



Teorie kognitivních systémů

3 Lineární regrese – II

- Lineární regrese více proměnných
- Gradientní sestup v nD
- Algoritmus GS v nD
- Problémy a omezení GS
- Aplikace a nastavení
- Polynomiální regrese
- Normální rovnice





Lineární regrese s více proměnnými

Definice problému

Přv. příklad predikce velikosti obuvi...

	1	2
1	161	36
2	165	38
3	170	39
4	170	41
5	173	41
6	175	42
7	180	44.5000
8	185	43
9	190	45
10	195	46

X
výška
[cm] **y**
vel. bot
[čís.]

$$h_{\Theta}(x) = \Theta_0 + \Theta_1 x$$

Nová verze příkladu predikce velikosti obuvi s více daty...

	1	2	3	4
1	161	24	47	36
2	165	21	50	38
3	170	31	55	39
4	170	19	51	41
5	173	35	73	41
6	175	27	86	42
7	180	18	90	44.5000
8	185	29	79	43
9	190	43	82	45
10	195	26	103	46

X₁
výška
[cm] **X₂**
věk
[rok] **X₃**
hmot.
[kg] **y**
vel. bot
[čís.]

$$h_{\Theta}(x) = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + \Theta_3 x_3$$



Lineární regrese s více proměnnými

Definice problému

Značení:

	1	2	3	4
1	161	24	47	36
2	165	21	50	38
3	170	31	55	39
4	170	19	51	41
5	173	35	73	41
6	175	27	86	42
7	180	18	90	44.5000
8	185	29	79	43
9	190	43	82	45
10	195	26	103	46

m – celkový počet vzorků v trénovací mn.

X_1 X_2 X_3 y
 výška [cm] věk [rok] hmot. [kg] vel. bot [čís.]

$$X_2^{(5)} = 35$$

n – počet příznaků (tj. počet složek vstup. vektoru)



Lineární regrese s více proměnnými

Definice problému

Lineární regrese s více proměnnými (*Multivariate Linear Regression*) – hypotéza bude mít tvar:

$$h_{\Theta}(x) = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + \dots + \Theta_n x_n$$

$$h_{\Theta}(x) = \Theta_0 x_0 + \Theta_1 x_1 + \Theta_2 x_2 + \dots + \Theta_n x_n, x_0^{(i)} = 1$$

$$h_{\Theta}(x) = \Theta^T x$$

$$\Theta = \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \vdots \\ \Theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$$



Gradientní sestup v nD „terénu“

Formulace úlohy a popis algoritmu

Hypotéza: $h_{\Theta}(x) = \Theta^T x = \Theta_0 x_0 + \Theta_1 x_1 + \Theta_2 x_2 + \dots + \Theta_n x_n$

Parametry: $\Theta = [\Theta_0, \Theta_1, \dots, \Theta_n]^T$

Cenová funkce: $J(\Theta) = \frac{1}{2m} \sum_{i=1}^m \left(h_{\Theta}(x^{(i)}) - y^{(i)} \right)^2$

Gradientní sestup:

```
while not converged() do {
```

```
    for j = 1 .. n do  $\Theta_j \leftarrow \Theta_j - \alpha \frac{\partial J(\Theta)}{\partial \Theta_j}$ 
```

```
}
```

„současná“ úprava
hodnot parametrů





Gradientní sestup v nD „terénu“

Parciální derivace cenové funkce

$$\begin{aligned}
 \frac{\partial J(\Theta)}{\partial \Theta_j} &= \frac{\partial}{\partial \Theta_j} \frac{1}{2m} \sum_{i=1}^m \left(h_\Theta(x^{(i)}) - y^{(i)} \right)^2 \\
 &= \frac{\partial}{\partial \Theta_j} \frac{1}{2m} \sum_{i=1}^m \left(\Theta_0 x_0^{(i)} + \Theta_1 x_1^{(i)} + \dots + \Theta_n x_n^{(i)} - y^{(i)} \right)^2 \\
 &= \frac{1}{m} \sum_{i=1}^m \left(\Theta_0 x_0^{(i)} + \Theta_1 x_1^{(i)} + \dots + \Theta_n x_n^{(i)} - y^{(i)} \right) \cdot x_j^{(i)}
 \end{aligned}$$

POZOR: Jedná se o derivaci výrazu ve tvaru $(f(x))^2$, tj. derivaci mocniny a vnitřní funkce...





