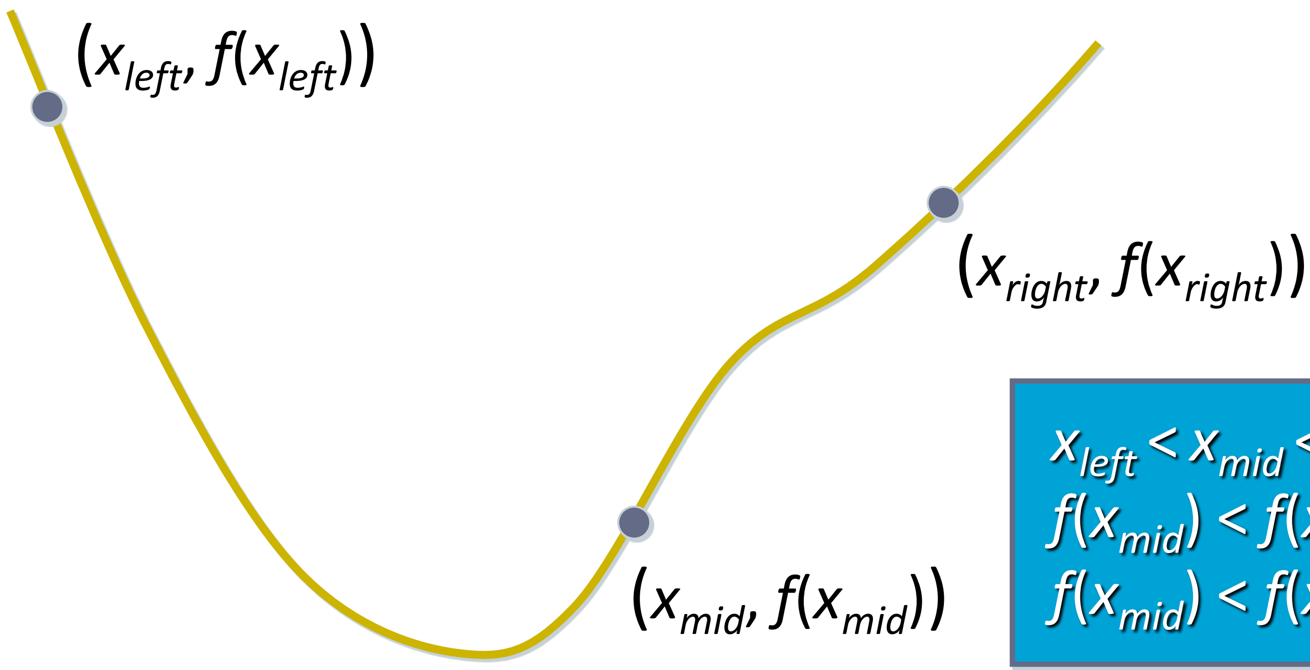


Kako določiti minimum?

- Več metod (Dichotomous, Fibonacci, Golden-Section)
- Metode aproksimacije
    1. Polinomska interpolacija
    2. Newton-ova metoda
- Kombinacija obojih (Davies, Swann, in Campey algoritmi)

Oglej si oklepaje in pomen

Kaj pomeni postaviti meji, med katerima je minimum?



$x_{left} < x_{mid} < x_{right}$

$f(x_{mid}) < f(x_{left})$

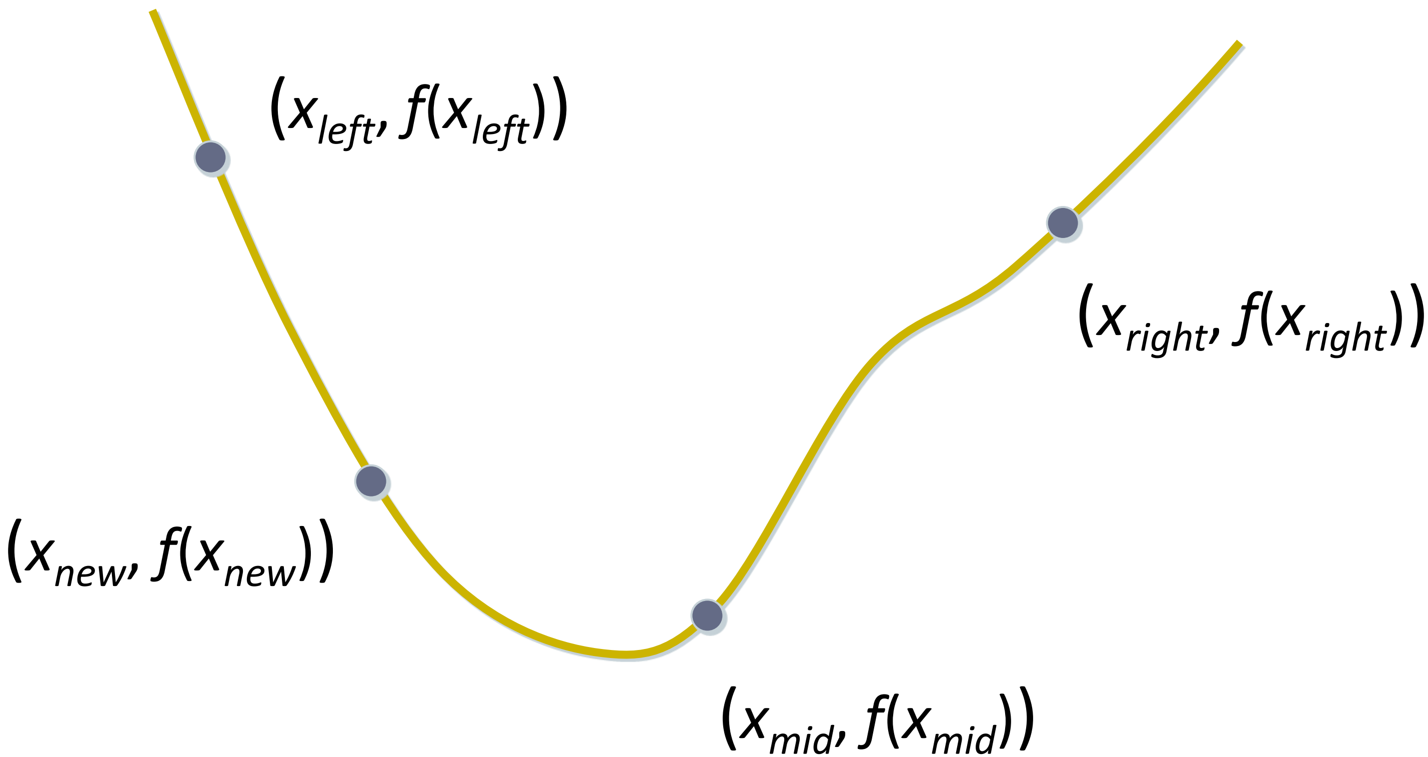
$f(x_{mid}) < f(x_{right})$

Običajno obstaja več lokalnih minimumov x_{left} and x_{right}

Postaviti obe meji pomeni:

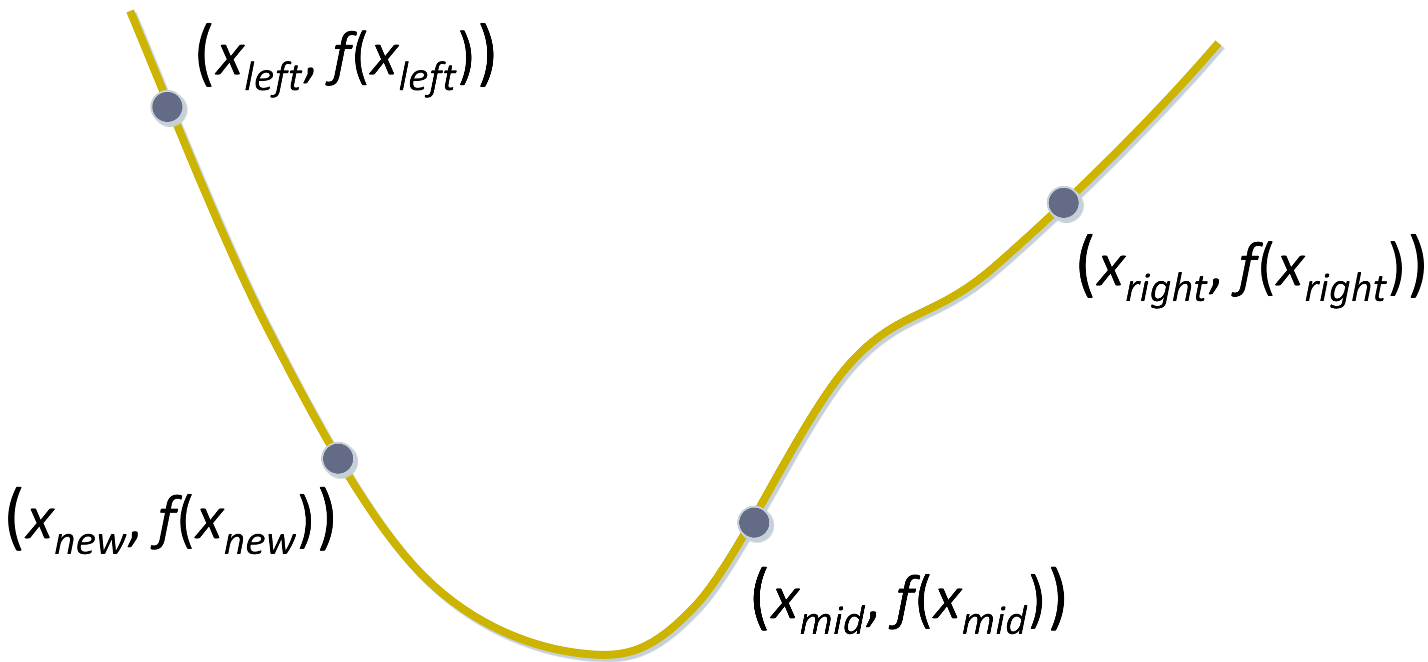
- Postavitev začetne točke $x_{initial}$, ter korak - *increment*
- Ocenimo $f(x_{initial}), f(x_{initial}+increment)$
- Zmanjšuj po korakih, dokler funkcija ne naraste,
- Napravi korak v obratni smeriz zmanjšanim korakom, dokler funkcija spet ne naraste,
- Spremeni korak po vsaki taki iteracijski zanki.

Strategija: oceni funkcijo v novem x_{new}



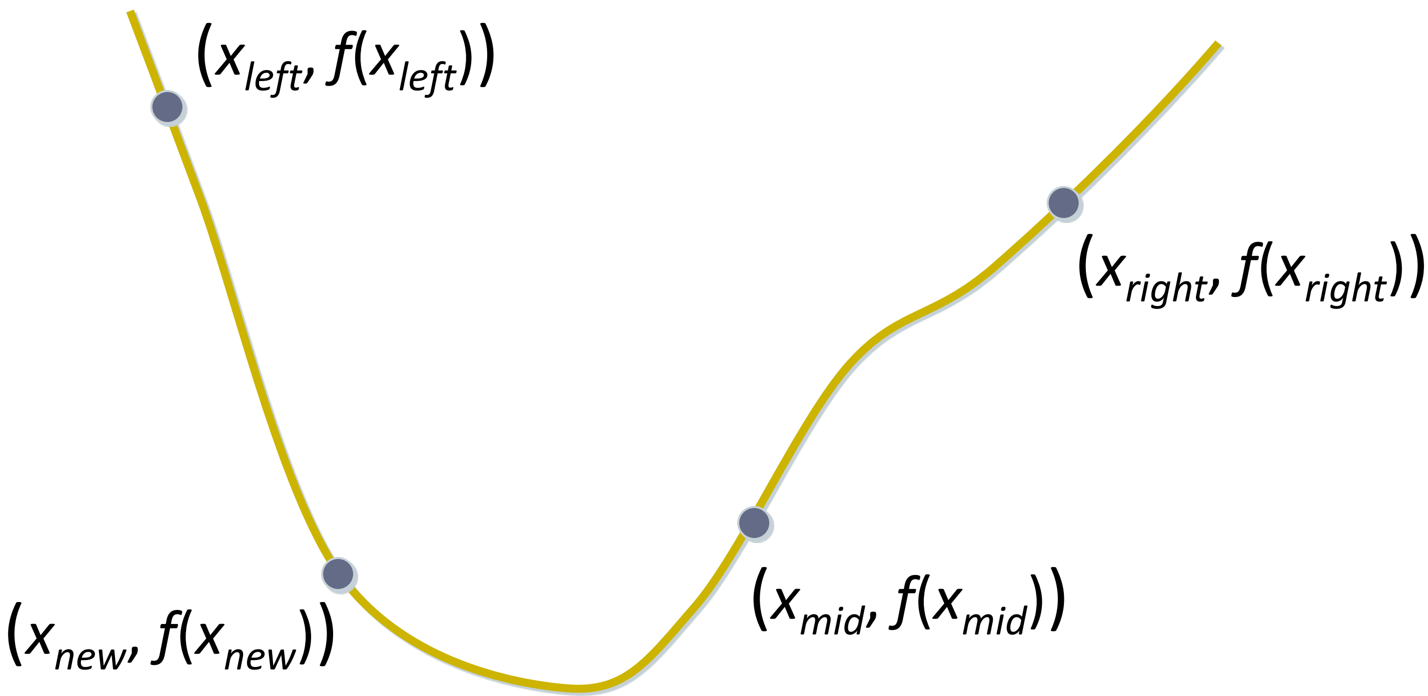
Strategija: oceni funkcijo v novem x_{new}

Postavi nove meje “bracket” točke so x_{new}, x_{mid}, x_{right}

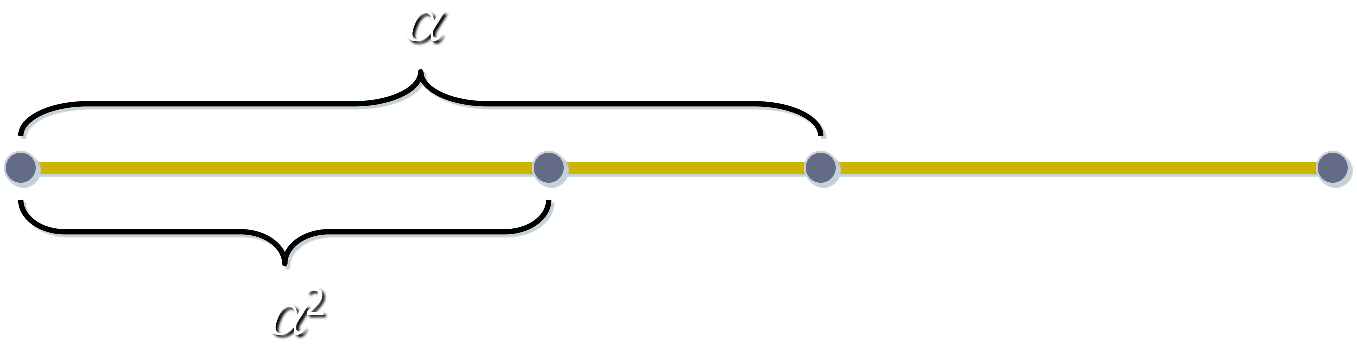


Strategija: oceni funkcijo v novem x_{new}

Nove točke omejitve so x_{left}, x_{new}, x_{mid}



Poiščimo optimalno točko x_{mid}, glede na levo in desno, tako da zagotavlja enako stopnjo redukcije.



if $f(x_{new}) < f(x_{mid})$

novi interval = α

else

novi interval = $1-\alpha^2$

Da zagotovimo enak interval, želimo $\alpha = 1-\alpha^2$

Velja,

$$\alpha = \frac{\sqrt{5}-1}{2} = \overline{\varphi}$$

To je fatktor “zlatega reza” = 0.618…

Interval se tako zmanjša za 30% na iteracijo

Linearna konvergenca

V bližini minimuma je odvod = 0, zato velja

$$f(x+\Delta x) = f(x) + \tfrac{1}{2} f''(x)\Delta x^2 + \ldots$$

$$f(x+\Delta x) - f(x) = \tfrac{1}{2} f''(x)\Delta x^2 = \text{machine } \varepsilon$$

$$\Rightarrow \Delta x \sim \sqrt{\varepsilon}$$

Pravilo palca: nesmiselno je iskati večjo natančnost, kot sqrt(ε)

Uporaba double precision, če želimo single-precision rezultat (in/ali imamo single-precision podatke)

Hitrejša 1-D optimizacija

Kompromis med super in linearno konvergenco za manjšo robustnost

Kombinacija z Zlatim rezom za varnost

Uporabni napotki:

Skozi 3 točke položimo parabolo in poiščemo minimum

Izračunamo odvode in položaje, uporabimo kubično prilagajanje

Uporabimo druge odvode: Newton

Iskalne metode

Začnemo s takšnim intervalom [x_L, x_U], v katerem leži iskani minimum x*.

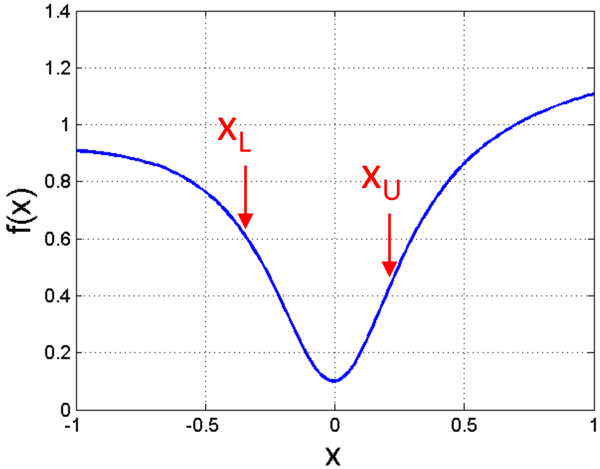
Določimo vrednost $f(x)$ v dveh točkah na intervalu.

Zmanjšamo interval.

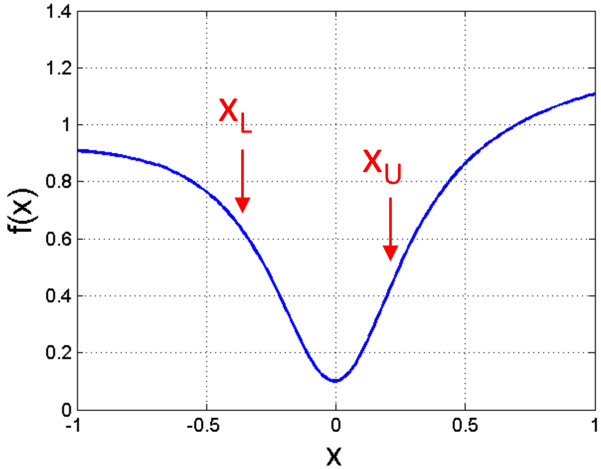
Ponovimo proces.

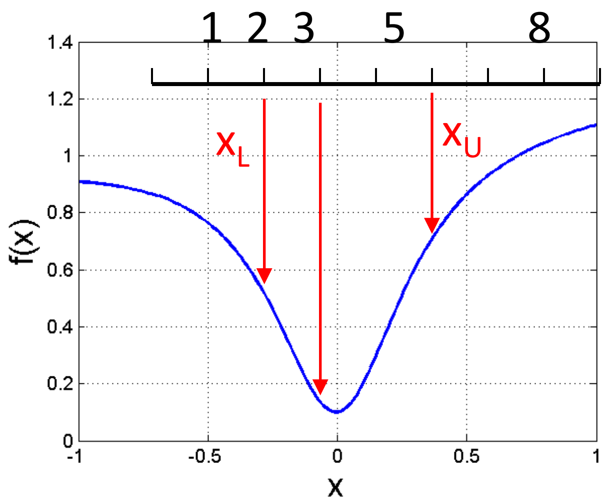
- Lahko uporabimo pri vsaki funkciji, odvedljivost ni ovira.

Iskalne metode



Dichotomous



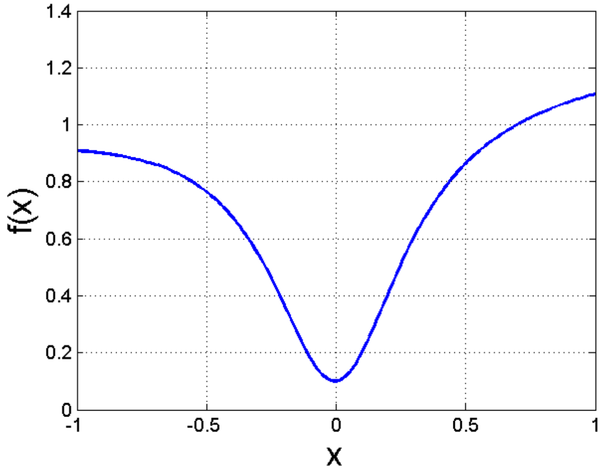


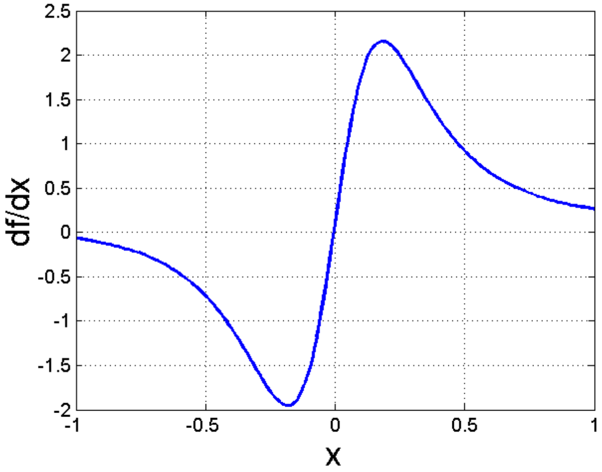
Fibonacci: 1 1 2 3 5 8 …

1D funkcije

Kot primer vzemimo funkcijo

$$f(x) = 0.1 + 0.1x + x^2/(0.1 + x^2)$$





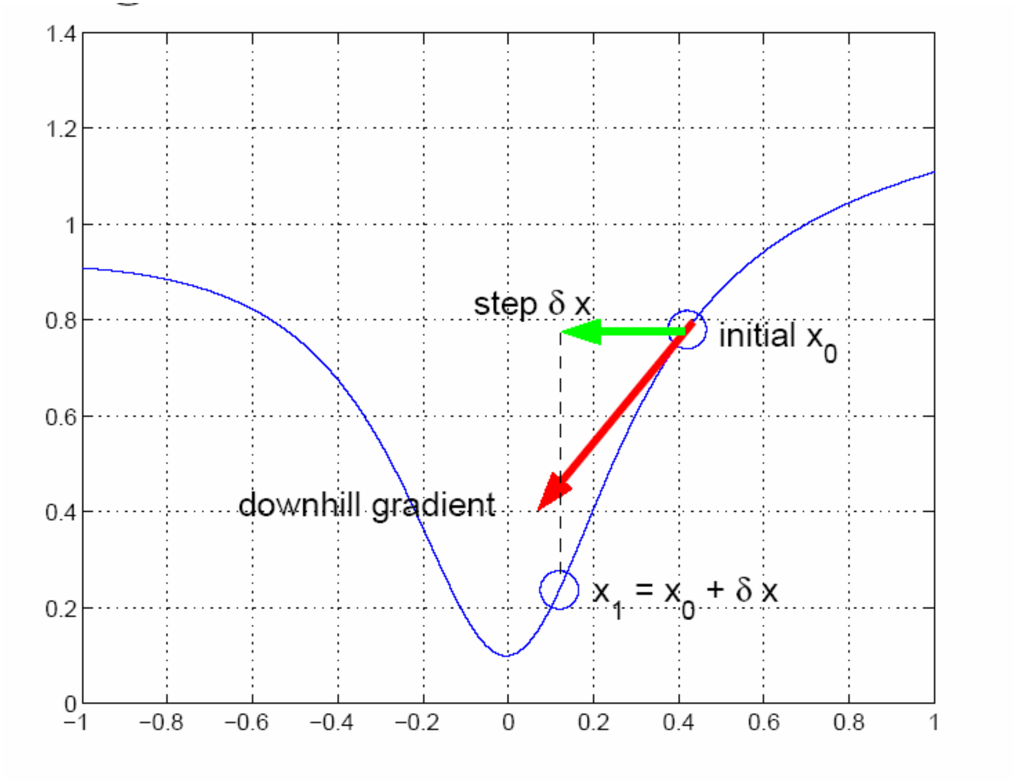
(od sedaj naprej predpostavljamo, da ne poznamo dejanskega funkcijskega izraza)

Zmanjševanje gradienta

Glede na začetno lokacijo, x_0, določimo df/dx

In se pomikamo navzdol po krivulji odvoda

da ustvarimo nov približek, $x_1 = x_0 + \delta x$



Kako določimo velikost koraka δx?

Metode zmanjševanja

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x^{(k)} \quad \text{with } f(x^{(k+1)}) < f(x^{(k)})$$

- other notations: $x^+ = x + t\Delta x$, $x := x + t\Delta x$
- Δx is the *step*, or *search direction*; t is the *step size*, or *step length*
- from convexity, $f(x^+) < f(x)$ implies $\nabla f(x)^T \Delta x < 0$ ($i.e.$, Δx is a *descent direction*)

General descent method.

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. Determine a descent direction Δx.
2. *Line search.* Choose a step size $t > 0$.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

Linijsko iskanje

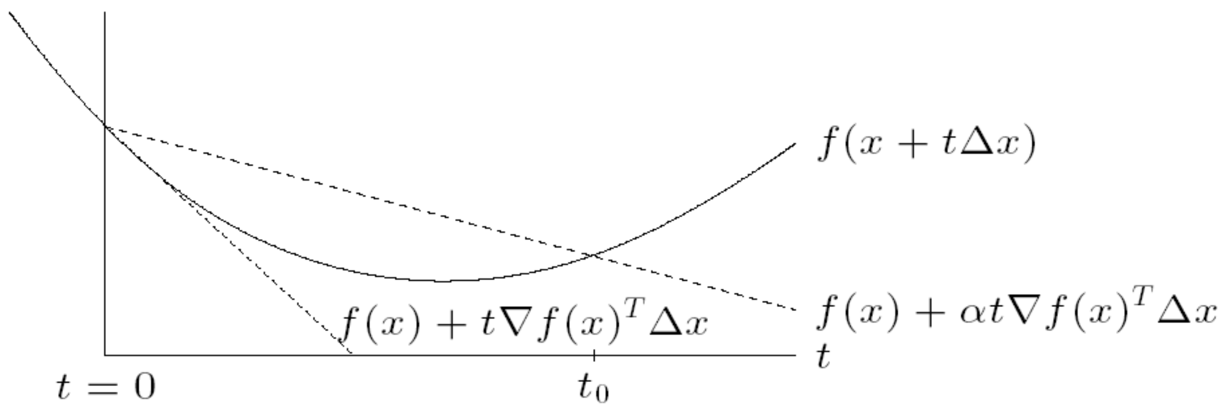
Natančno linijsko iskanje: $t = \operatorname{argmin}_{t>0} f(x + t\Delta x)$

Vzvratno linijsko iskanje (parametri $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$)

- začetek $t = 1$, ponavljaj $t := \beta t$ dokler

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$$

- Grafična interpretacija : vzvratno dokler $t \leq t_0$



Metoda zmanjševanja gradienta

Splošna gradientna metoda $\quad \Delta x = -\nabla f(x)$

---

glede na začetno točko $x \in \mathbf{dom}\, f$.
ponavljaj

1. $\Delta x := -\nabla f(x)$.
2. Linijsko iskanje. Izberi korak *t* glede na natančno ali vzvratno linijsko iskanje
3. Posodobi $x := x + t\Delta x$.

Dokler ni zadoščen ustavitveni kriterij

- Ustavitveni kriterij je običjano podan v obliki $\|\nabla f(x)\|_2 \leq \epsilon$
- Rezultat konvergence: za močno koveksne *f*

$$f(x^{(k)}) - p^\star \leq c^k (f(x^{(0)}) - p^\star)$$

$c \in (0,1)$ odvisno od m, $x^{(0)}$, tipa linijskega iskanja

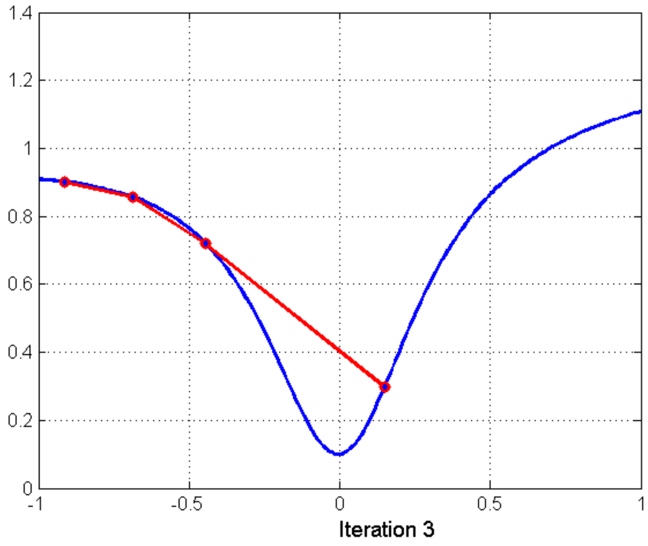
- Zelo preprosto, vendar pogosto počasno, redko uporabljano

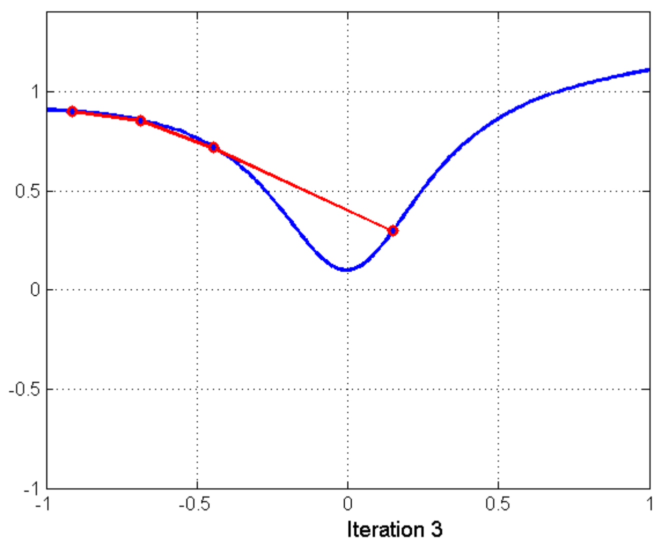
Polinomska interpolacija

- Zajem minimuma v intervalu.
- Uporaba kvadratne ali kubične funkcije za interpolacijo nekaterih točk $f(x)$ na intervalu.
- Pomik na lahko določljiv minimum izbranega polinoma.
- Završemo najslabšo točko in ponovimo proces.

Newton-ova metoda

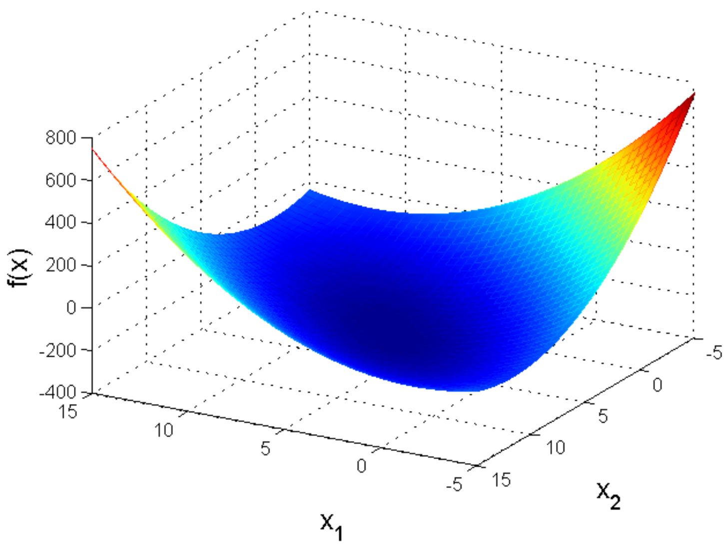
- Globalno gledano je konvergenca Newton-ove metode slaba.
- Pogosto odpove, če je začetna točka preveč oddaljena od minimuma.

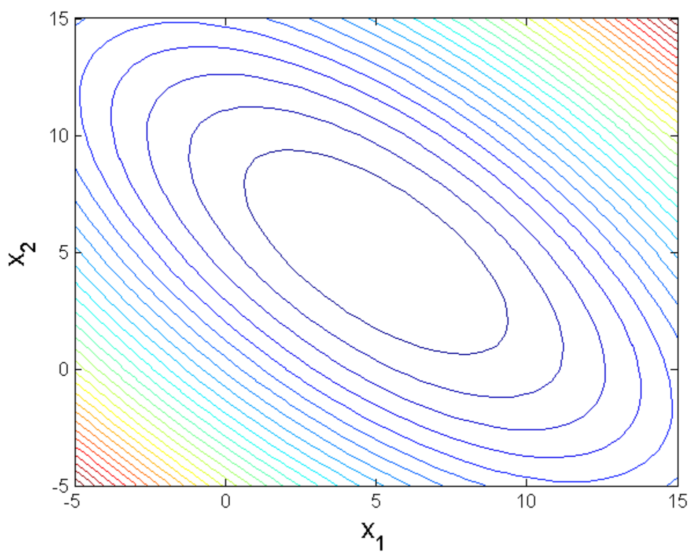




Razširitev na N (multivariantne) dimenzije

- Kako velik je lahko N?
  - Velikosti variirajo od zgolj nekaj do več tisoč parametrov
- Obravnavali bomo primer za N=2, da bo lahko vizualizirana stroškovna funkcija.





Optimizacijski algoritem

- Začetna točka $\mathbf{x}_0$, $k = 0$.

1. Izračunamo smer iskanja $\mathbf{p}_k$
2. Izračunamo dolžino koraka α_k, da bo $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) < f(\mathbf{x}_k)$
3. Določimo $\mathbf{x}_k = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
4. Preverimo približevanje (kriteriji) e.g. $df/d\mathbf{x} = \mathbf{0}$

$k = k+1$

Taylor-jeva razširitev

Vsako funkcijo lahko razvijemo v Taylorjevo vrsto okrog točke $\mathbf{x}^*$

$$\boxed{f(\mathbf{x}^* + \mathbf{x}) \approx f(\mathbf{x}^*) + \nabla f^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{H} \mathbf{x}}$$

Kjer je gradient $\nabla f(\mathbf{x}^*)$vektor

$$\nabla f(\mathbf{x}^*) = \left[\frac{\partial f}{x_1} \dots \frac{\partial f}{x_N}\right]^T$$

In $\mathbf{H}(\mathbf{x}^*)$ Hessova simetrična matrika

$$\mathbf{H}(\mathbf{x}^*) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_N} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_N \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_N^2} \end{bmatrix}$$

Kvadratne funkcije

$$f(\mathbf{x}) = a + \mathbf{g}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}$$

- Vektor $\mathbf{g}$ in Hessova matrika $\mathbf{H}$ sta constantna.
- Aproksimacija drugega reda katerekoli funkcije s Taylorjevo razširitvijo je kvadratna funkcija.

Predpostavili bomo samo kvadratne funkcije.

Potrebni pogoji za minimum

$$f(\mathbf{x}) = a + \mathbf{g}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}$$

Razširimo $f(\mathbf{x})$ okrog stacionarne točke $\mathbf{x}^*$ v smeri $\mathbf{p}$

$$\begin{aligned} f(\mathbf{x}^* + \alpha\mathbf{p}) &= f(\mathbf{x}^*) + \mathbf{g}(\mathbf{x}^*)^T\alpha\mathbf{p} + \frac{1}{2}\alpha^2\mathbf{p}^T\mathbf{H}\mathbf{p} \\ &= f(\mathbf{x}^*) + \frac{1}{2}\alpha^2\mathbf{p}^T\mathbf{H}\mathbf{p} \end{aligned}$$

Ker v stacionarni točki velja $\mathbf{g}(\mathbf{x}^*) = 0$

V stacionarni točki je obnašanje funkcije določeno s $\mathbf{H}$

- $\mathbf{H}$ je simetrična matrika, zato ima ortogonalne lastne vektorje

$$\mathbf{H}\mathbf{u}_i = \lambda_i \mathbf{u}_i \qquad \|\mathbf{u}_i\| = 1$$

$$\begin{aligned} f(\mathbf{x}^* + \alpha \mathbf{u}_i) &= f(\mathbf{x}^*) + \frac{1}{2}\alpha^2 \mathbf{u}_i^T \mathbf{H} \mathbf{u}_i \\ &= f(\mathbf{x}^*) + \frac{1}{2}\alpha^2 \lambda_i \end{aligned}$$

- Ko se $|\alpha|$ povečuje, se $f(\mathbf{x}^* + \alpha \mathbf{u}_\mathrm{i})$ povečuje, zmanjšuje ali je nespremenljiva glede na to, ali je λ_i pozitiven, negativen ali nič

Primeri kvadratnih funkcij

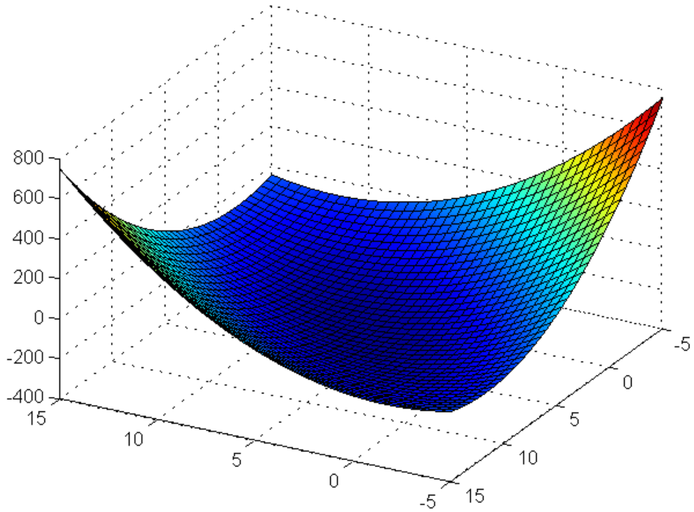
Primer1: obe lastni vrednosti sta pozitivni

$$f(\mathbf{x}) = a + \mathbf{g}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}$$

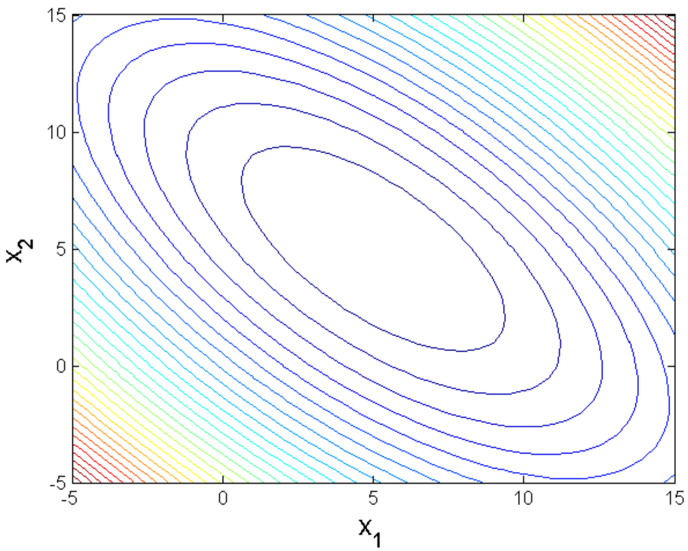
z

$$a = 0\,, \qquad \mathbf{g} = \begin{bmatrix} -50 \\ -50 \end{bmatrix}\,, \qquad \mathbf{H} = \begin{bmatrix} 6 & 4 \\ 4 & 6 \end{bmatrix}$$

Pozitivno določeno



minimum



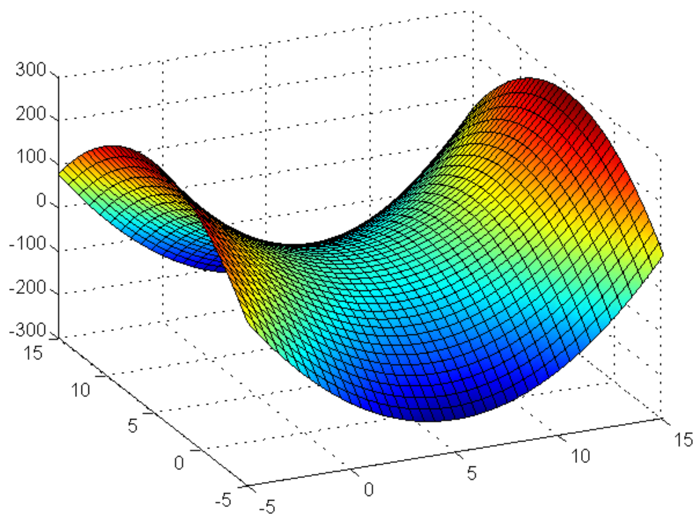
Primeri kvadratnih funkcij

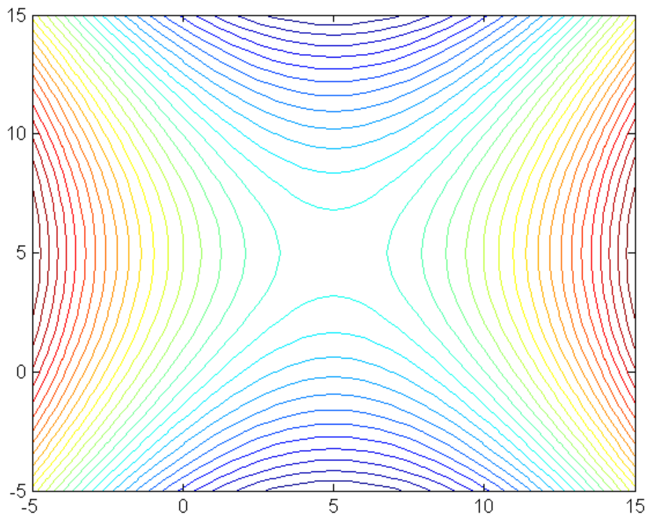
Primer1: lastni vrednosti imata različen predznak

$$f(\mathbf{x}) = a + \mathbf{g}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}$$

z

$$a = 0\,, \qquad \mathbf{g} = \begin{bmatrix} -30 \\ 20 \end{bmatrix}, \qquad \mathbf{H} = \begin{bmatrix} 6 & 0 \\ 0 & -4 \end{bmatrix}$$ nedoločeno





sedlo

Primeri kvadratnih funkcij

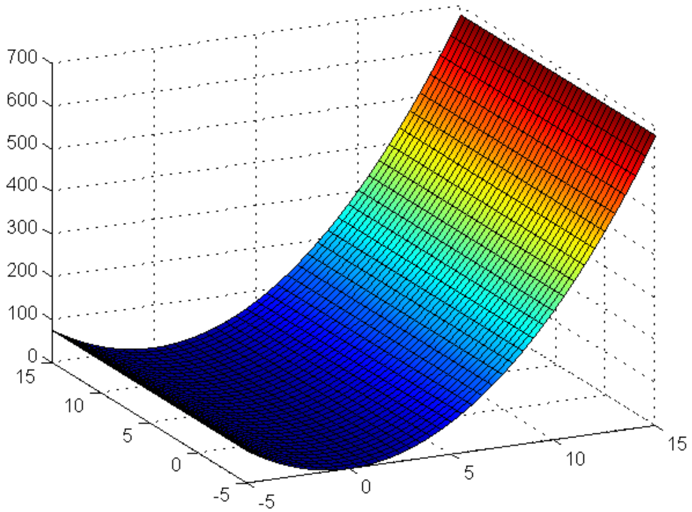
Primer1: obe lastni vrednosti sta enaki 0

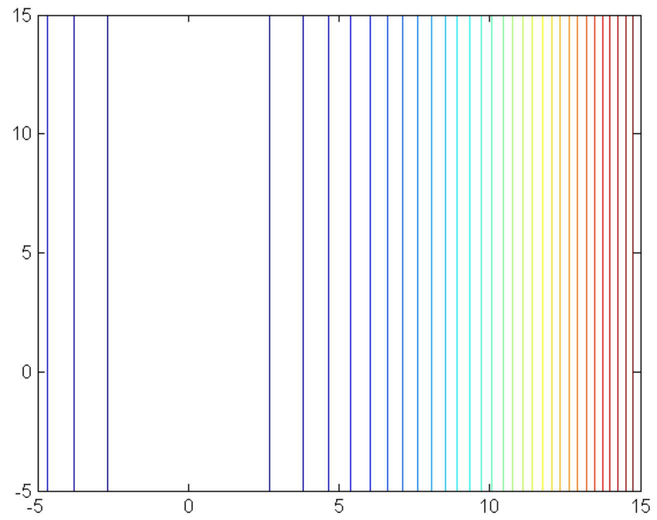
$$f(\mathbf{x}) = a + \mathbf{g}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{H}\mathbf{x}$$

z

$$a = 0\,, \qquad \mathbf{g} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \mathbf{H} = \begin{bmatrix} 6 & 0 \\ 0 & 0 \end{bmatrix}$$

Pozitivno poldoločeno





parabolični cilinder

Optimizacija kvadratnih funkcij

Predpostavimo, da je $\mathbf{H}$ pozitiven in določen

$$f(\mathbf{x}) = a + \mathbf{g}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}$$

$$\nabla f(\mathbf{x}) = \mathbf{g} + \mathbf{H}\mathbf{x}$$

Obstaja edinstven minimum

$$\mathbf{x}^* = -\mathbf{H}^{-1}\mathbf{g}$$

Če je N velik, potem ta inverzija ni direktno mogoča.

Najstrmejše zmanjševanje

- Osnovni princip je minimizirati N-dimenzionalno funkcijo s serijo 1D premočrtnih-minimizacij:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

- Metoda najstrmejšega zmanjševanja izbere $\mathbf{p}_k$ tako, da je vzporeden z gradientom

$$\mathbf{p}_k = -\nabla f(\mathbf{x}_k)$$

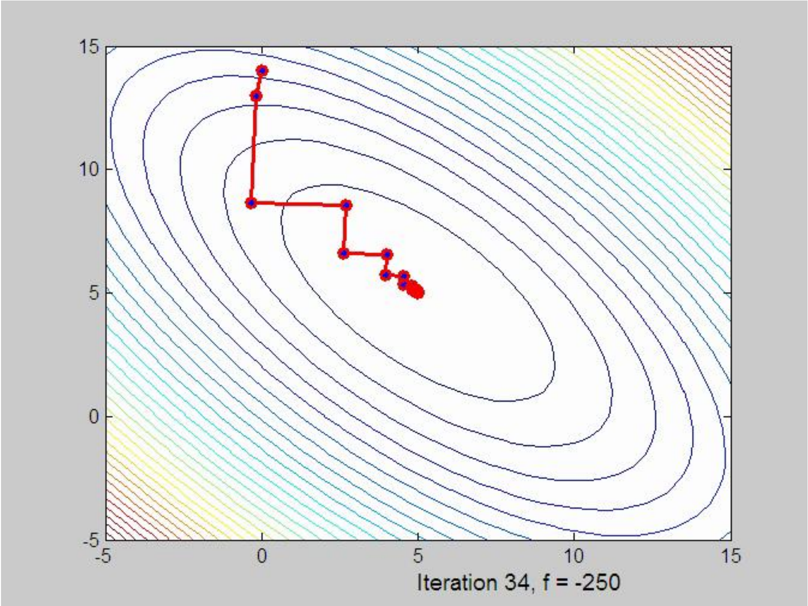
- Korak α_k je izbran tako da minimizira $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$.

  Za kvadratične oblike obstaja rešitev:

$$\alpha_k = \frac{\mathbf{p}_k^T \mathbf{p}_k}{\mathbf{p}_k^T \mathbf{H} \mathbf{p}_k}$$

Dokaži!

Najstrmejše zmanjševanje



- Gradient je povsod pravokoten na linije.
- Po vsaki premočrtni minimizaciji je novi gradient vedno ortogonalen na smer predhodnega koraka.
- Posledično, imajo iteracije cik-cak smer.