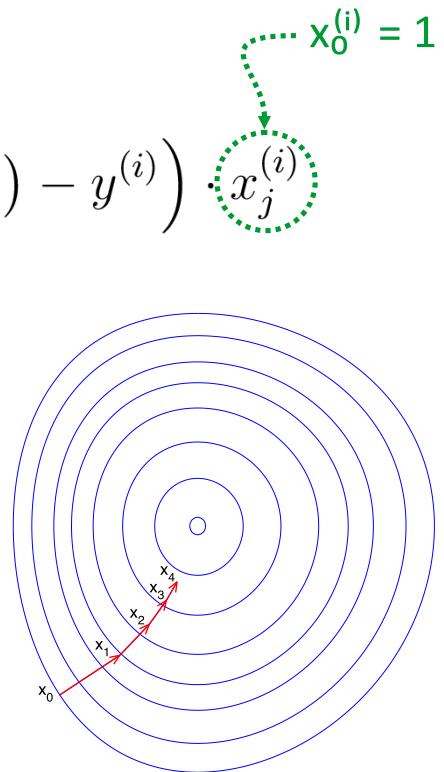
Gradientní sestup v nD „terénu“

Algoritmus v pseudokódu (s derivací)

```

while not converged() do {
    for j = 1 .. n do
         $\Theta_j \leftarrow \Theta_j - \alpha \frac{1}{m} \sum_{i=1}^m \left( h_\Theta(\mathbf{x}^{(i)}) - y^{(i)} \right) \cdot x_j^{(i)}$ 
}
  
```

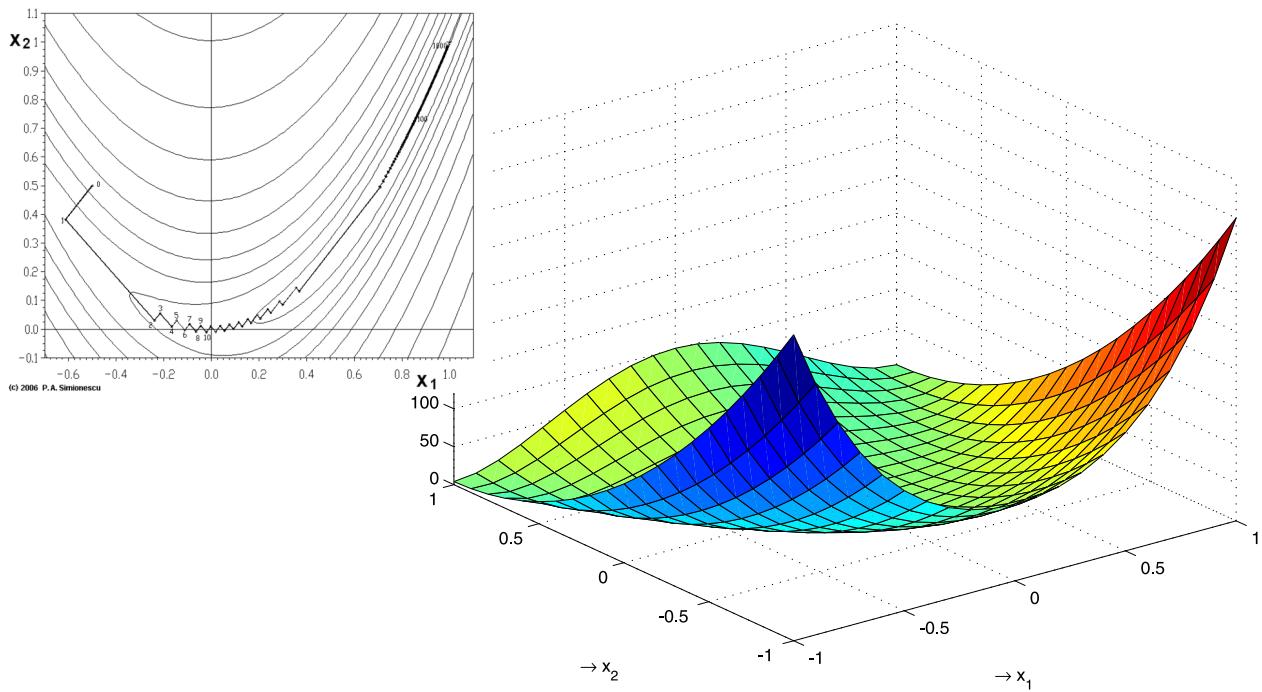
Gradientní sestup je tzv. **optimalizační algoritmus 1. řádu (First-order Optimization Algorithm)** – používá 1. parciální derivaci cenové funkce.