Konjugirani gradient

- Vsak $\mathbf{p}_k$ je določen kot konjugirana vrednost prejšnjih smeri glede na $\mathbf{H}$:

$$\mathbf{p}_i^T \mathbf{H} \mathbf{p}_j = 0 \,, \qquad i \neq j$$

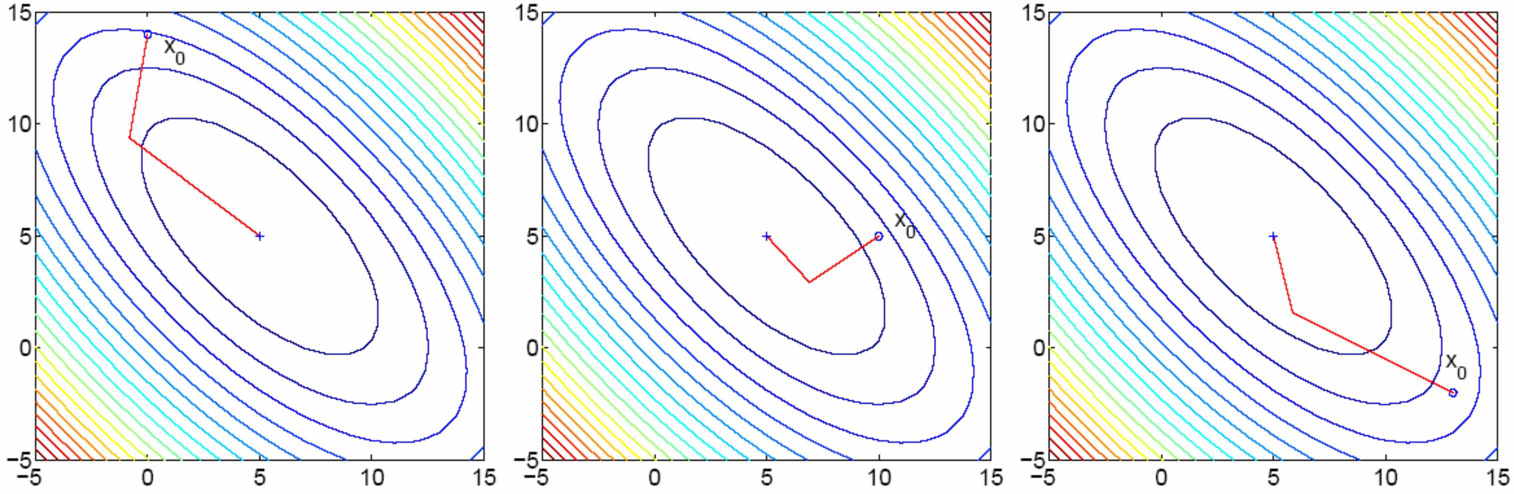
- Sledi: linearna neodvisnost smeri iskanja.

- $\mathbf{p}_k$ lahko izberemo zgolj glede na poznavanje $\mathbf{p}_{k-1}$, $\nabla f(\mathbf{x}_{k-1})$, and $\nabla f(\mathbf{x}_k)$

$$\mathbf{p}_k = \nabla f_k + \left(\frac{\nabla f_k^\top \nabla f_k}{\nabla f_{k-1}^\top \nabla f_{k-1}} \right) \mathbf{p}_{k-1}$$

Konjugirani gradient

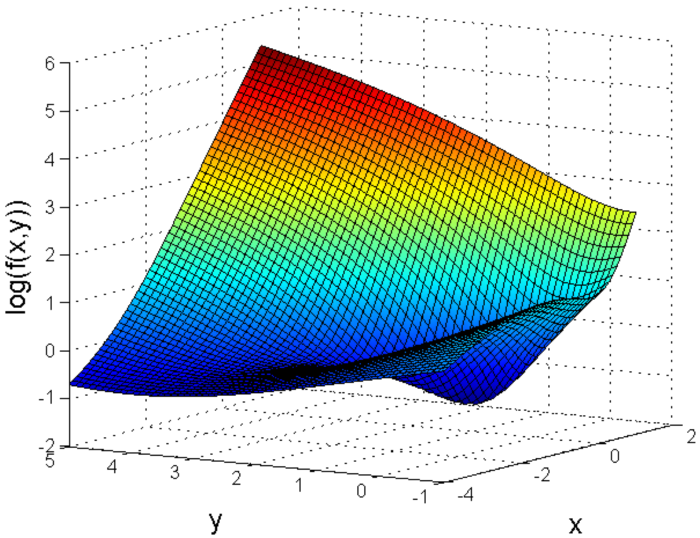
- N-dimenzionalno kvadratno obliko je mogoče minimizirati v največ N korakih konjugiranega gradienta.

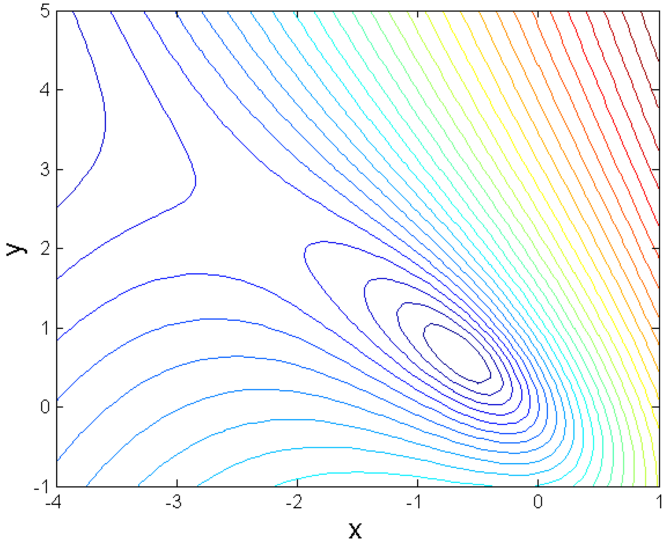


- Primer s tremi različnimi začetnimi točkami.
- Minimum je dosežen v natanko dveh korakih.

Optimizacija za splošne funkcije

$$f(x, y) = \exp(x)(4x^2 + 2y^2 + 4xy + 2x + 1)$$

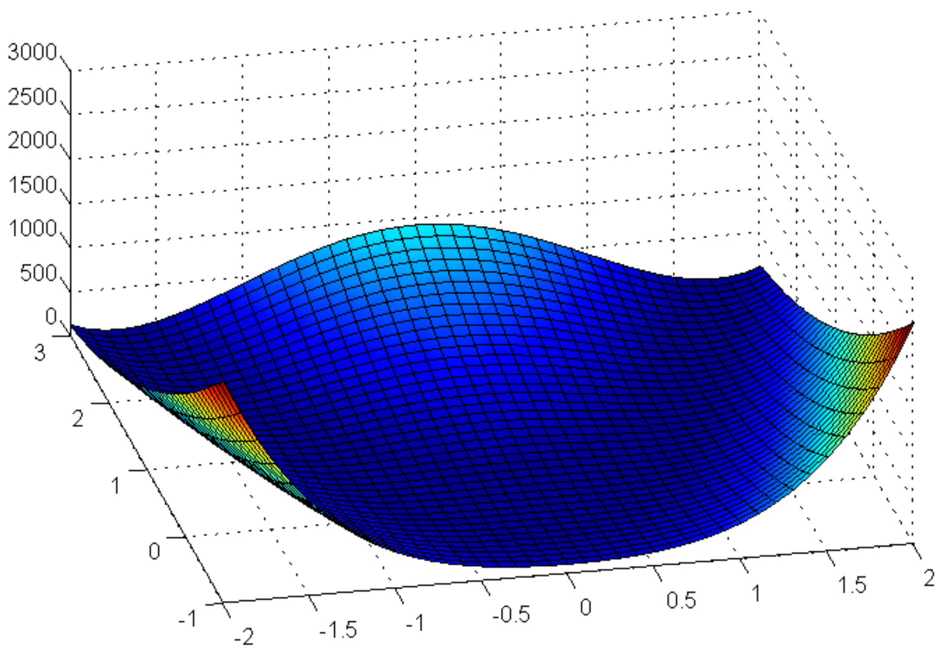


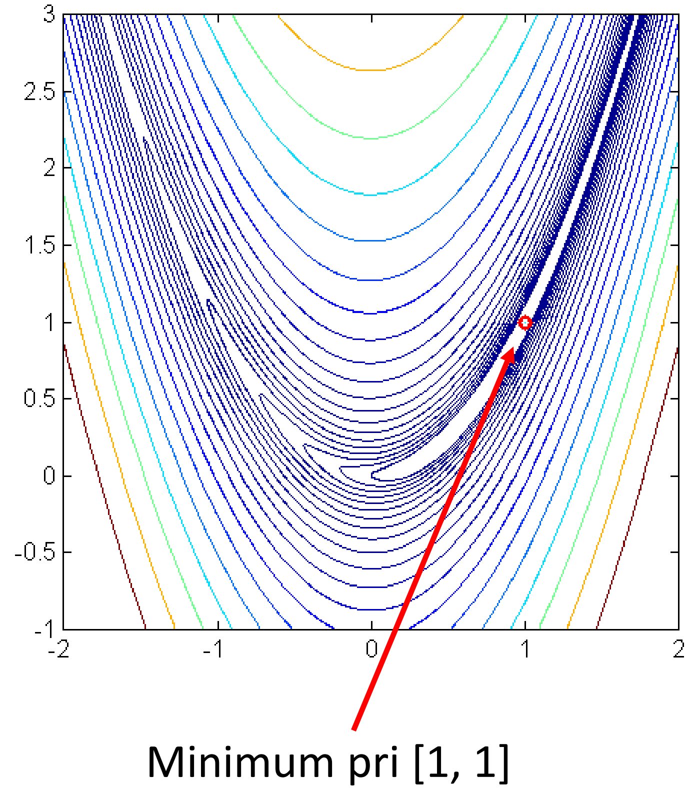


Metode, ki uporabljajo kvadratično Taylorjevo razširitev v vrsto

Rosenbrock-ova funkcija

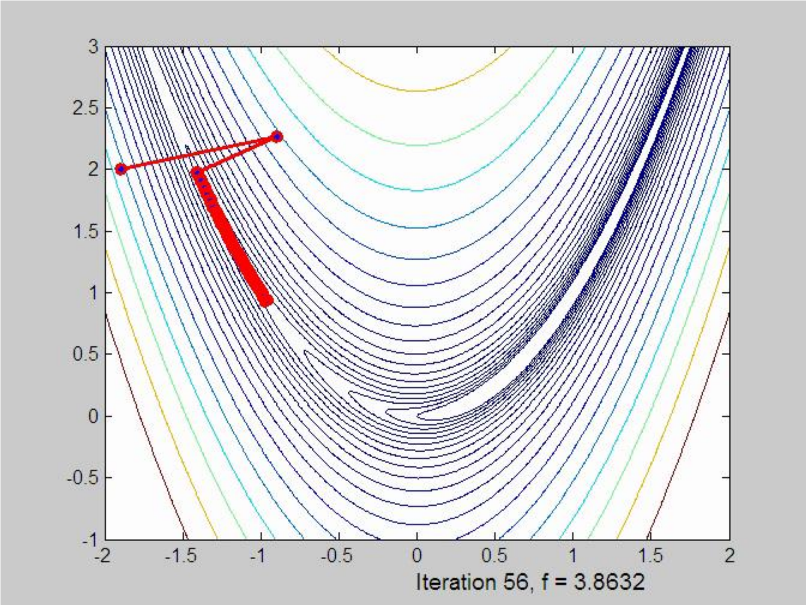
$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2$$

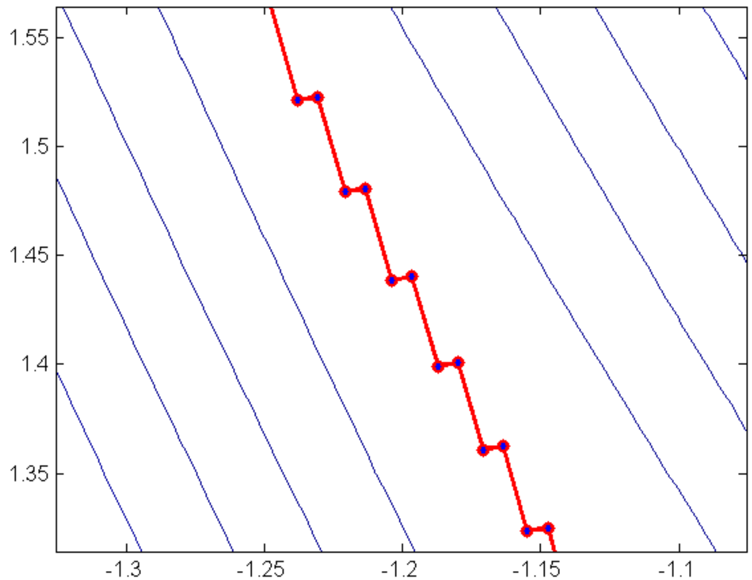




Najstrmejši spusta

- 1-D premočrtna minimizacije je izvedena z uporabo ene od prej navedenih metod (običajno kubična polinomska interpolacija)

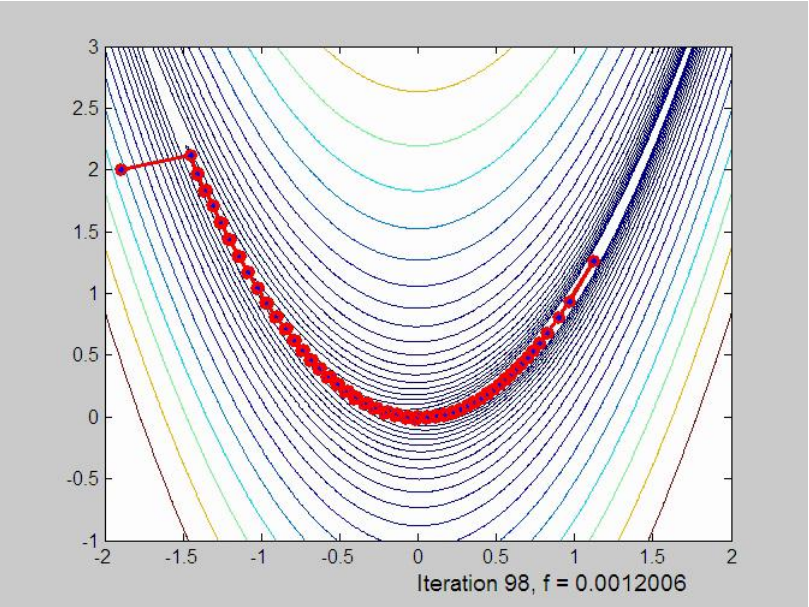




- Razvidno je cik-cak obnašanje (levo)
- Algoritem počasi napreduje proti minimumu (‘‘down the valey‘‘)

Konjugirani gradient

- Ponovno je potrebna uporaba eksplicitne premočrtne minimizacije v vsakem koraku



- Algoritem je končan po 98 iteracijah
- Boljše kot metoda najstrmejšega zmanjševanja

Newtonova metoda

Razširimo $f(\mathbf{x})$ v Taylorjevo vrsto okrog točke $\mathbf{x}_k$

$$f(\mathbf{x}_k + \delta\mathbf{x}) \approx f(\mathbf{x}_k) + \mathbf{g}_k^T \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^T \mathbf{H}_k \delta\mathbf{x}$$

Pri tem je gradientni vektor

$$\mathbf{g}_k = \nabla f(\mathbf{x}_k) = \left[\frac{\partial f}{x_1} \ldots \frac{\partial f}{x_N}\right]^T$$

In Hessova simetrična matrika

$$\mathbf{H}_k = \mathbf{H}(\mathbf{x}_k) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_N \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_N^2} \end{bmatrix}$$

Newtonova metoda

V minimumu velja $\nabla f(\mathbf{x}) = \mathbf{0}$, torej

$$\nabla f(\mathbf{x}) = \mathbf{g}_k + \mathbf{H}_k \delta\mathbf{x} = \mathbf{0}$$

Z rešitvijo $\delta\mathbf{x} = -\mathbf{H}_k^{-1}\mathbf{g}_k$. Rezultat je iterativni izračun

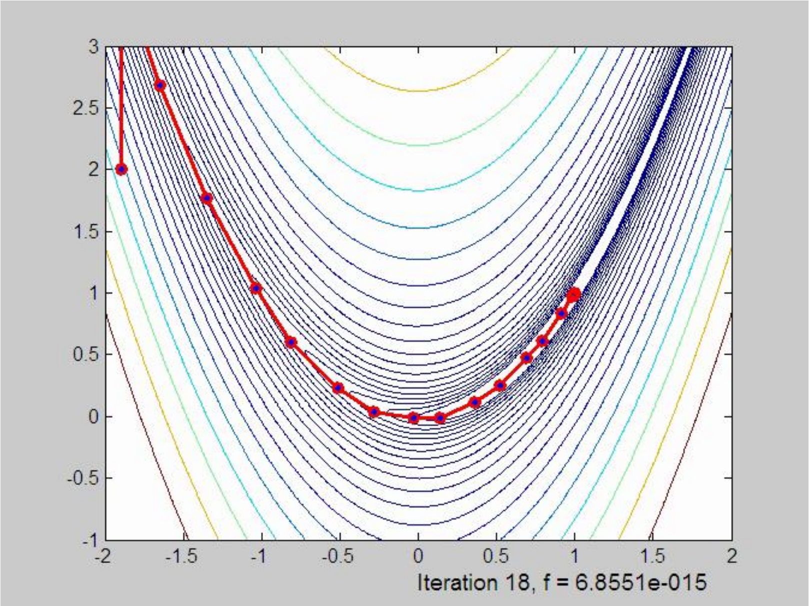
$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1}\mathbf{g}_k$$

- Če je $f(\mathbf{x})$ kvadratna, potem najdemo rešitev v enem koraku.
- Metoda ima kvadratno konvergenco (kot v 1D primeru).
- Rešitev $\delta\mathbf{x} = -\mathbf{H}_k^{-1}\mathbf{g}_k$ je gotovo v smeri zmanjševanja.
- Boljše, kot skočiti direktno v minimum, je uporabiti premočrtno minimizacijo, ki zagotavlja globalno konvergenco

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}_k^{-1}\mathbf{g}_k$$

- Če je $\mathbf{H}=\mathbf{I}$ potem se to poenostavi v najstrmejše zmanjševanje.

Newtonova metoda – primer



- Algoritem najde minimum po samo 18 iteracijah v primerjavi z 98-imi konjugiranega gradienta.
- Kljub temu pa metoda zahteva izračun Hessove matrike pri vsaki iteraciji, kar pa ni vedno izvedljivo.

Kvazi-Newtonove metode

- Če je je velikost problema velika in je Hessova matrika gosta, potem je lahko direkten izračun neizvedljiv.
- Kvazi-Newtonove metode zaobidejo ta problem z uporabo približka matrike H(x), ki ga posodablja ob vsaki iteraciji.
- Skupne sheme so Broyden, Goldfarb, Fletcher and Shanno (BFGS) in Davidson, Fletcher and Powell (DFP).
- Osnovni princip izkorišča, da lahko pri kvadratnih funkcijah določimo $\mathbf{H}$ na podlagi $\mathbf{g}_k$ in $\mathbf{x}_k$.

$$\mathbf{g}_{k+1} - \mathbf{g}_k = \mathbf{H}(\mathbf{x}_{k+1} - \mathbf{x}_k)$$

Metode konjugiranega gradienta

- Konjugirani gradient implicitno pridobi informacijo o Hessovi matriki
- Za kvadratno funkcijo v *n* dimenzijah pridobi natančno rešitev v *n* korakih (ob neupoštevanju napake zaokroževanja)
- Deluje dobro v praksi…

Metode vrednotenja v več dimenzijah

- Če ni mogoče ovrednotiti gradienta, imamo težave
- Lahko uporabimo približne (numerično ocenjene) gradiente:

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial e_1} \\ \frac{\partial f}{\partial e_2} \\ \frac{\partial f}{\partial e_3} \\ \vdots \end{pmatrix} \approx \begin{pmatrix} \frac{f(x+\delta \cdot e_1) - f(x)}{\delta} \\ \frac{f(x+\delta \cdot e_2) - f(x)}{\delta} \\ \frac{f(x+\delta \cdot e_3) - f(x)}{\delta} \\ \vdots \end{pmatrix}$$

Strategije generične optimizacije

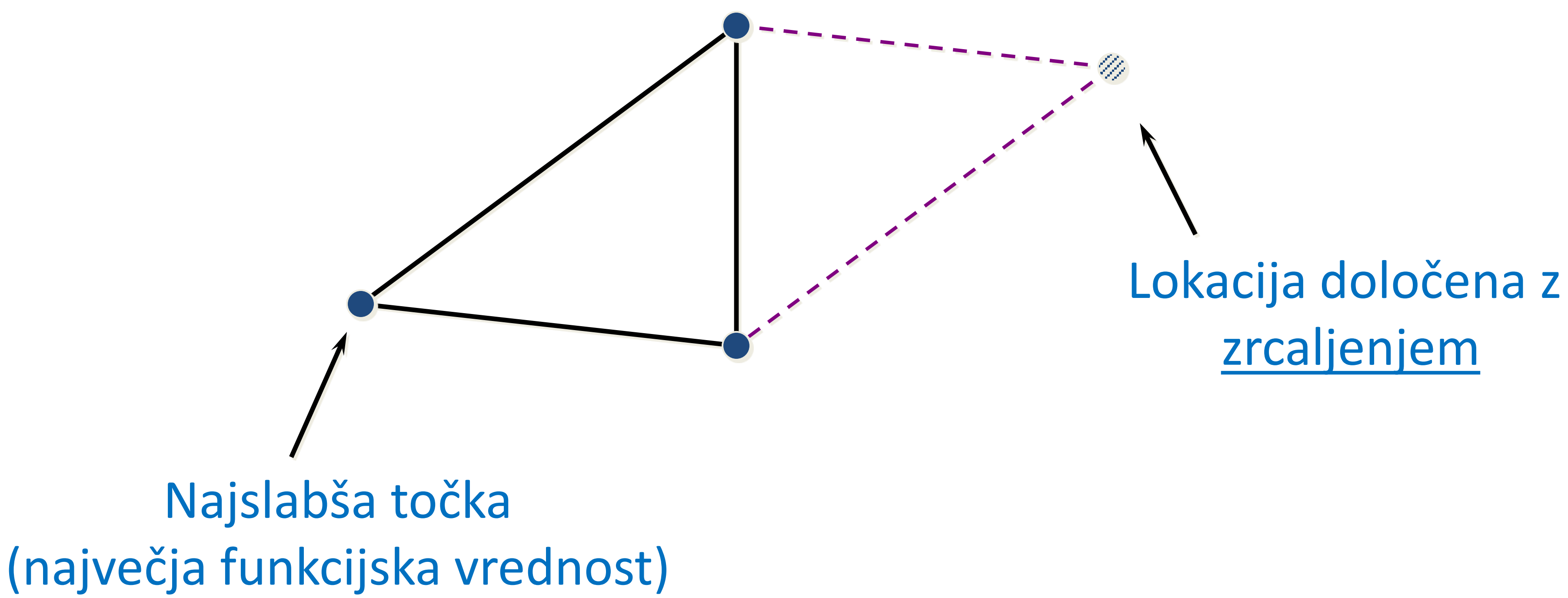
- Enotno vzorčenje:
  - Zahtevnost narašča eksponentno s številom dimenzij
- Simulirano žarjenje:
  - Iskanje v naključnih smereh
  - Na začetku z velikimi koraki, ki jih postopno zmanjšujemo
  - “Razpored žarjenja” – kako hitro upočasniti?

Simplex metoda (Nelder-Mead)

- Sledimo *n*+1 točkam v *n* dimenzijah
  - Vozlišča *simplex* (trikotnik v 2D tetraeder v 3D, itd.)
- Pri vsaki iteraciji: simplex se lahko premakne, razširi ali skrči
  - Poznana tudi kot *ameba metoda*: simplex "izžareva" vzdolž funkcije

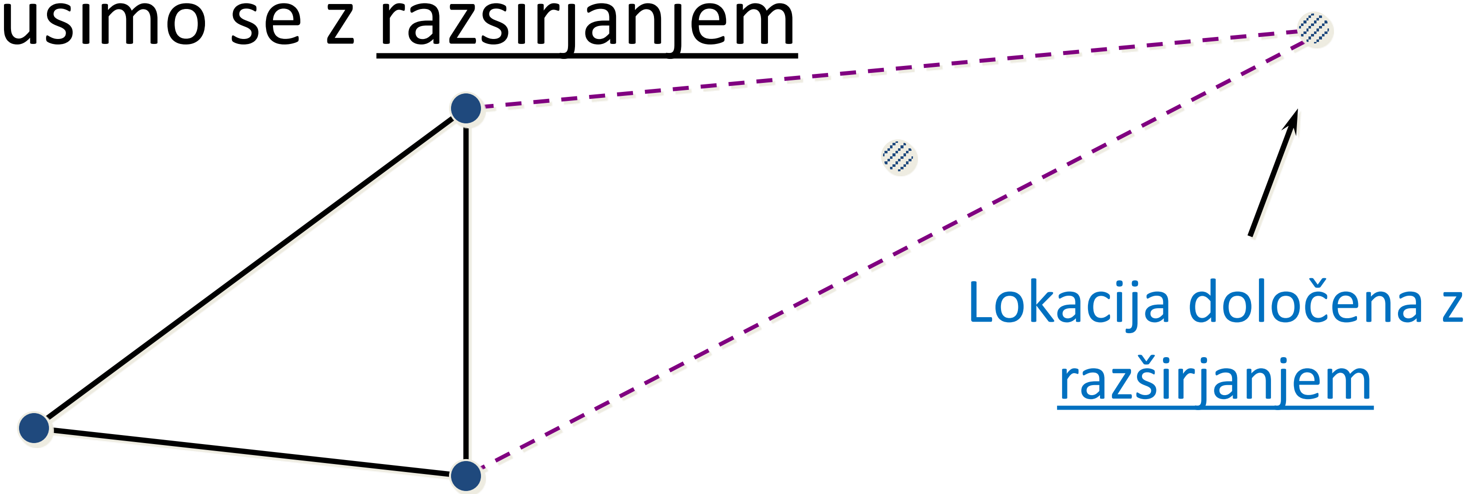
Simplex metoda (Nelder-Mead)

- Osnovna operacija: <u>zrcaljenje</u>



Simplex metoda (Nelder-Mead)

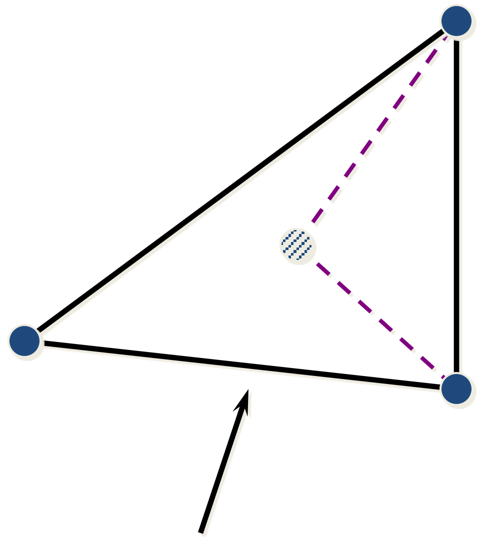
- Če je zrcaljenje določilo najboljšo (najmanjšo) vrednost do sedaj, potem poskusimo še z razširjanjem



- V nasprotnem primeru, če je zrcaljenje učinkovalo v kakršni koli meri, ga nadaljujemo

Simplex metoda (Nelder-Mead)

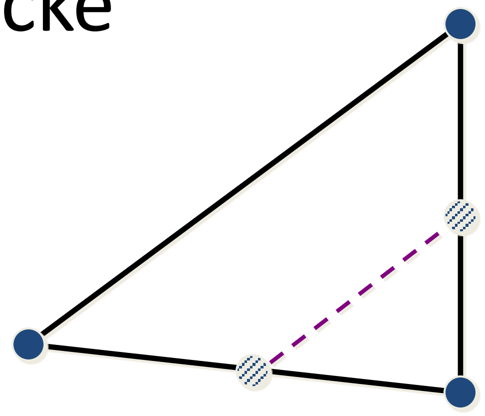
- Če zrcaljenje ni učinkovalo (zrcaljena točka je še vedno najslabša) poskusimo s <u>krčenjem</u>



Lokacija določena s <u>krčenjem</u>

Simplex metoda (Nelder-Mead)

- Če ni delovala nobena od prej omenjenih operacij, <u>zmanjšamo</u> simplex okrog *najboljše* točke



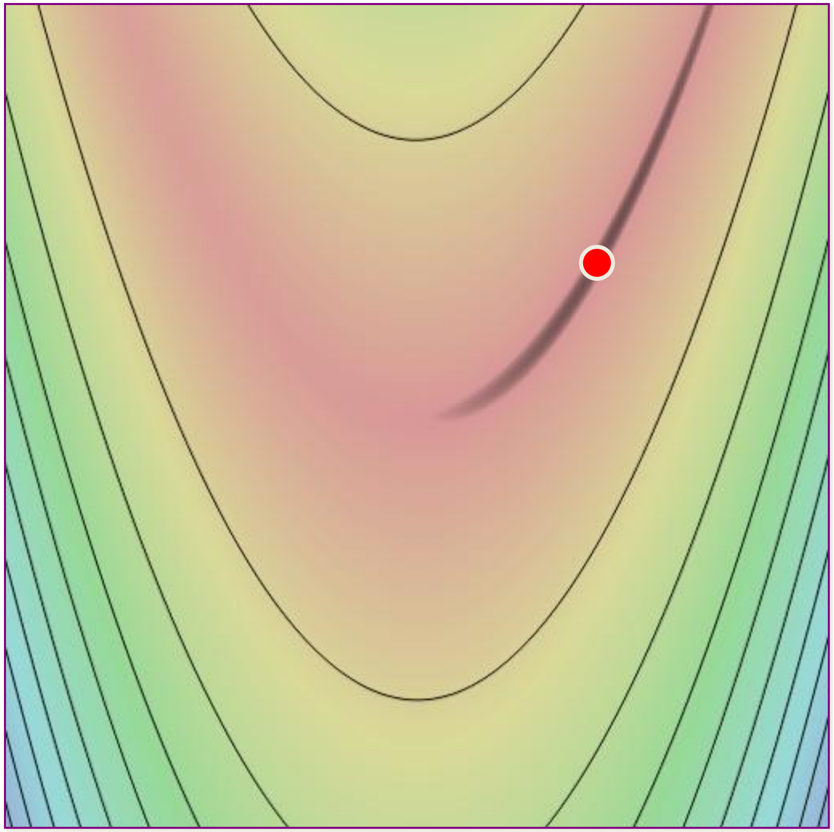
Simplex metoda (Nelder-Mead)

- Metoda je precej učinkovita ob vsaki iteraciji (tipično 1 – 2 oceni funkcije)
- Lahko zahteva *veliko* iteracij
- Nekoliko nezanesljiva – včasih je potreben *ponoven začetek* zaradi sesedanja simplex-a
- Prednosti: preprosta za implementacijo, ne potrebuje odvoda, ni odvisna od gladkosti funkcije itd.

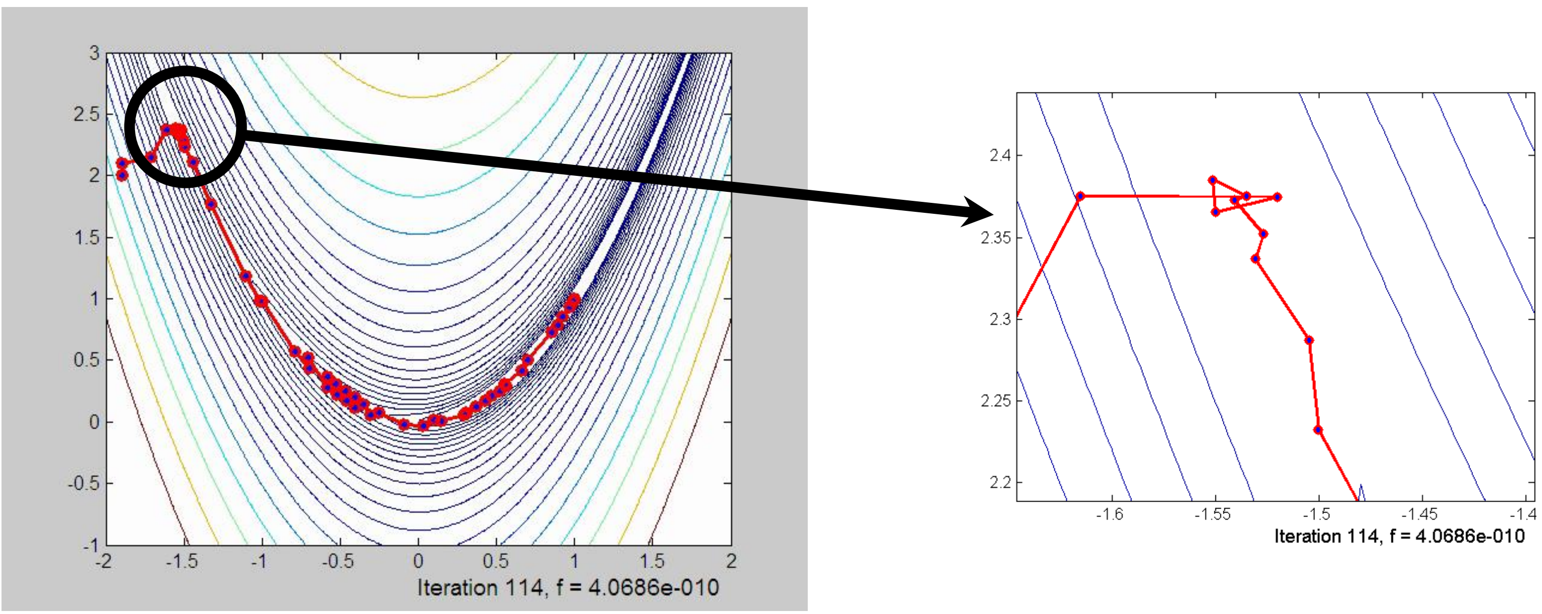
Rosenbrock-ova funkcija

- Oblikovana posebej za preizkušanje optimizacijskih tehnik
- Ukrivljena, ozka dolina

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2$$



Metoda Simplex



Optimizacije z omejitvijo

$$f(\mathbf{x}) : \mathbb{R}^N \longrightarrow \mathbb{R}$$

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} f(\mathbf{x})$$

Glede nas:

- Omejitvene enačbe: $a_i(\mathbf{x}) = 0 \quad i = 1, 2, \ldots, p$
- Omejitvene neenačbe : $c_j(\mathbf{x}) \geq 0 \quad j = 1, 2, \ldots, q$

- Omejitve določajo dopustna področja, ki so neprazna.

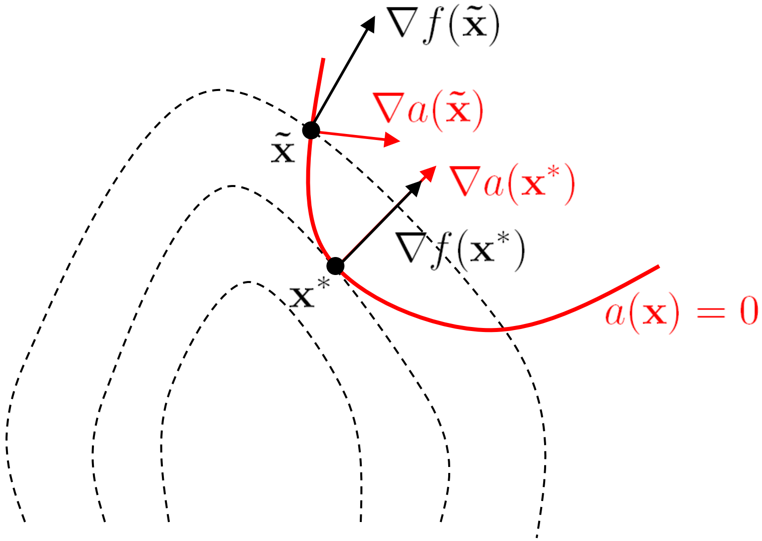
- Osnovna ideja je pretvorba problema z omejitvijo v problem brez omejitve.

Omejitvene enačbe

- Minimiziramo $f(\mathbf{x})$ glede na: $a_i(\mathbf{x}) = 0$ za $i = 1, 2, \ldots, p$

- Gradient funkcije $f(\mathbf{x})$ v lokalni točki minimizacije je enak linearni kombinaciji gradientov omejitev $a_i(\mathbf{x})$ z Lagrange-ovimi multiplikatorji kot koeficienti.

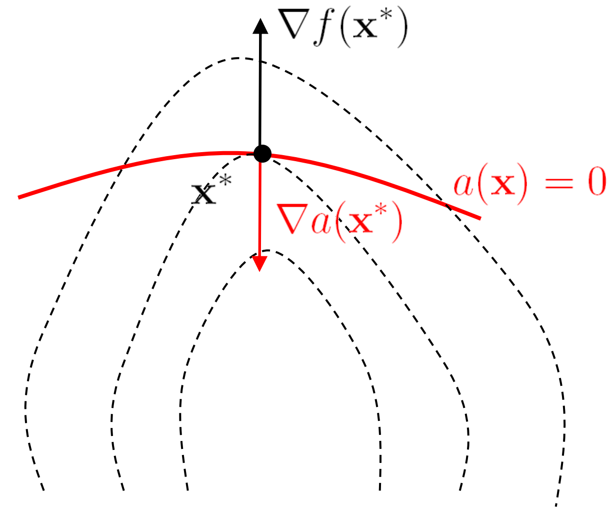
$$\nabla f(\mathbf{x}^*) = \sum_{i=1}^{p} \lambda_i^* \nabla a_i(\mathbf{x}^*)$$



$f_3 > f_2 > f_1$

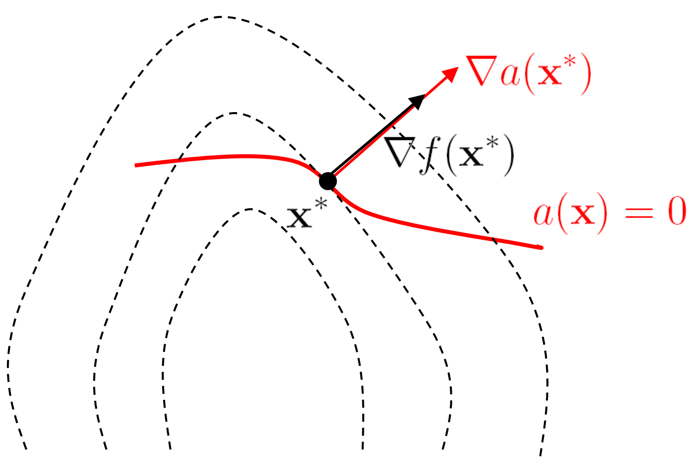
$\tilde{\mathrm{x}}$ ni točka minimuma

x* je točka minimuma,
$\lambda^*>0$



$f_3 > f_2 > f_1$

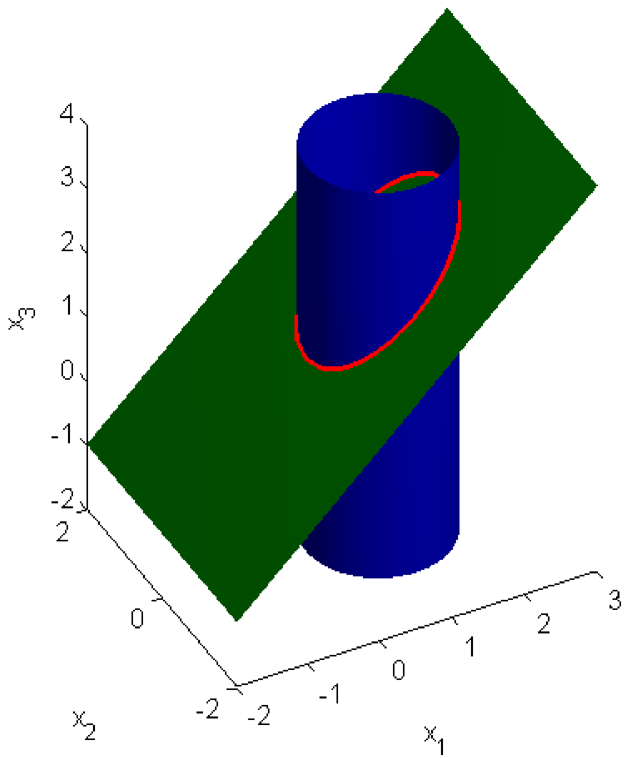
x* je točka minimuma,
$\lambda^*<0$



x* ni točka minimuma

$f_3 > f_2 > f_1$

3D Primer

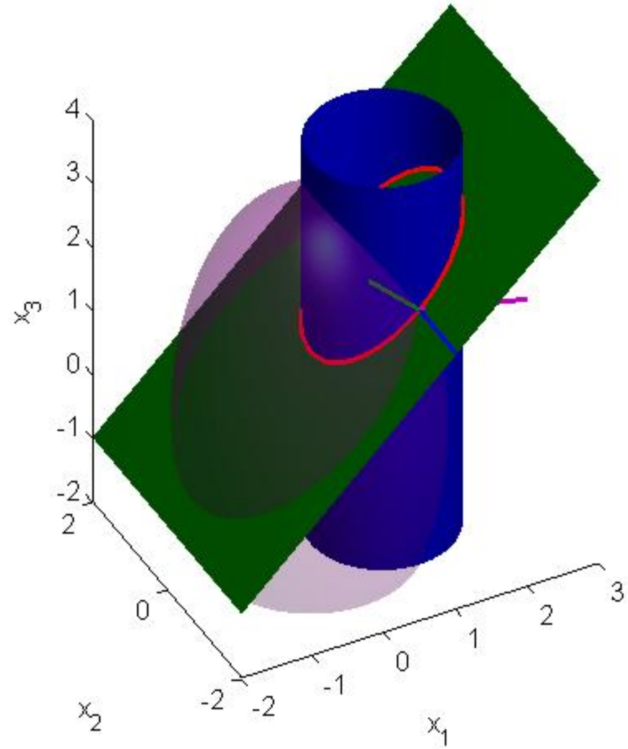


$$a_1(\mathbf{x}) = -x_1 + x_3 - 1 = 0$$

$$a_2(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1 = 0$$

3D Primer

$$f(\mathbf{x}) = x_1^2 + x_2^2 + \frac{1}{4}x_3^2$$



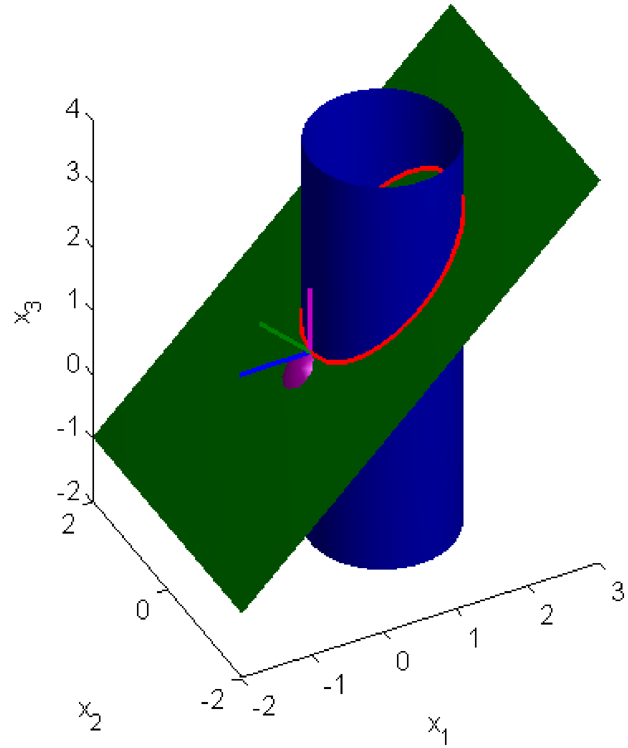
$$f(\mathbf{x}) = 3$$

Gradienti omejitev in kriterijske funkcije so linearno neodvisni.

3D Primer

$$f(\mathbf{x}) = x_1^2 + x_2^2 + \frac{1}{4}x_3^2$$

•



$$f(\mathbf{x}) = 1$$

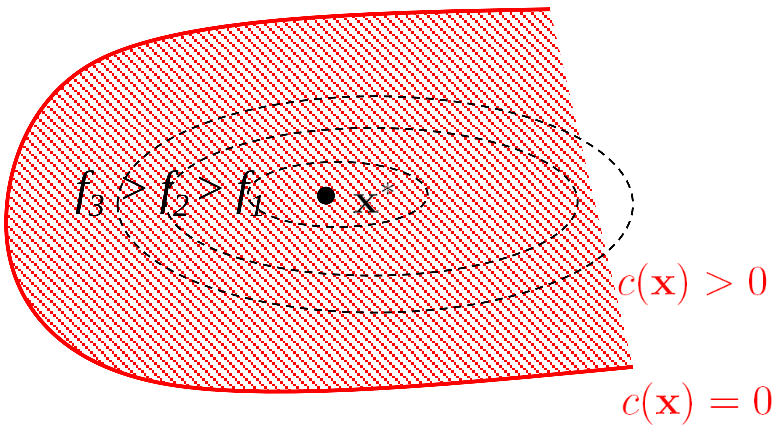
Gradienti omejitev in kriterijske funkcije so linearno odvisni.

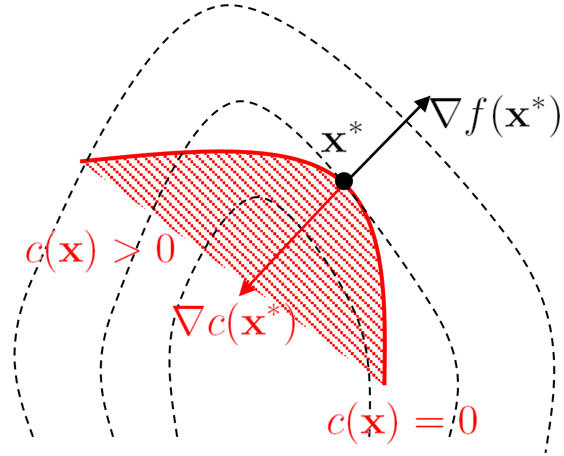
Omejitvene neenačbe

- minimiziraj $f(\mathbf{x})$ glede na: $c_j(\mathbf{x}) \geq 0$ za $j = 1, 2, \ldots, q$

- Gradient funkcije $f(\mathbf{x})$ v točki lokalnega minimuma je enak linerani kombinaciji gradientov omejitve $c_j(\mathbf{x})$, ki je aktivna v ($c_j(\mathbf{x}) = 0$)
- in Lagrange-ovi multiplikatorji morajo biti pozitivni, $\mu_j \geq 0,\ j \in A$

$$\nabla f(\mathbf{x}^*) = \sum_{j \in A} \mu_j^* \nabla c_j(\mathbf{x}^*)$$



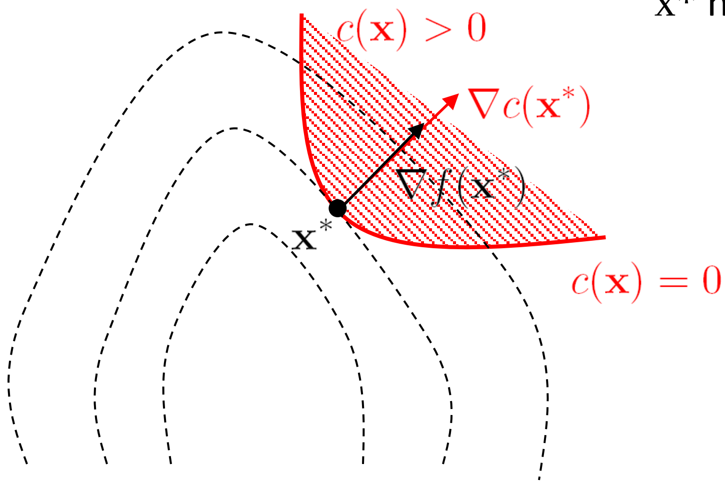


$f_3 > f_2 > f_1$

V točki x* ni aktivnih omejitev, tam je

$$\nabla f(\mathbf{x}) = \mathbf{0}$$

x* ni optimalna, $\mu<0$



$f_3 > f_2 > f_1$

Xje optimalna, $\mu>0$

Lagrangien

- Uvedimo novo funkcijo (Lagrangien)

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) - \sum_{i=1}^{p} \lambda_i a_i(\mathbf{x}) - \sum_{j=1}^{q} \mu_j c_j(\mathbf{x})$$

- Potreben pogoj za obstoj lokalnega minimuma

$$\nabla_x L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{0}$$

in mora biti dovoljena točka(i.e. znotraj omejitev).