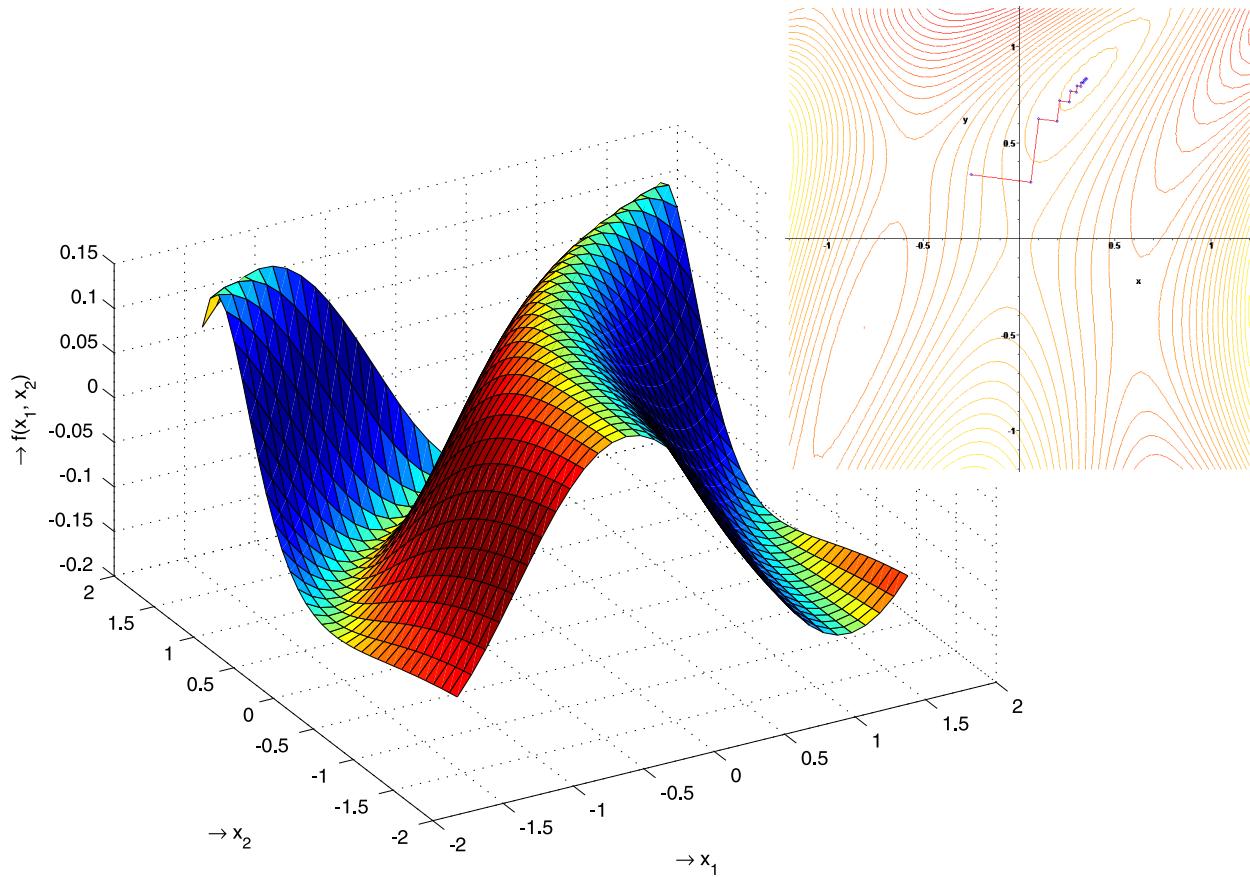
Problémy a omezení gradientního sestupu „Patologické“ funkce