- To so znani Karush-Kuhn-Tucker-jevi pogoji

Kvadratično programiranje (QP)

- Zakaj je pomembno?
- Ker generalne probleme rešujemo kot nize z njihovo kvadratično aproksimacijo.
- QP z omejitvami

Minimiziraj $$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{p}$$

glede na linearne omejitve.

- $\mathbf{H}$ je simetrična in pozitivno semidefinitna.

QP z omejitvenimi enačbami

- Minimiziraj $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{p}$

  Glede na: $\mathbf{A}\mathbf{x} = \mathbf{b}$

- $\mathbf{A}$ je $p \times N$ in je polno vrstični rank(p<N)
- Prevedemo v neomejen problkem z eliminacijo spremenljivk:

$$\mathbf{x} = \mathbf{Z}\phi + \mathbf{A}^{+}\mathbf{b}$$

$\mathbf{Z}$ je ničelni prostor $\mathbf{A}$

$\mathbf{A}^{+}$ je pseudo-inverz.

Minimiziraj

$$\hat{f}(\phi) = \frac{1}{2}\phi^T\hat{\mathbf{H}}\phi + \phi^T\hat{\mathbf{p}}$$

$$\hat{\mathbf{H}} = \mathbf{Z}^T\mathbf{H}\mathbf{Z}$$

$$\hat{\mathbf{p}} = \mathbf{Z}^T(\mathbf{H}\mathbf{A}^{+}\mathbf{b} + \mathbf{p})$$

Ta kvadratični problem .

QP omejitvenimi neenačbami

- Minimiziraj $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{p}$

  Subject to: $\mathbf{A}\mathbf{x} \geq \mathbf{b}$

- Prvo verificiraj dopustnost rešitve enačbe $\mathbf{x}^* = -\mathbf{H}^{-1}\mathbf{p}$
- Če je tako, potem je rešitev znana.

  Če je tako potem rešitev leži na meji omejitve tako da nadaljujemo z metodo aktivnega-niza.

  - $\mathbf{x}_k$ trenutna dopustna točka
  - $\mathcal{A}_k$ je indeks aktivnih omejitev v $\mathbf{x}_k$
  - Naslednja točka je dana z

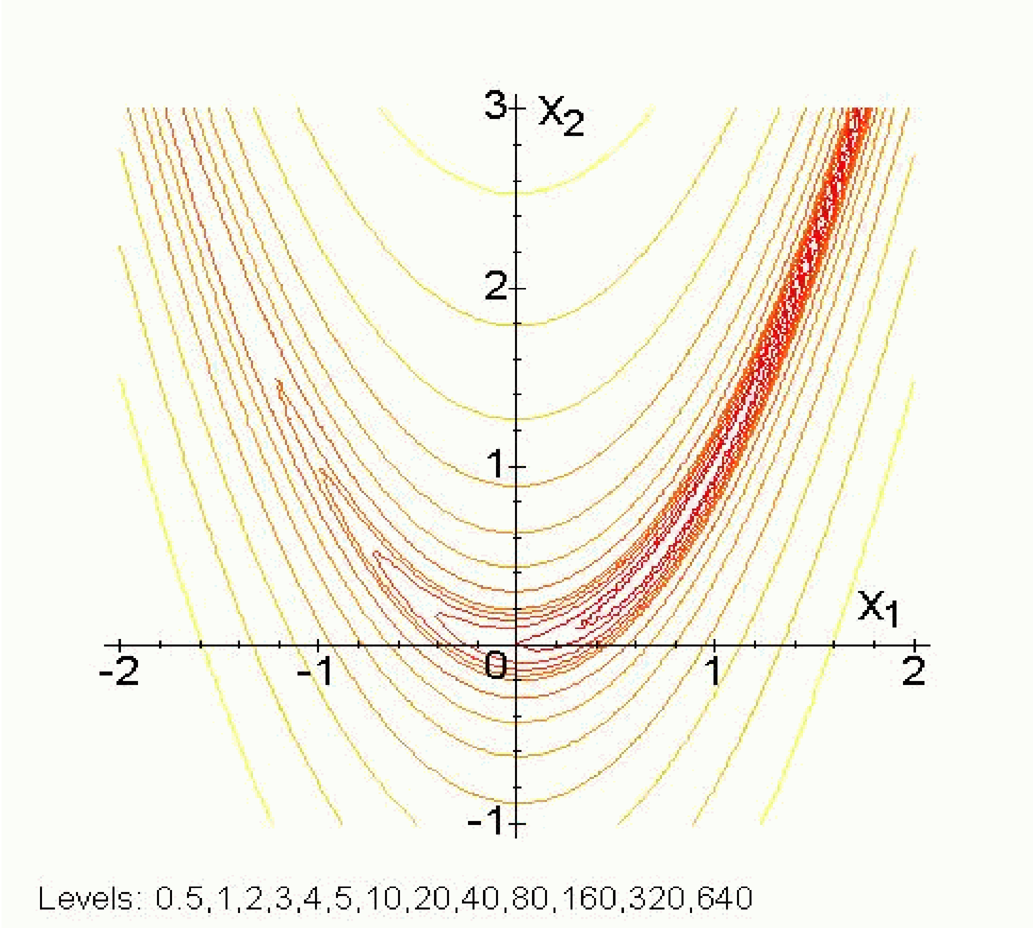
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$$

OPTIMIZACIJSKE TEHNIKE

Neomejena multiparameterska optimizacijska metoda.

Težave s tipičnimi problemi.

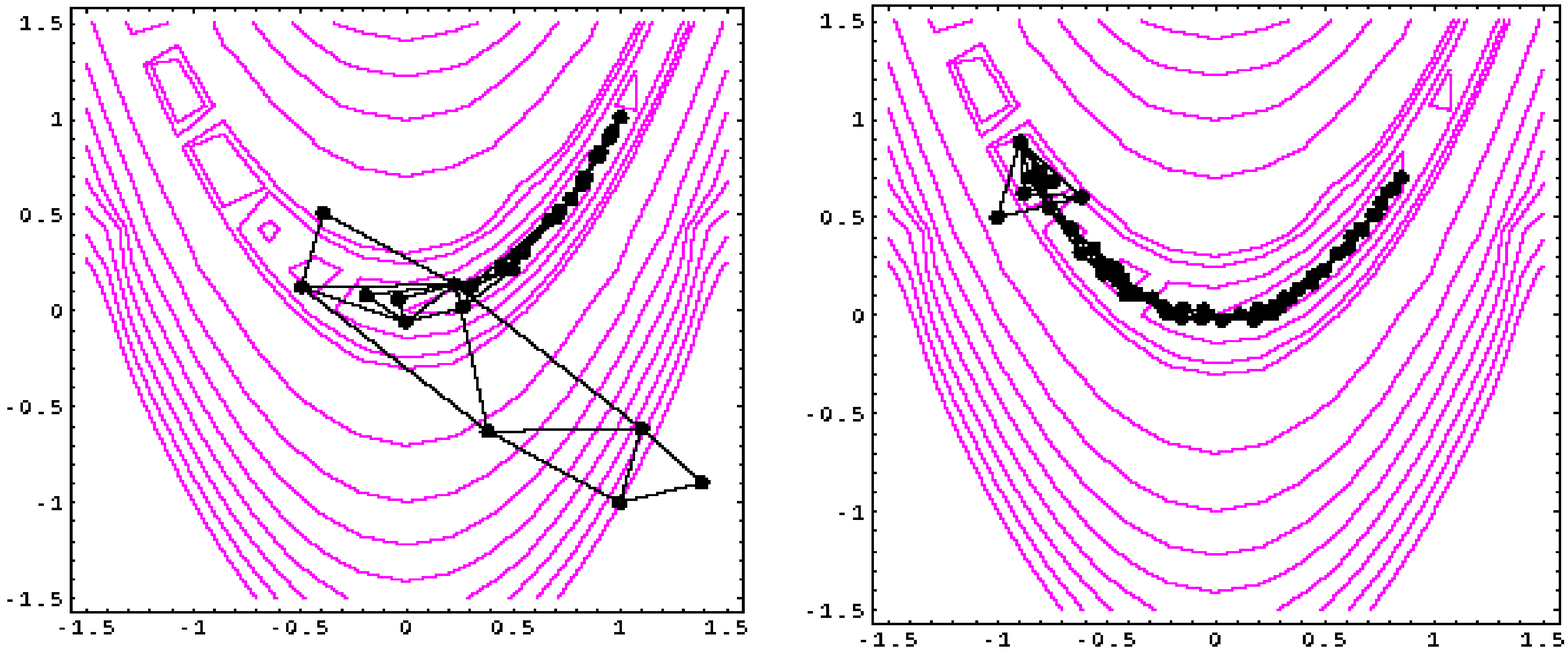
Rosenbrockova “banana” funkcija
$F=100(x_2-x_1^2)^2+(x_1-1)^2$



TEHNIKE NUMERIČNE OPTIMIZACIJE. PRIMER 1

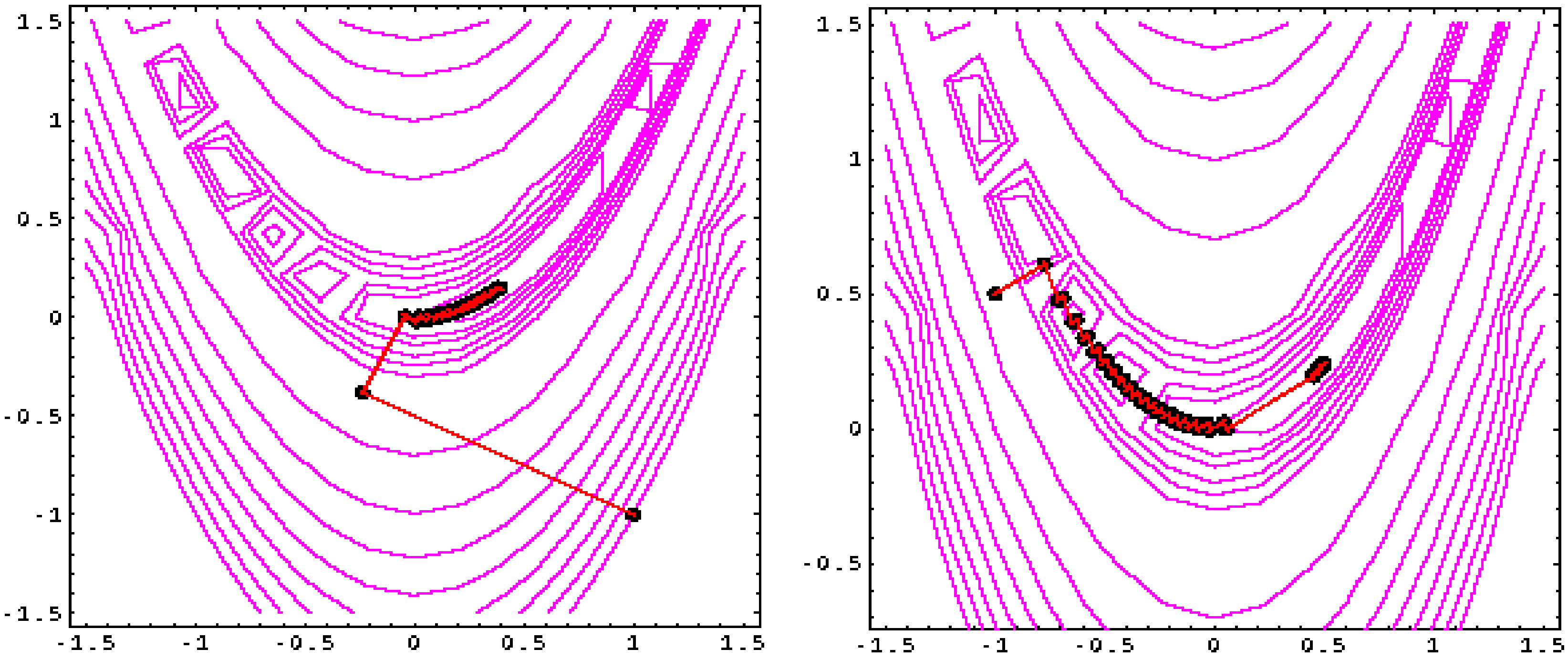
OPTIMIZACIJSKE TEHNIKE

Nelder-Mead-ova sekvenčna metoda Simplex



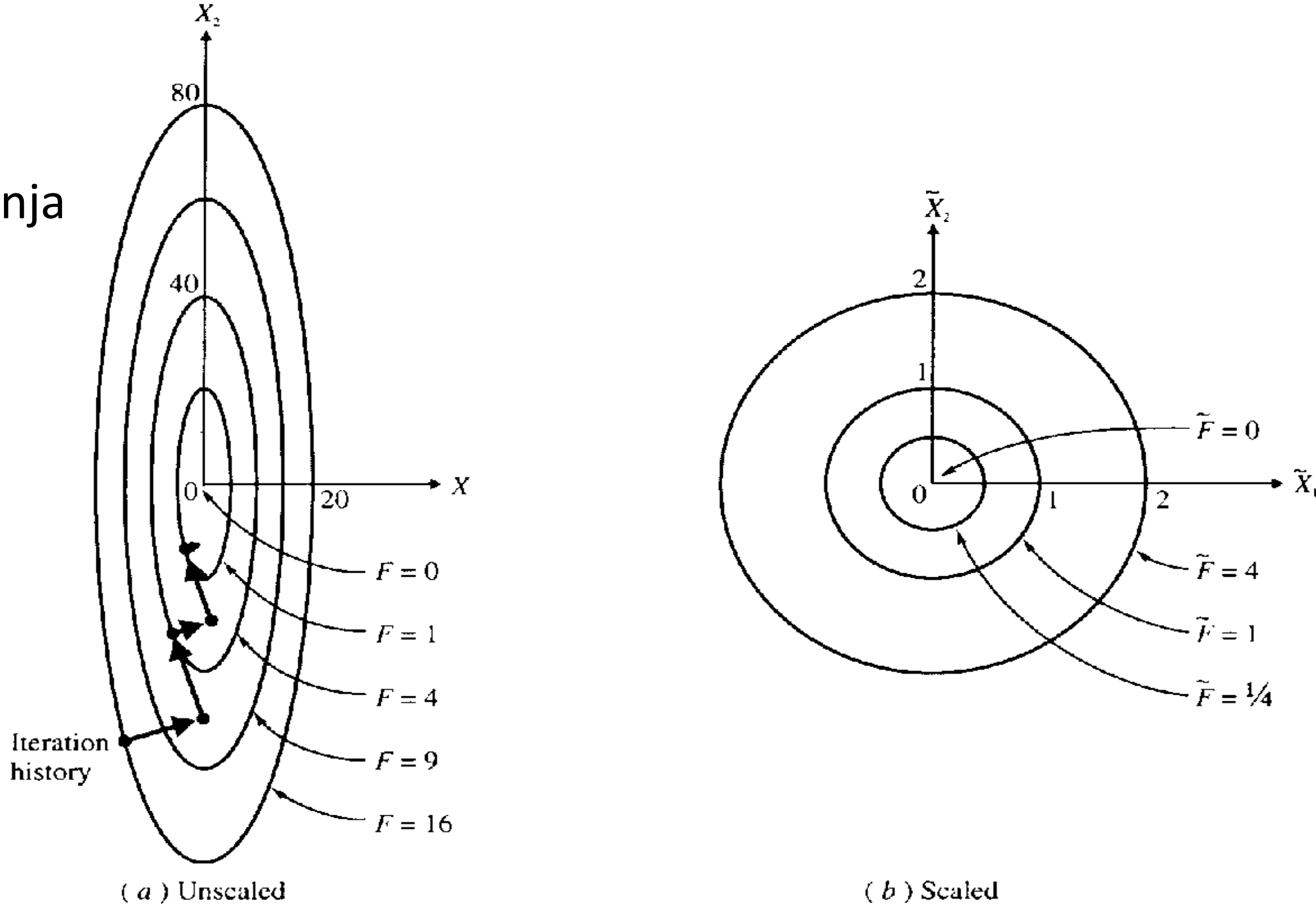
OPTIMIZACIJSKE TEHNIKE

Metoda najhitrejšega spusta



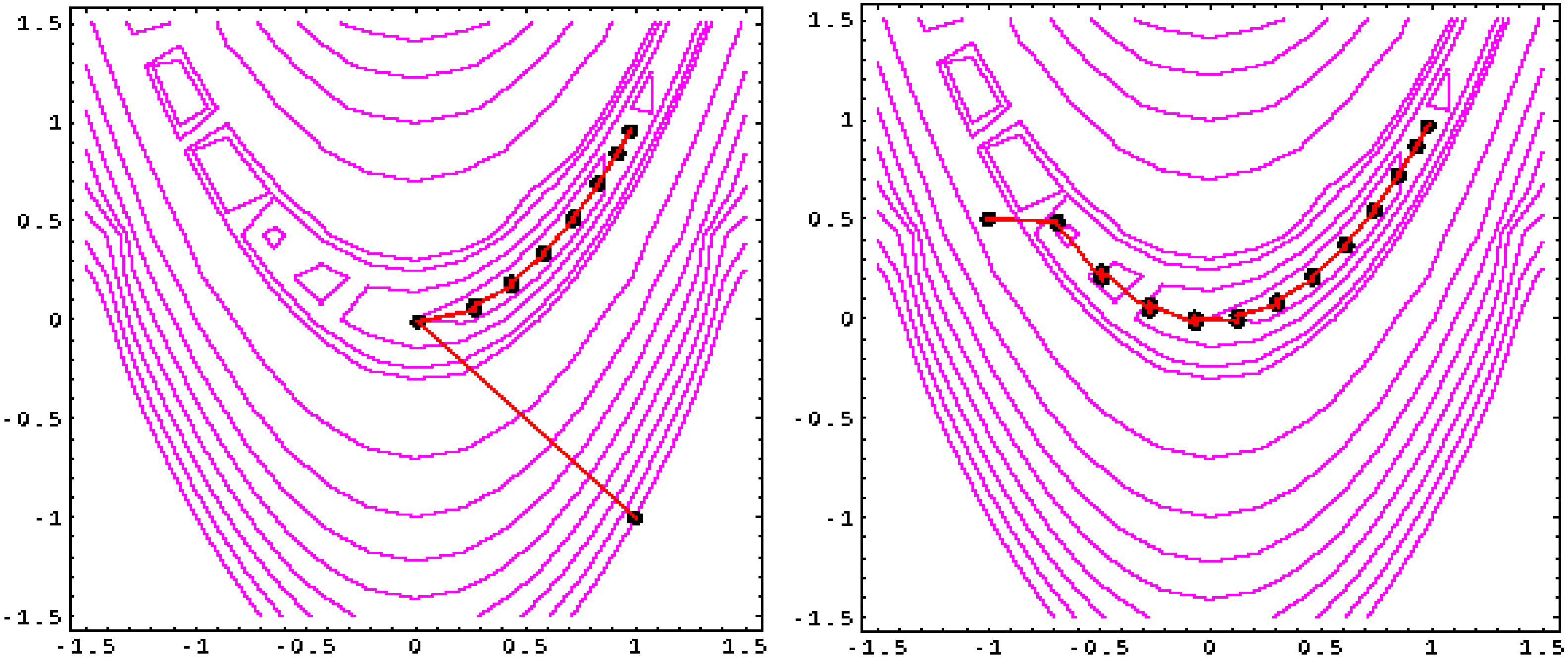
OPTIMIZACIJSKE TEHNIKE

Efekt skaliranja



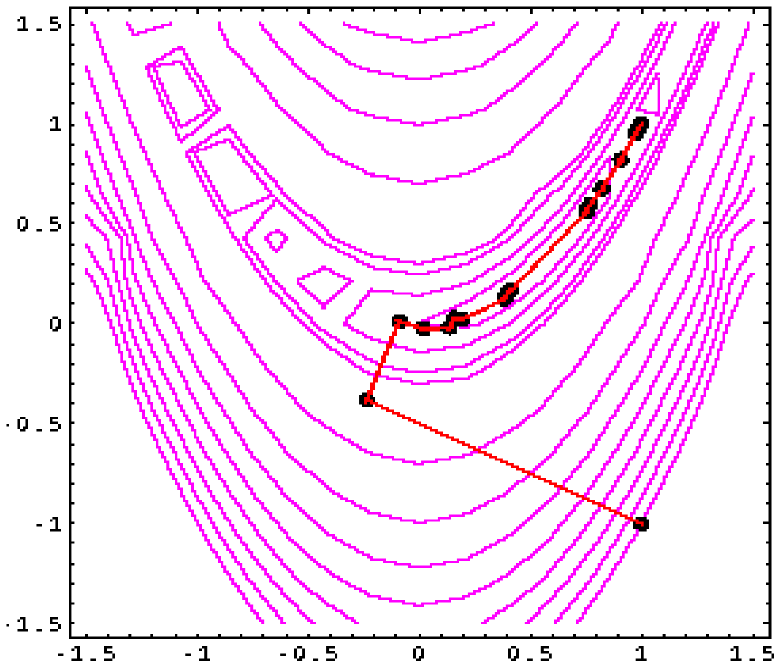
OPTIMIZACIJSKE TEHNIKE

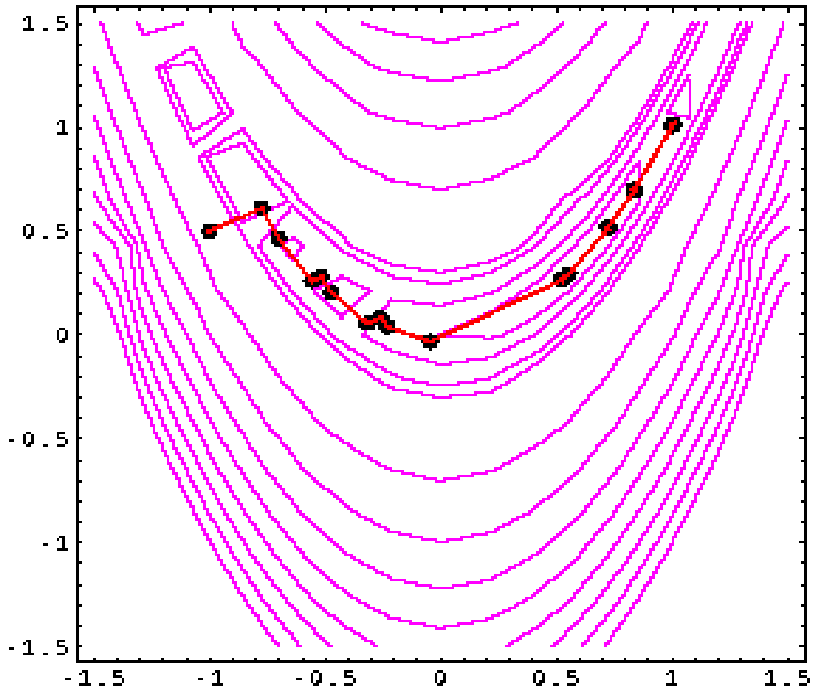
Powellova metoda konjugiranih smeri



OPTIMIZACIJSKE TEHNIKE

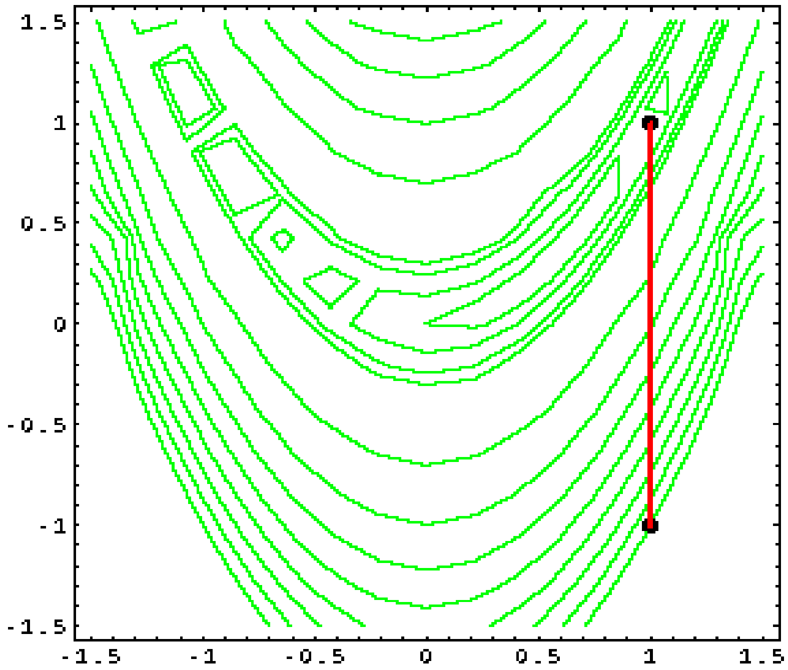
Fletcher-Reeves-ova metoda Konjugiranih Gradientov

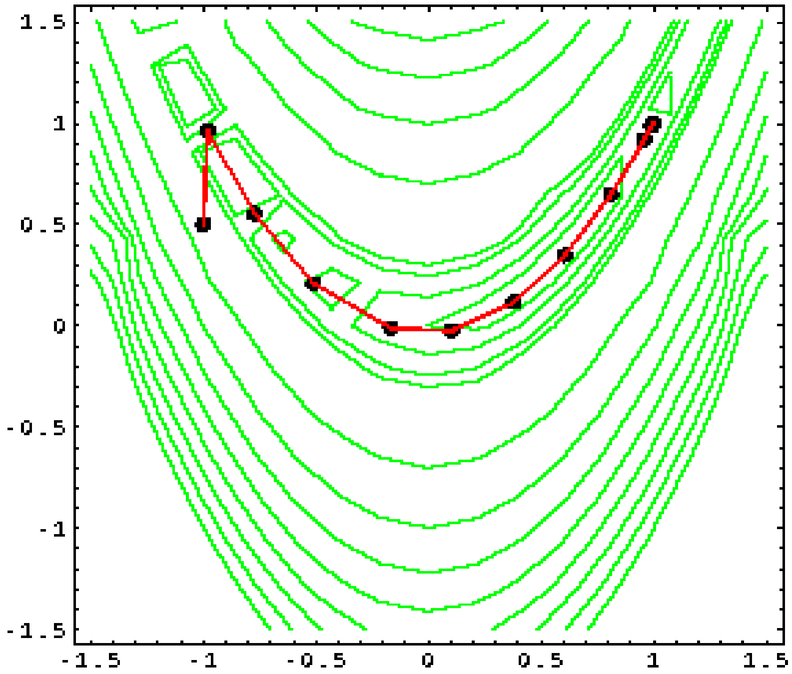




OPTIMIZACIJSKE TEHNIKE

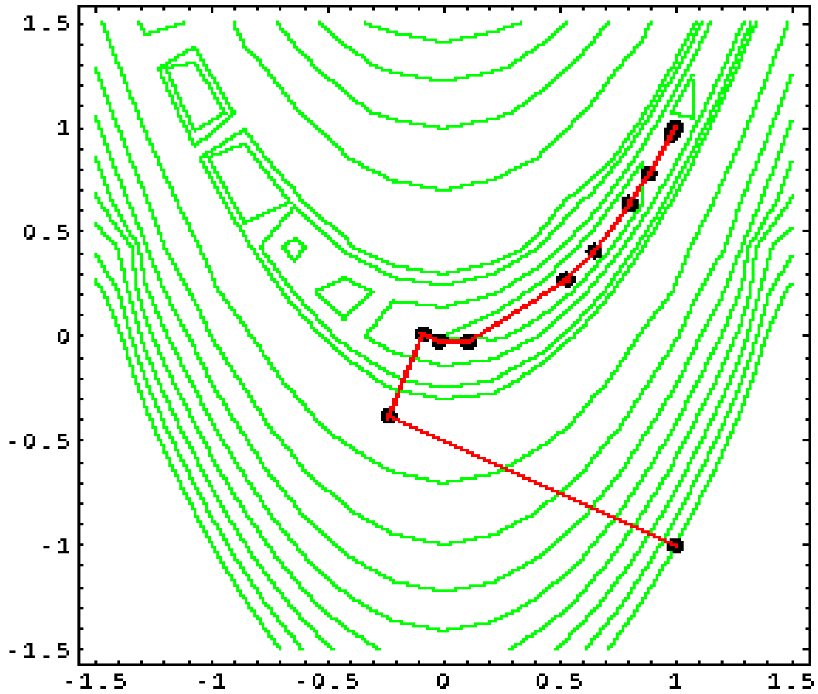
Newtonova Metoda

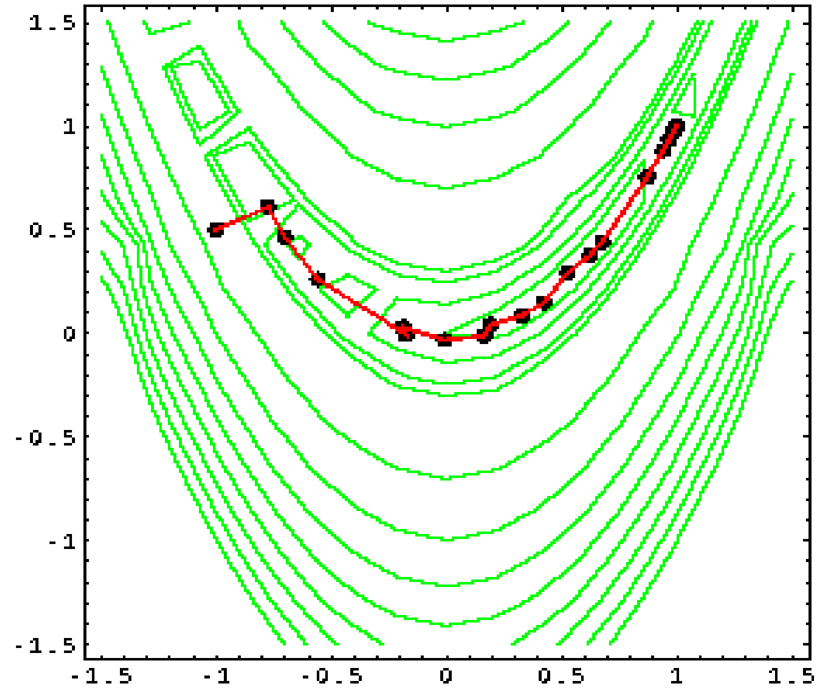




OPTIMIZACIJSKE TEHNIKE

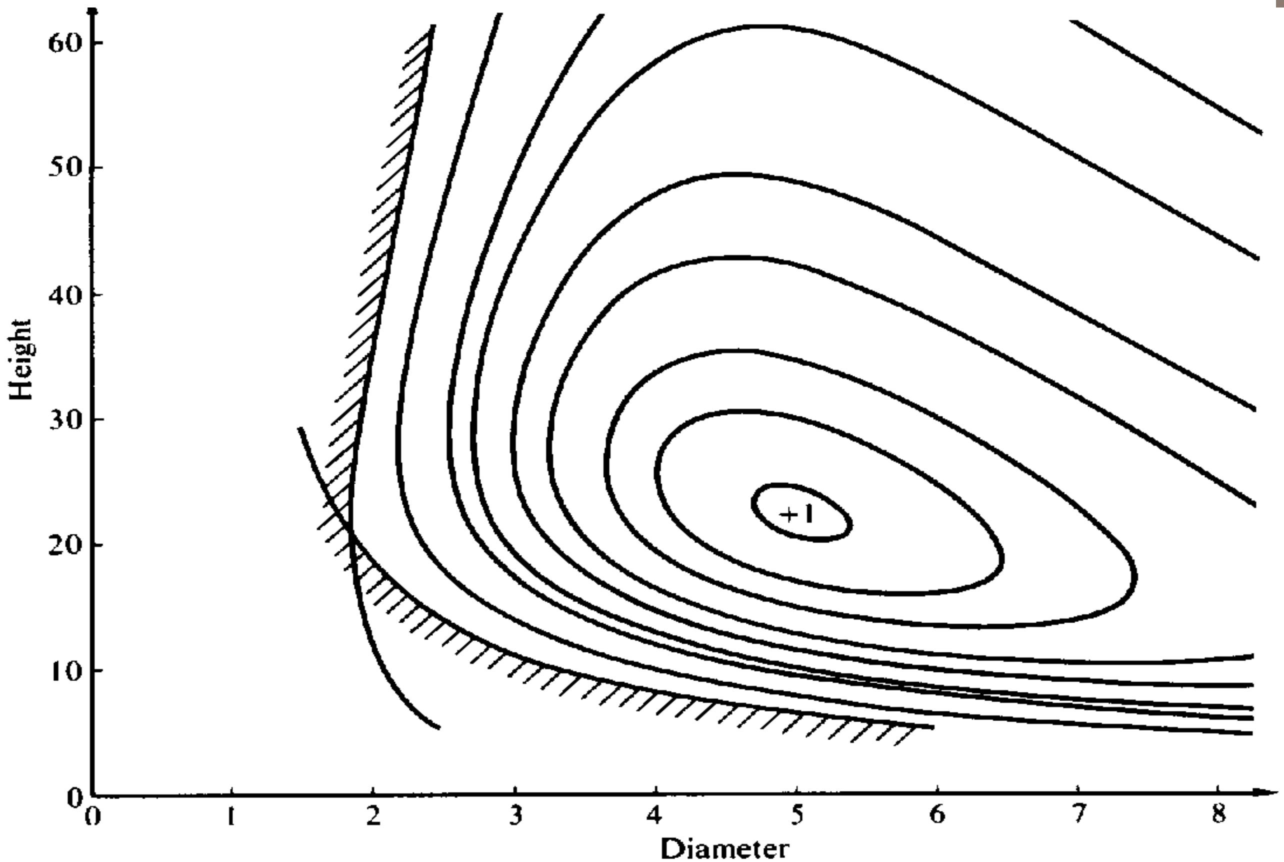
Kvazi Newtonova Metoda





OPTIMIZACIJSKE TEHNIKE

Primer kriterijske funkcije znotraj dopustnega področja.



- Minimiziraj funkcijo

$$f(x,y) = 0.5(\alpha x^2 + y^2)$$

$$-11 \leq x, y \leq 11$$

- Minimiziraj funkcijo

$$f(x,y) = \cos(x)\cos(y)$$

$$-4 \leq x, y \leq 4$$

Začetna točka

Konvergenca k globalnemmu min/max.

Razredi lepih optimizacijskih problemov problems

Primer: $f(x,y) = 0.5(\alpha x^2+y^2)$, $\alpha > 0$

Vsak lokalni, je tudi globalni minimum.

Nezvezna in neodvedljiva funkcija $\Rightarrow$

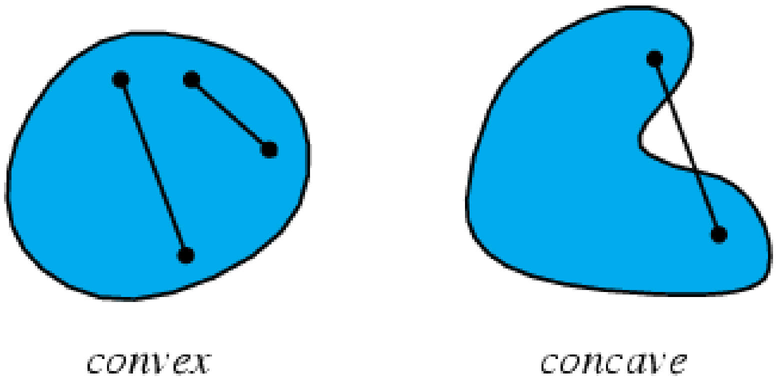
ne moremo izračunati gradienta f $\Rightarrow$

ne moremo uporabiti gradientne metode

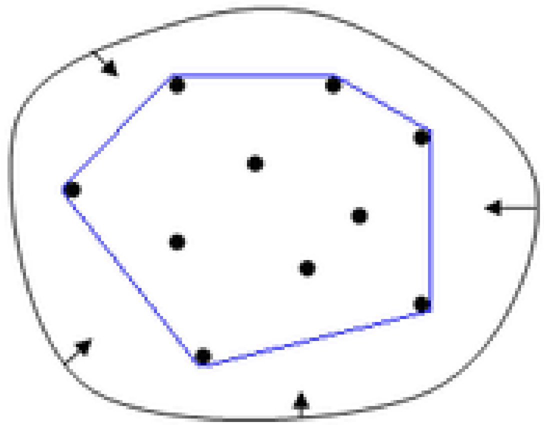
Nelder-Mead Metoda

Druge

Niz C je **konveksen** če je vsaka točka segmenta linije, ki povezuje točki *x* in *y* znotraj *C.*



Konveksno telo - hull niza točk X je minimalni konveksni niz, ki vsebuje točke X.



Simplex ali ***n*-simplex** je konveksno telo reda (*n* +1) . **Simplex** je *n*-dimenzionalno telo trikotne oblike.

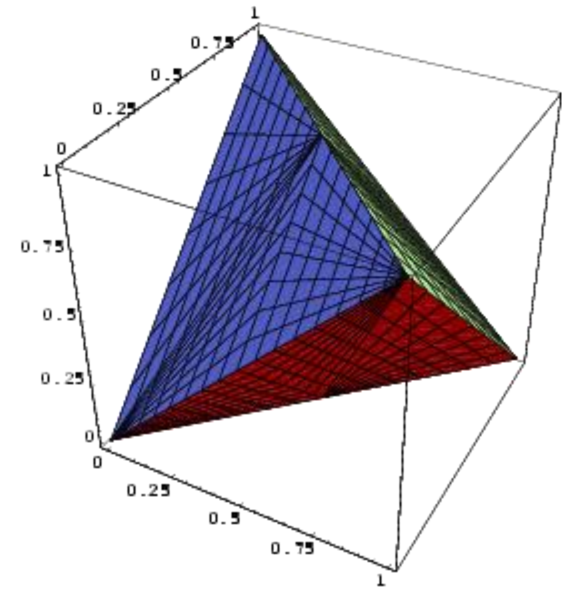
Primer:

1-simplex je linijski segment

2-simplex je trikotnik

3-simplex je tetraeder

4-simplex je pentatope



n = število spremenljivk, n+1 točk

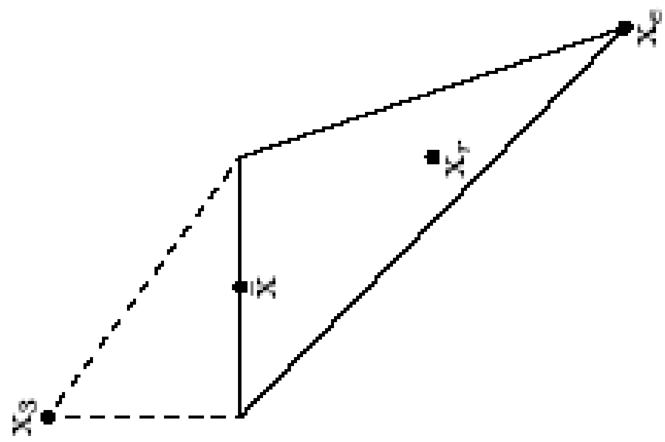
Oblikujemo metodo Simplex za vse točke; konveksno telo pomikamo v smeri proč od najslabše točke, z najslabšo vrednostjo kriterijske funkcije in sicer z operacijami: zrcaljenja, širjenja, krčenja, zmanjševanja reda

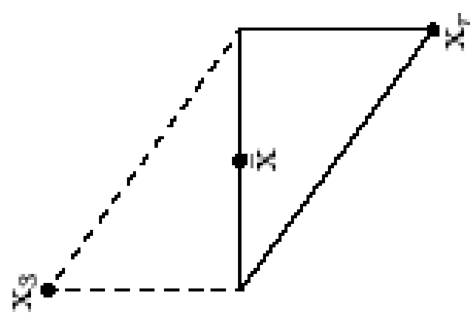
Primer:

2 spremenljivki $\Rightarrow$ 3 točke $\Rightarrow$ simplex je trikotnik

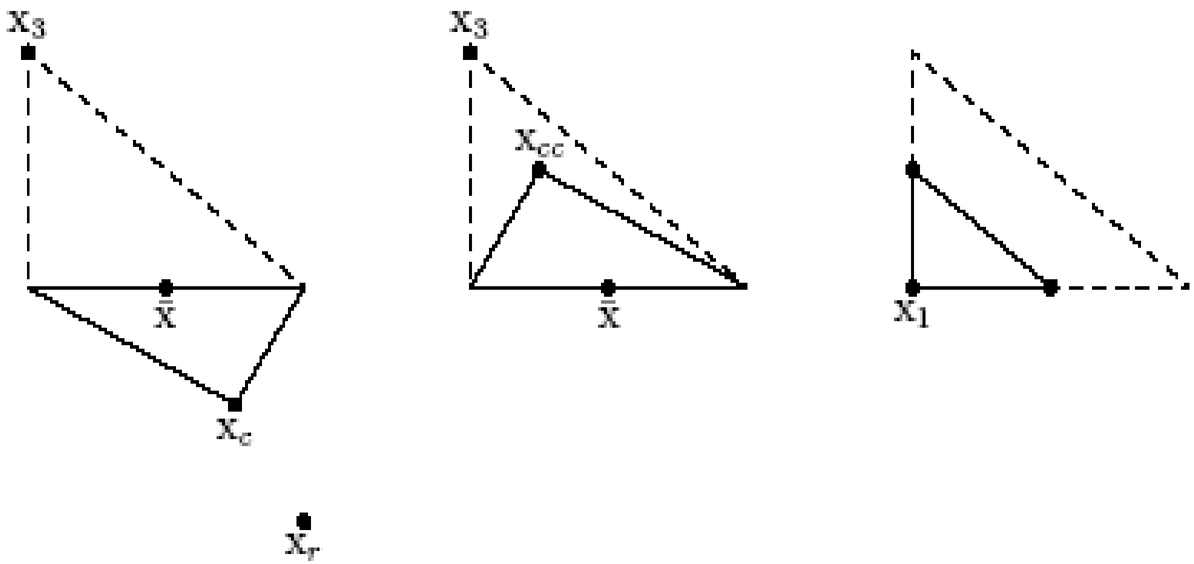
3 spremenljivke $\Rightarrow$ 4 točke $\Rightarrow$ simplex je tetraeder

Nelder-Mead metoda – zrcaljenje, širjenje

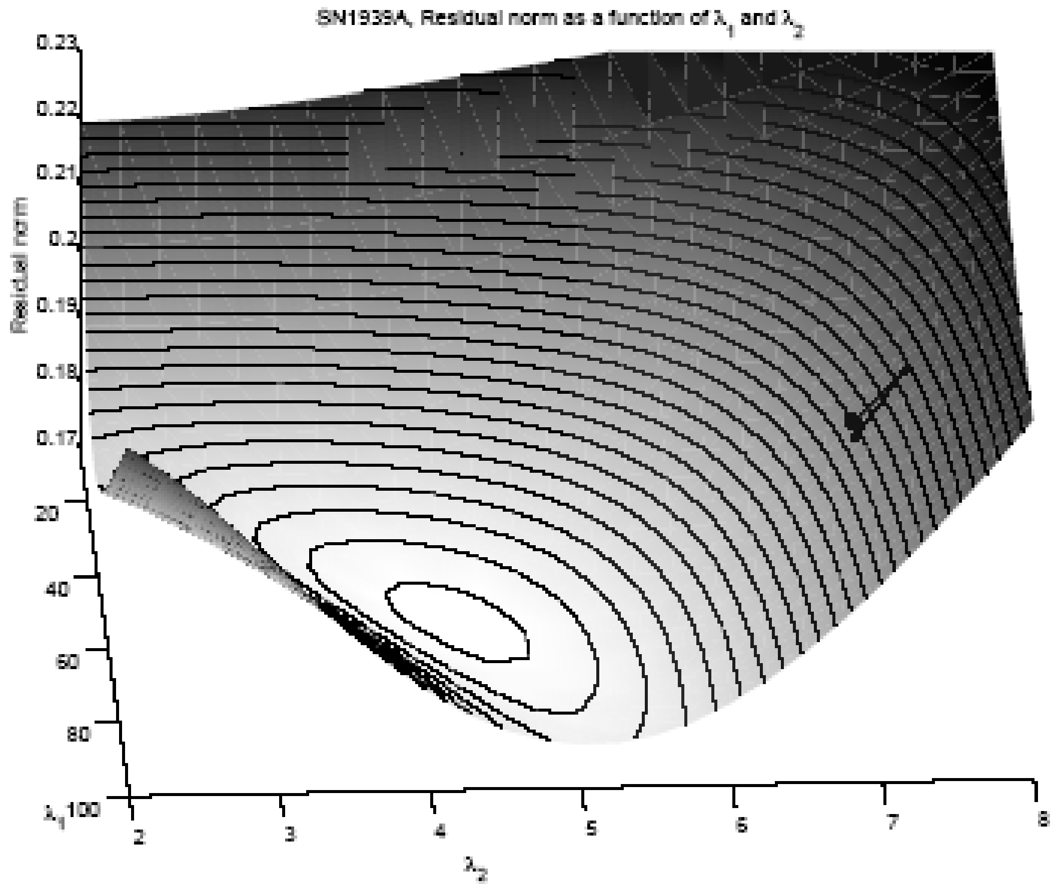




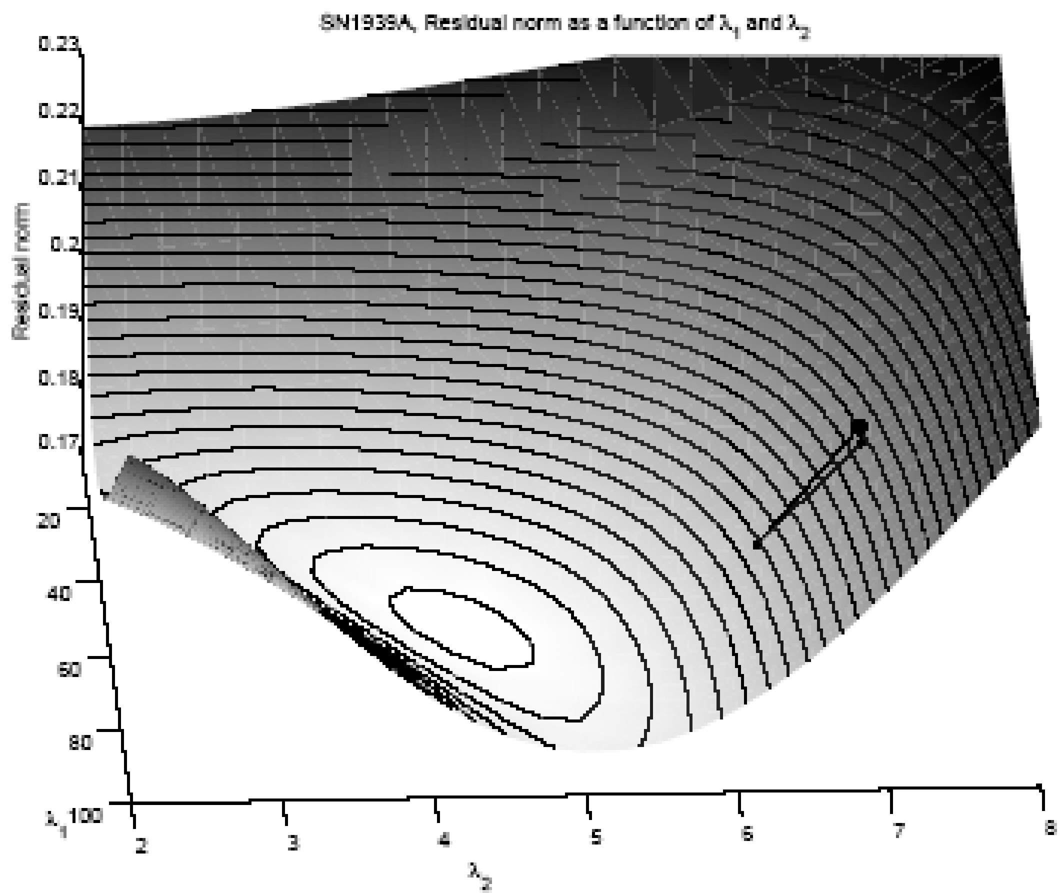
Nelder-Mead metoda – zrcaljenje, krčenje