Gradientní sestup má problémy s oscilací a pomalou konvergencí v případech f-cí podobných tzv. **Rosenbrockově funkci**
 $f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$





Problémy a omezení gradientního sestupu „Patologické“ funkce





Problémy a omezení gradientního sestupu vyplývající z matem. vlastností metody

- často velmi pomalý v blízkosti optima – obecně je asymptotická míra konvergence GS **infimum** AMK řady jiných (pokročilejších) metod
- v případě špatně podmíněných konvexních úloh (předchozí obrázky) „kličkuje“ se zvyšující se amplitudou, neboť gradient směruje téměř kolmo k nejkratší vzdálenosti k optimu
- není-li cenová funkce derivovatelná, jsou gradientní metody obecně špatně podmíněné (lze řešit např. vyhlazením cenové funkce pomocí approximace splinem, apod. nebo aplikací nějaké negradientní metody)
- nalezení lokálního optima s danou přesností může vyžadovat mnoho iterací tehdy, je-li zakřivení cenové funkce v jednotlivých směrech (daných parciálními derivacemi) vzájemně rádově různé



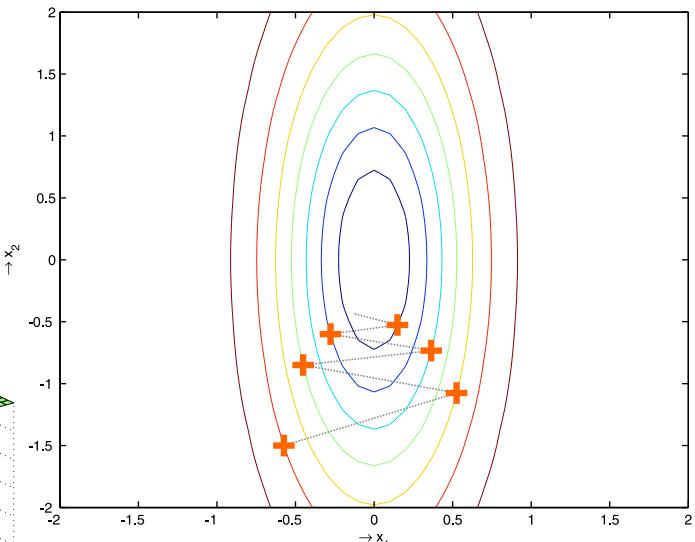
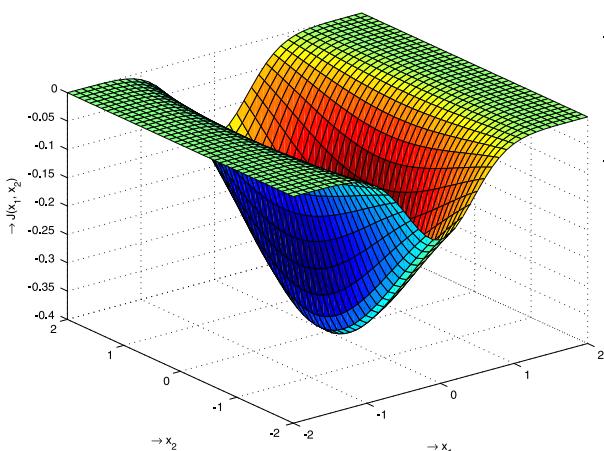


Aplikace gradientního sestupu

Škálování příznaků (*Feature Scaling*)

Klíčová myšlenka: Zajistit, aby měly jednotlivé složky vstupního (příznakového) vektoru zhruba stejný rozsah hodnot...