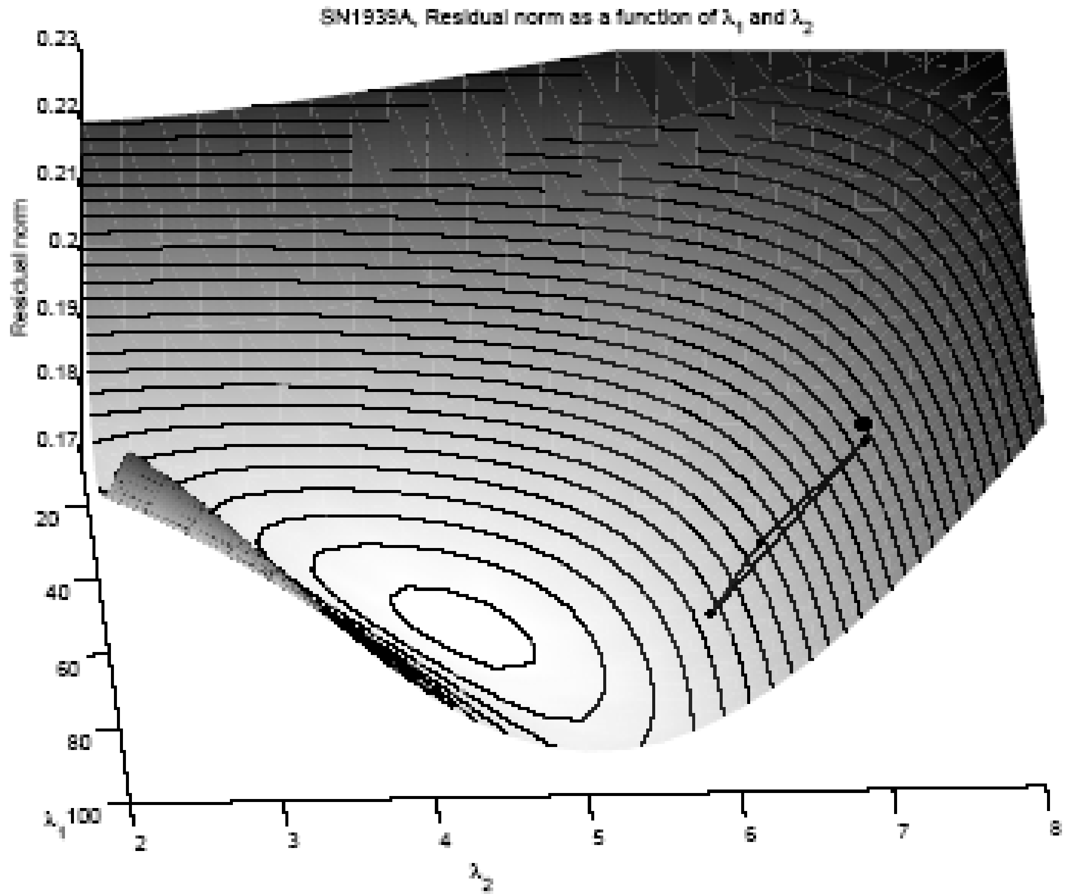
Minimizacija v Matlabu: primeri



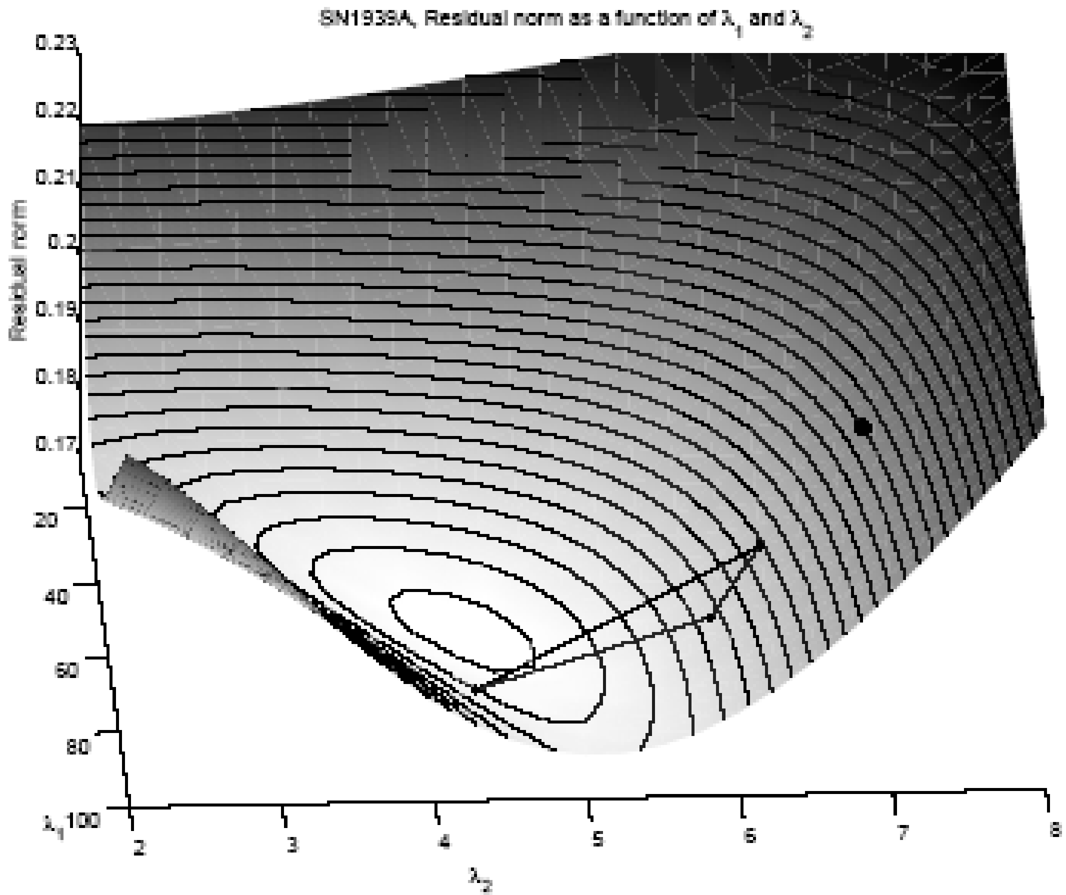
Minimizacija v Matlabu: primeri



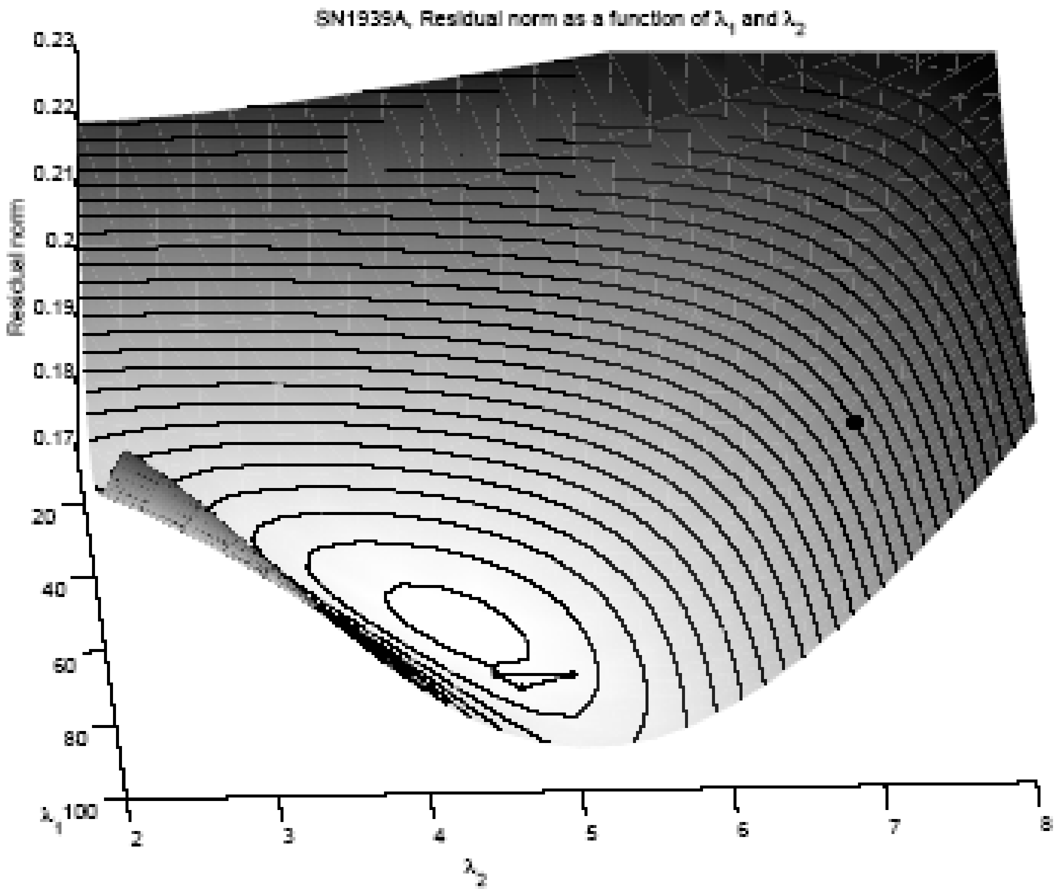
Minimizacija v Matlabu: primeri



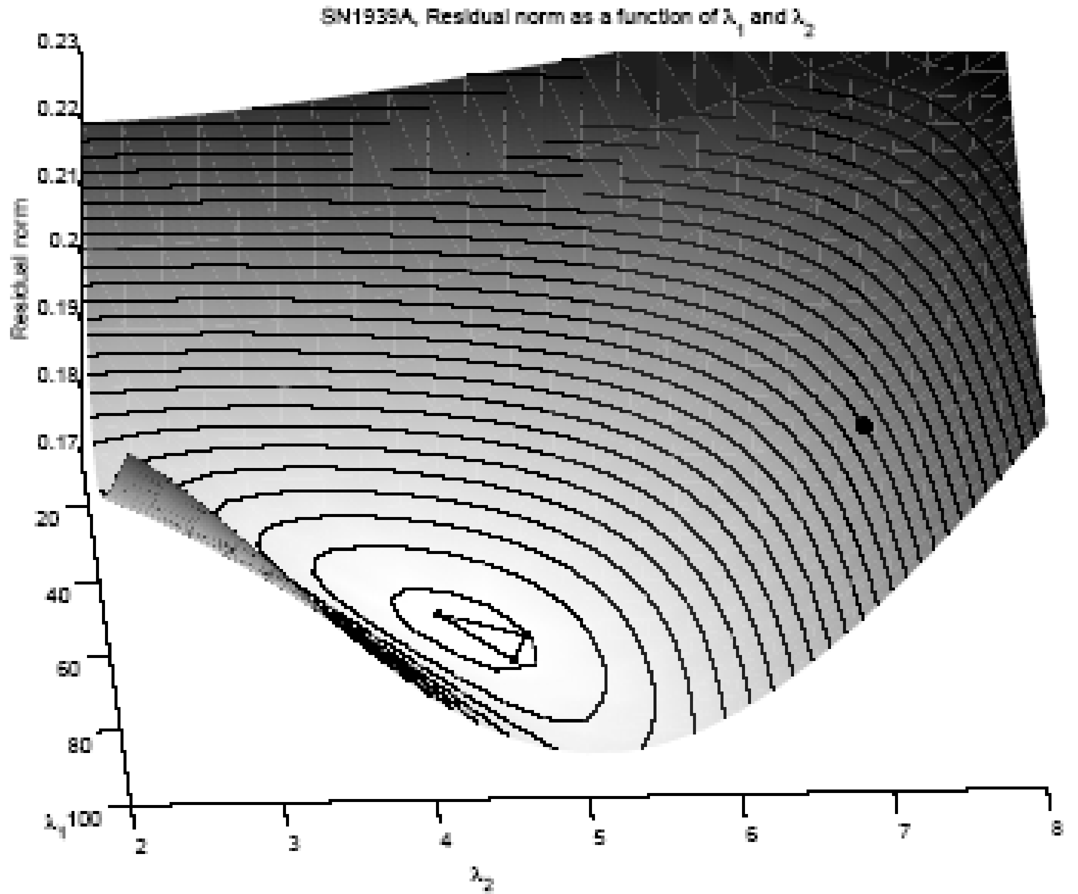
Minimizacija v Matlabu: primeri



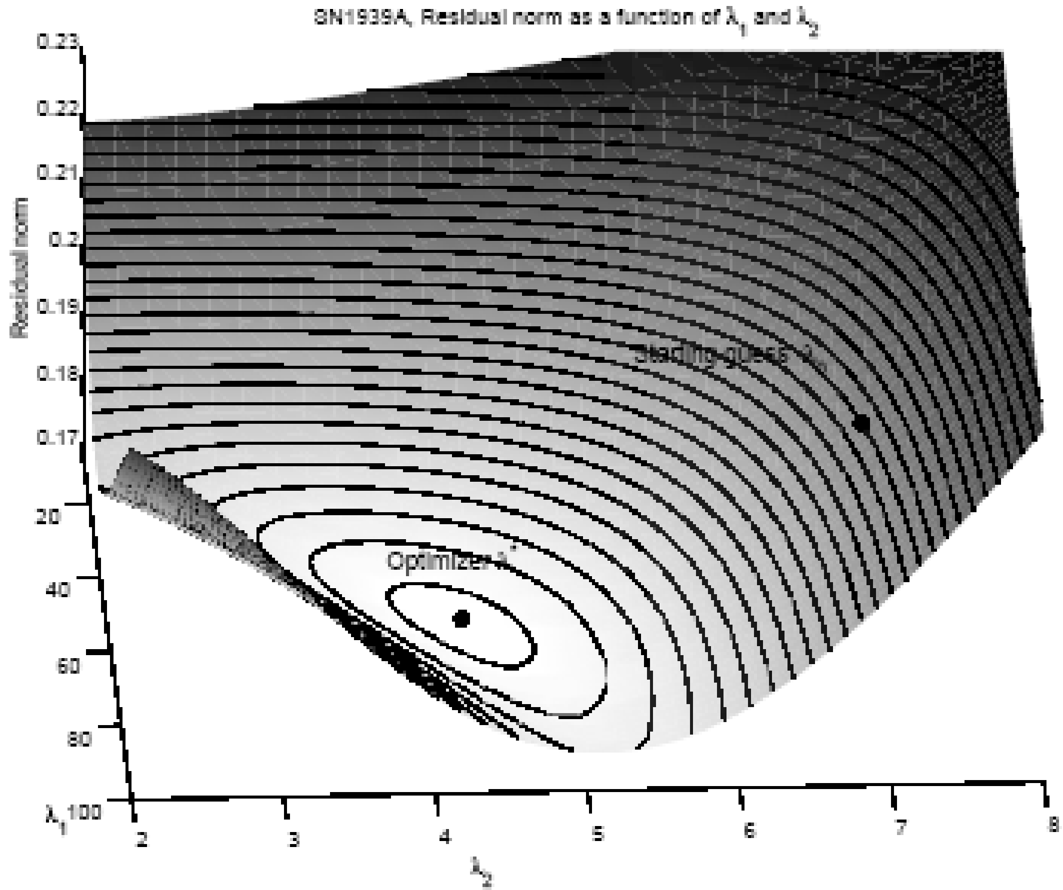
MINIMIZACIJA V MATLABU: PRIMERI



MINIMIZACIJA V MATLABU: PRIMERI



MINIMIZACIJA V MATLABU: PRIMERI



Matematične osnove Optimizacije pomenijo Kuhn Tucker-jeve potrebne pogoje

Primer problema matematičnega programiranja

Minimiziraj: $y(x)$

Glede na: $f_i(x) \leq 0$ za $i = 1, 2, ..., h$

$f_i(x) = 0$ za $i = h+1, ..., m$

$y(x)$ in $f_i(x)$ so dvakrat odvedljive realne funkcije.

Kuhn Tucker potrebni pogoji

Lagrangeova Funkcija
– pretvori omejen problem v neomejenega

$$L(x,\lambda) = y(x) + \sum_{i=1}^{h} \lambda_i \left[f_i(x) + x_{n+i}^2 \right] + \sum^{m} \lambda_i f_i(x)$$

λ_i so Lagrangeovi multiplikatorji

x_{n+i} so pomožne spremenljivke uporabne pri pretvorbi omejitvenih neenačb v enačbe.

Lagrangeovi Multiplikatorji

Uporabljeni kot:

- Nedoločeni multiplikatorji – pomnoži omejitve z λ_i in jih prišteje k $y(\mathbf{x})$
- Spremenljivke - $L(x,\lambda)$
- Konstante – numerične vrednosti v točki optimuma.

Lagrangeovi Multiplikatorji

optimiziraj: $y(x_1, x_2)$

glede na: $f(x_1, x_2) = 0$

$$dy = \frac{\partial y}{\partial x_1} dx_1 + \frac{\partial y}{\partial x_2} dx_2$$

$$0 = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 \longrightarrow dx_2 = -\frac{\frac{\partial f}{\partial x_1}}{\frac{\partial f}{\partial x_2}} dx_1$$

Lagrangeovi Multiplikatorji

$$dy = \frac{\partial y}{\partial x_1} dx_1 - \frac{\partial y}{\partial x_2} \frac{\dfrac{\partial f}{\partial x_1}}{\dfrac{\partial f}{\partial x_2}} dx_1$$

Preuredi delne odvode v drugem izrazu

Lagrangeovi Multiplikatorji

$$dy = \left[\frac{\partial y}{\partial x_1} + \left(\frac{-\dfrac{\partial y}{\partial x_2}}{\dfrac{\partial f}{\partial x_2}} \right) \frac{\partial f}{\partial x_1} \right] dx_1$$

$$dy = \left[\frac{\partial y}{\partial x_1} + \lambda \frac{\partial f}{\partial x_1} \right] dx_1$$

$(\quad) = \lambda$

Prispeva razmerje delnih odvodov v ( ) kot Lagrangeove multiplikatorje λ

Lagrangeovi multiplikatorji λ so razmerje delnih odvodov v točki optimuma.

Lagrangeovi multiplikatorji λ

$$dy = \frac{\partial (y + \lambda f)}{\partial x_1} dx_1 = 0$$

Določi L = y +λf , neomejeno funkcijo

$$\frac{\partial L}{\partial x_1} = 0 \qquad \textbf{in po istem postopku} \qquad \frac{\partial L}{\partial x_2} = 0$$

Interpretirajmo L kot neomejeno funkcijo, delni odvodi, katerih vrednost je nič, predstavljajo potrebne pogoje te neomejene funkcije.

Lagrangeovi multiplikatorji λ

Optimirajmo: $y(x_1,x_2)$

glede na: $f(x_1,x_2) = b$

Glede na zahteve dobimo:

$$\frac{\partial y}{\partial b} = -\lambda$$

Oz po sposameznih i :

$$\frac{\partial y}{\partial b_i} = -\lambda_i$$ delni strošek (cena na enoto b_i)

Splošne zahteve linearnega programiranja

kriterijska funkcija:

Maksimiraj: $c_1x_1 + c_2x_2 + ... + c_nx_n = p$ (4-1a)

Omejitvene enačbe:

glede na: $a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \leq b_1$ (4-1b)

$$
\begin{array}{lll}
a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n & \leq & b_2 \\
\cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots \\
a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n & \leq & b_m
\end{array}
$$

$x_j \geq 0$ za $j = 1,2,...n$ (4-1c)

Pregled strojnega učenja

Vsebina

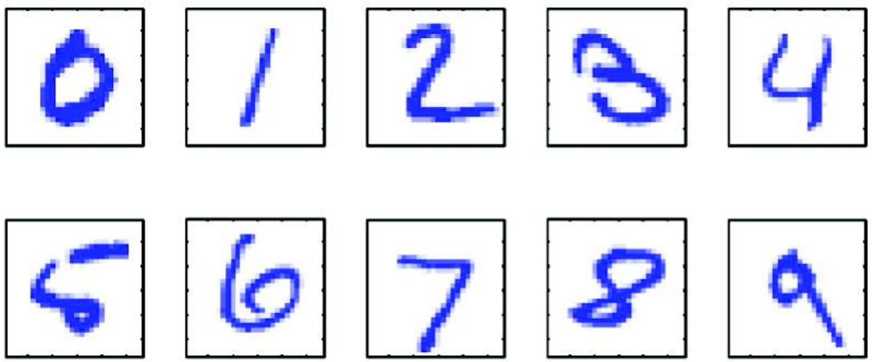
1. Kaj je strojno učenje (Machine Learning - ML)?
2. Tipi obdelave informacij

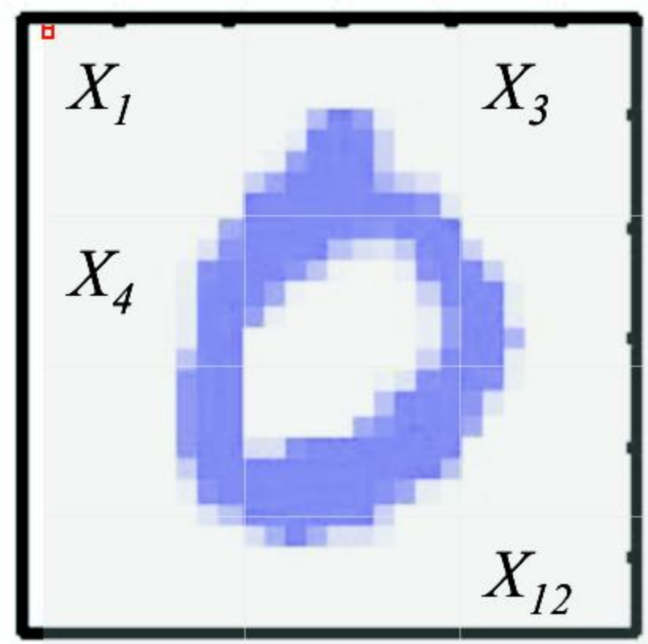
    Rešeni problemi

    1. Regresija
    2. Klasifikacija
    3. Grozdenje
    4. Verjetnostno sklepanje
3. Nova gibanja
    1. Popoln Bayesov pristop
4. Povzetek

Klasifikacija: Prepoznava števil

- Vhod (X_i): Značilke slike
- Izhod (Y): Labele razredov {y_0, y_1,.y_9}





Široka variabilnost

- Ročno ustvarjena pravila povzročajo veliko število pravil in izjem
- Bolje je imeti stroj, ki se uči iz velike učne množice

Značilke (X_i):

X_i $i=1,..,12$

$$\left.\begin{array}{l} x_i^0=0\text{-}10\% \\ x_i^1=10\text{-}20\% \\ \dots \end{array}\right\} Val(X_i)=10$$

Število parametrov=10^{12} - 1

ML in PR

- ML izvira iz računalništva
- PR izvira iz elektrotehnike
- Različne plati istega področja
- Prisotne več kot 50 let
- Oživitev Bayesovih metod
  - Zaradi razpoložljivih računalniških metod

Zgodovina ML in DAR

- Vhod je perceptualen (slike, tekst)

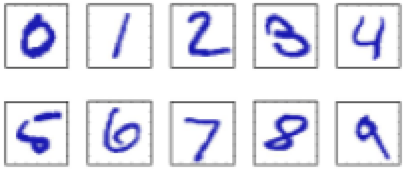
– Ni popolnega ujemanja: zahtevane so verjetnostne metode

- ML najprej s slikami/tekstom

– Perceptroni/nevronske mreže
  - – Prepoznava števil
- – SVM
  - – Ročno napisane poštne številke
- – Skriti modeli Markova (Hidden Markov Models)
  - – Prepoznava govora
- – Pogojna naključna polja:
  - – Segmentacija slik , analiza teksta (NE Tagging)

Kaj je strojno učenje?

- Problemi vključujejo negotovost
  - učenje iz podatkov in ne iz kode
- Preobremenitev z informacijami
  - Velike količine učnih podatkov
  - Omejitve človeške kognitivne sposobnosti



- Spremenljivi podatkovni tokovi
  - Iskalnik se mora neprestano prilagajati
  - Rokopisi in stili pisav niso enaki kot pred 10/25 leti
- Strojno učenje povzroča napredek v programiranju

Tipologija problematike

1. Klasifikacija (razredne oznake)
   – OCR, prepoznava števil
2. Regresija (zvezne vrednosti)
   – rangiranje spletnih strani
3. Kolektivna klasifikacija
   – segmentacija slik
   – zaporedno
     - prepoznava besed, POS Tagging, prepoznava govora
4. Pridobivanje verjetnosti
   – najboljše ujemanje v podatkovni bazi, redkost podatkov

Aplikacije

- Programiranje računalnikov:
  - za naloge, ki jih človek zna dobro opraviti, vendar jih je težje opisati z algoritmom
- Primaren način gradnje informacijskih sistemov visokih zmogljivosti
  - iskalniki
  - adaptivni, personalizirani uporabniški vmesniki
  - znanstvene aplikacije (računalniška znanost)

Uspešne ML aplikacije

- Prepoznava govorjenih besed
  - govorcu prilagojene strategije za prepoznavo primitivnih zvokov, fonemov in besed iz govora
  - Nevronske mreže in metode za učenje HMM za prilagajanje govorcem, bedenjakom in mikrofonskim karakteristikam

Regresija: Rangiranje spletnih iskanj

Vhod (X_i): (Query-URL) par

Izhod (Y): Vrednost relevance

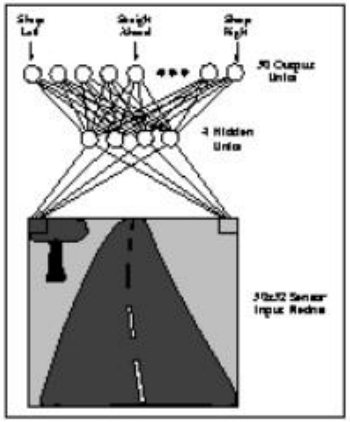
Značilke (Google jih uporablja prek 200)

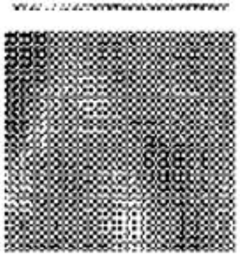
- Log frequency of query word in anchor text
- Query word in color on page
  # of images on page
  # of (out) links on page
- PageRank of page
- URL length
- URL contains “~”
- Page length

Uspešne ML aplikacije

- Učenje vožnje z avtomatskim vozilom
  – Učenje računalniško vodenih vozil za avtomatsko zavijanje
  – prilagajanje hitrosti omejitvam
  – Povezava upravljanja volana s slikami

ALVINN [Pomerleau] drives 70 mph on highways





Prepoznava rokopisa

- **Naoga *T***

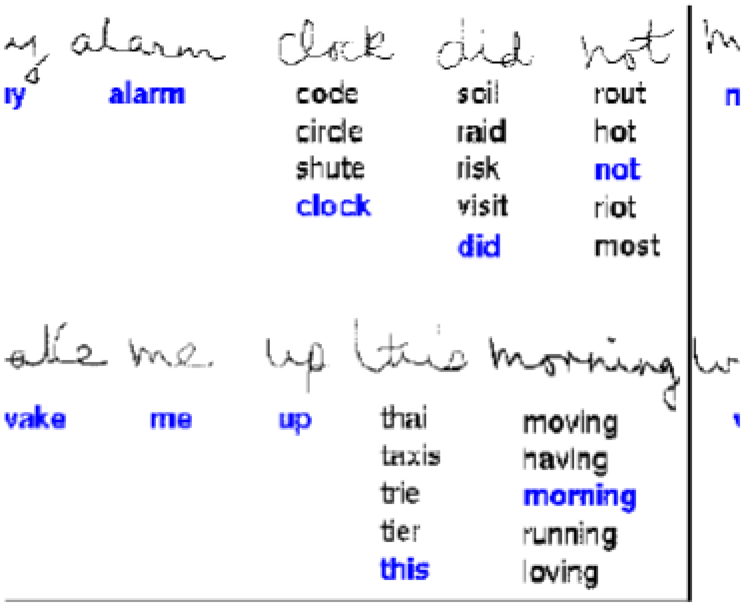
  – prepoznava in klasifikacija besed napisanih na slike

- **Merilo učunkovitosti *P***

  – odstotek pravilno razvršččenih besed

- **Izkušnje učenja *E***

  – baza že klasificiranih napisanih besed



ML pristop

- Zbiranje podatkov
- Izbira modela
  - Porazdelitev verjetnosti za modeliranje procesa
- Ocena parametrov
  - Vrednosti/porazdelitve
- Sklepanje
  - Iskanje natančnega ali približnega odgovora

Problemi, za katere je uporabljeno strojno učenje

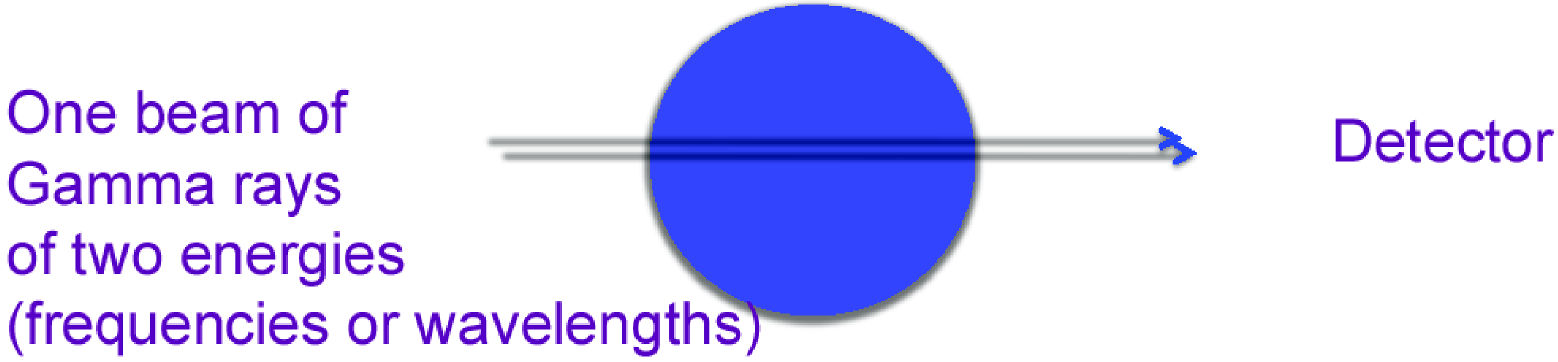
1. Klasifikacija
2. Regresija
3. Grozdenje (Data Mining)
4. Modeliranje/Sklepanje

Primer klasifikacije

- Cevovodi (pod morjem)
- Neinvazivno merjenje razmer
    - nafta, voda in plin
- Imenuje se trofazni Nafta/Voda/Plin proces

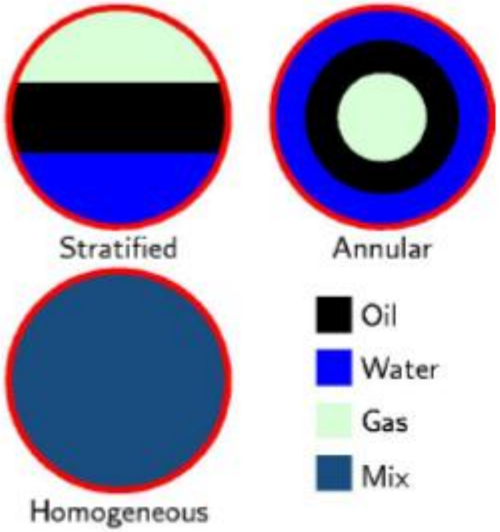
Dvojno energijska gama densitometrija

- Žarek gama žarkov je usmerjen skozi cev
- Slabljenje podaja informacijo o gostoti materiala cevi
- En sam žarek ne zadostuje
  - Dve stopnji prostosti: delež nafte, delež vode



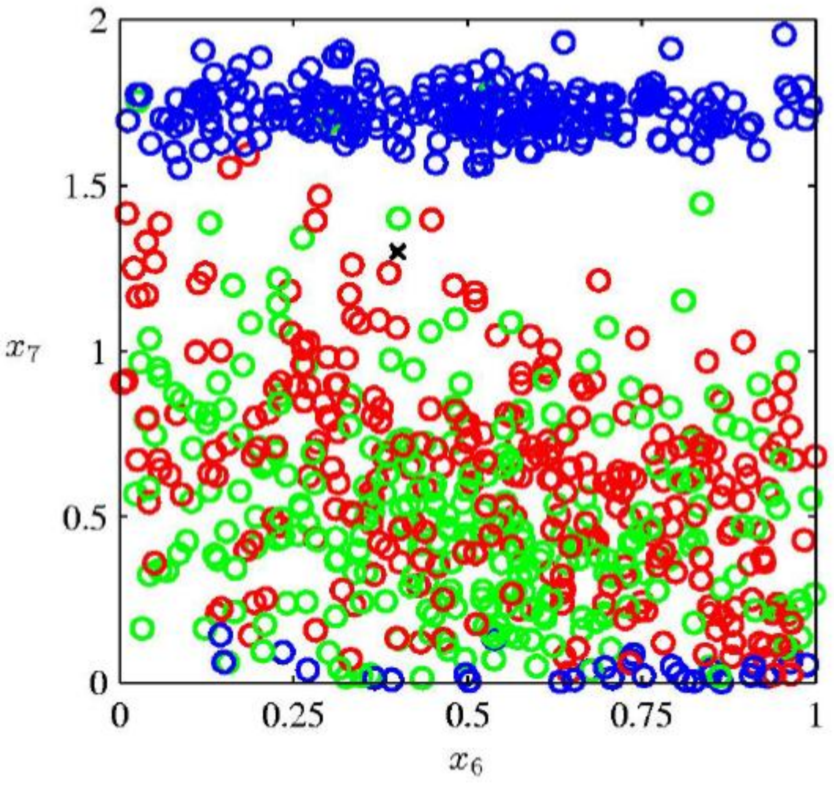
Predikcija

1. Napoved volumskih deležev nafta/voda/plin
2. Napoved geometrijske konfiguracije vseh treh faz
   - 12 značilk
     - deleži nafte in vode vzdolž poti
   - učenje razlikovanja med podatki



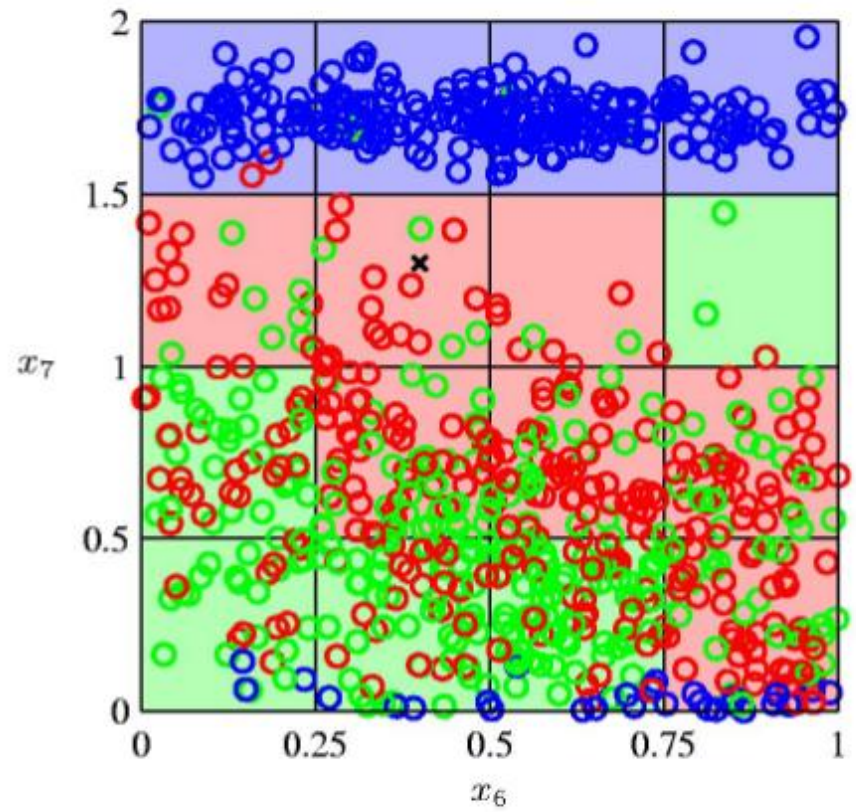
Prostor značilk

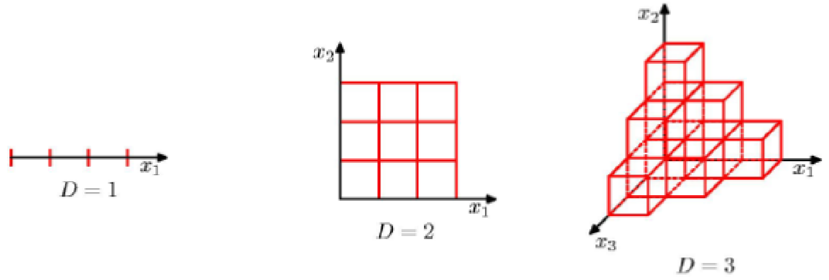
- Trije razredi (Stratified,Annular,Homogeneos)
- Prikazani sta 2 spremenljivki
- 100 točk
  - V kateri razred spada x?



Celična klasifikacija

- Preprosti pristop glasovanja na osnovi pripadnosti celici bo neuspešen zaradi eksponentnega naraščanja števila celic z naraščanjem dimenzije
- Malo točk v celicah





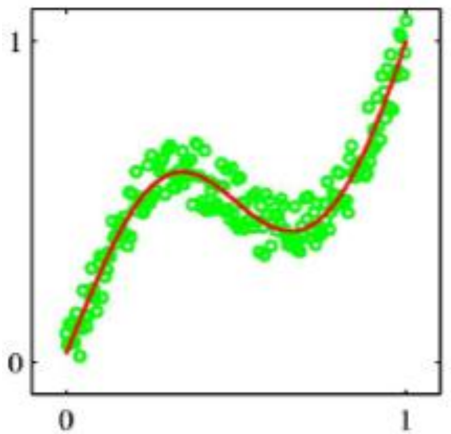
Priljubljeni statistični modeli

- Splošni
  - Naive Bayes
  - Mixtures of multinomials
  - Mixtures of Gaussians
  - Hidden Markov Models (HMM)
  - Bayesian networks
  - Markov random fields

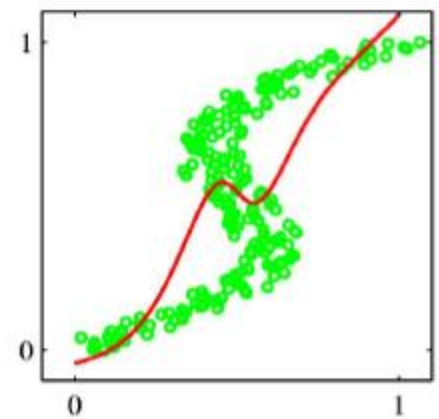
- Diskriminativni
  - Logistic regression
  - SVMs
  - Traditional neural networks
  - Nearest neighbor
  - Conditional Random Fields (CRF)

Regresija

Ustrezni inverzni
Problem z inerzom
x in *t*



ultat
prilagajanja dvonivojske nevronske
mreže z metodo najmanjših
kvadratov



Zelo slabo
prilagajanje

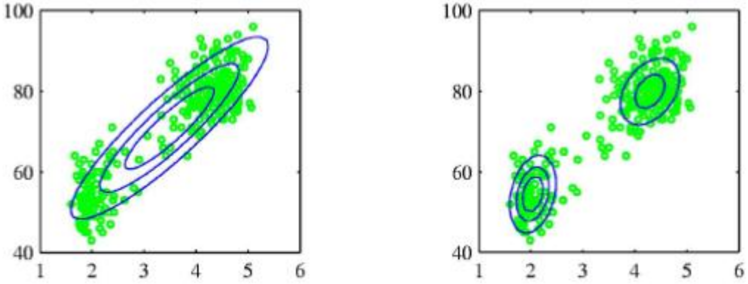
Uporaba GMM

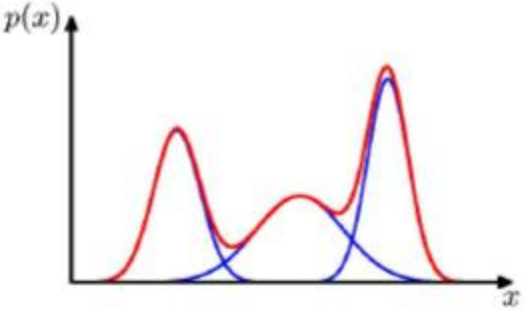
Grozdenje

- Old Faithful (hidrotermalni gejzir v parku Yellowstone)
  - 272 opazovanj
  - Trajanje (min, horiz os) *proti* času do naslednjega izbruha (vertikalna os)
  - Gaussov model ne more zajeti strukture
  - Linearna superpozicija dveh Gauss. se izkaže bolje
- Gaussov model ima omejitve pri modeliranju podatkov
- Gaussovi mešani modeli dajejo zelo kompleksne gostote

$$p(\mathrm{x}) = \sum_{k=1}^{K} \pi_k N(x \mid \mu_k, \Sigma_k)$$

- p_k so mešani koeficienti





Ocena Gaussovih mešanih modelov

- Log funkcija verjetja

$$\ln p(X \mid \pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N\left(\mathrm{x}_n \mid \mu_k, \Sigma_k\right) \right\}$$

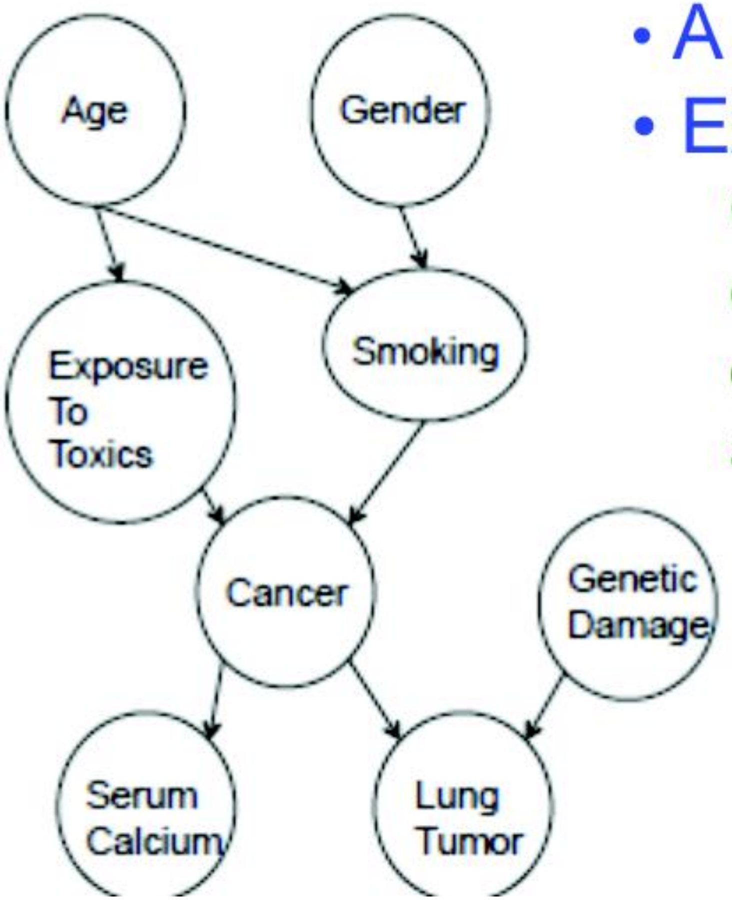
- Ni zaprte oblike rešitve
- Uporabi se iterativna numerična optimizacija ali maksimizacija pričakovanja

Bayesova predstavitev negotovosti

- Ne samo frekvenca naključja, tudi ponavljajoči se dogodki
- Verjetnost je kvantifikacija negotovosti
- Ali bo arktični ledeni pokrov izginil do konca stoletja
  - Imamo nekaj podatkov o hitrosti taljenja polarnega ledu
  - Preučimo jih ponovno na podlagi novih dokazov, podatkov (satelitske slike)
  - Ocena bo vplivala na prihodnja dejanja (zmanjševanje toplogrednih plinov)
- Vse to lahko dosežemo s splošno Bayesovo interpretacijo

Predstavitev negotovosti z verjetnostjo ni ”ad-hoc” izbira
Če za predstavitev stopnje verjetja uporabimo numerične vrednosti, potem preprosti aksiomi za spreminjanje stopnje verjetja vodijo k pravilom vsote in produkta za verjetnost

Modeliranje negotovosti



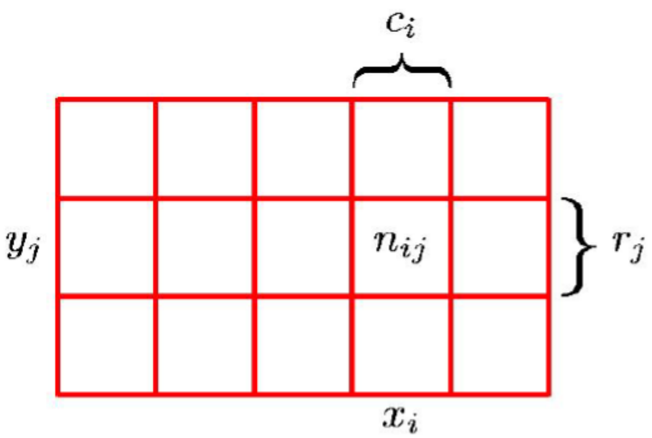
- A Causal Bayesian Network
- Example of Inference:

  Cancer is independent
  of Age and Gender given
  exposure to Toxics
  and Smoking

Popoln Bayesov pristop

- Bayesovo pravilo
- Bayesove verjetnosti
- koncept konjugacije
- Monte Carlo vzorčenje

Pravila verjetnosti



- Imamo naključni spremenljivki X in Y
- Pravilo vsote daje marginalno verjetnost

$$p(X = x_i) = \sum_{j=1}^{L} p(X = x_i, Y = y_j) = \frac{c_i}{N}$$

- Pravilo produkta: skupna verjetnost glede na pogojno in marginalno

$$p(X,Y) = \frac{n_{ij}}{N} = p(Y \mid X)\,p(X) = \frac{n_{ij}}{c_i} \times \frac{c_i}{N}$$

- S kombiniranjem dobimo Bayesov teroem

$$p(Y \mid X) = \frac{p(X \mid Y)\,p(Y)}{p(X)}$$

kjer je

$$p(X) = \sum_{Y} p(X \mid Y)\,p(Y)$$

kot Posteriorna α verjetnost x pred

Verjetnostne porazdelitve: povezave

Discrete-Binary

Binomial
N samples of Bernoulli

N=1

Bernoulli
Single binary variable

Conjugate Prior

Beta
Continuous variable between {0,1]

K=2

Discrete-Multi-valued

Large N

Multinomial
One of K values = K-dimensional binary vector

Conjugate Prior

Dirichlet
K random variables between [0.1]

Continuous

Gaussian

Student's-t
Generalization of Gaussian robust to Outliers
Infinite mixture of Gaussians

Gamma
ConjugatePrior of univariate Gaussian precision

Wishart
Conjugate Prior of multivariate Gaussian precision matrix

Exponential
Special case of Gamma

Gaussian-Gamma
Conjugate prior of univariate Gaussian Unknown mean and precision

Gaussian-Wishart
Conjugate prior of multi-variate Gaussian Unknown mean and precision matrix

Angular

Von Mises

Uniform

Popoln Bayesov pristop

- Izvedljiv s povečano računsko močjo
- A posteriori porazdelitev je lahko obravnavana z
  - variacijskim Bayesom ali
  - stohastičnim (naključnim) vzorčenjem
    - npr., Markov Chain Monte Carlo, Gibbs

Povzetek

- ML je sistematičen pristop k sestavljanju sistemov za obdelavo informacij
- Upoabno za reševanje problemov klasifikacije, regresije, grozdenja in sklepanja
- popoln Bayesov pristop skupaj z Monte Carlo vzorčenjem omogoča visoko zmogljive sisteme
- velika praktična vrednost v več aplikacijah
  - Računalništvo
    - iskalniki
    - obdelava besedil
    - prepoznava dokumentov
  - Elektrotehnika

- Literatura:
  - Sargur N. Srihari, Lecture notes, University at Buffalo, State University of New York