Z příkladu s obuví:
 x_1 (výška) $\in <120, 200>$
 x_2 (věk) $\in <10, 80>$



Hrozí riziko sestupu „cik-cak“ a pomalé (nebo žádné) konvergence...



Aplikace gradientního sestupu

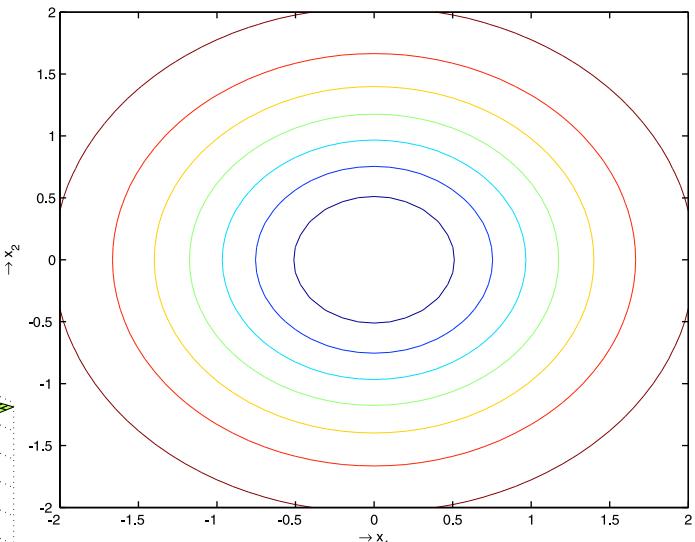
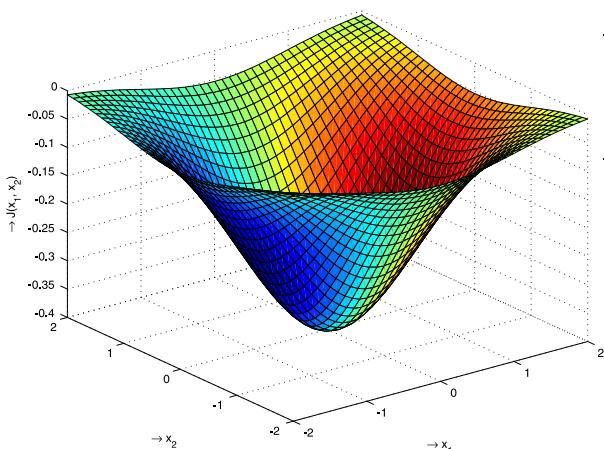
Škálování příznaků (*Feature Scaling*)

Vydělíme tedy každou složku příznakového vektoru maximem dosaženým v abs. hodnotě v této složce:

$$x_j^{(i)} = x_j^{(i)} / \max_i |x_j^{(i)}|$$

načež platí (**žádoucí**):

$$\forall_{i,j} : x_j^{(i)} \in \langle -1, 1 \rangle$$





Aplikace gradientního sestupu

Normalizace střední hodnoty

Klíčová myšlenka: Zajistit, aby měly jednotlivé složky vstupního (příznakového) vektoru zhruba nulovou střední hodnotu...

$$x_j^{(i)} = x_j^{(i)} - \mu_j^{(i)} / (\max_i x_j^{(i)} - \min_i x_j^{(i)})$$

průměrná hodnota
v „j-tém sloupci“ dat

rozptyl (Range | StD), tj.
rozdíl maxima a minima
v „j-tém sloupci“ dat

POZOR: Neaplikovat na $x_0 = 1$. Složka x_0 musí všude v datech zůstat jednotková, jinak optimalizace neproběhne korektně...





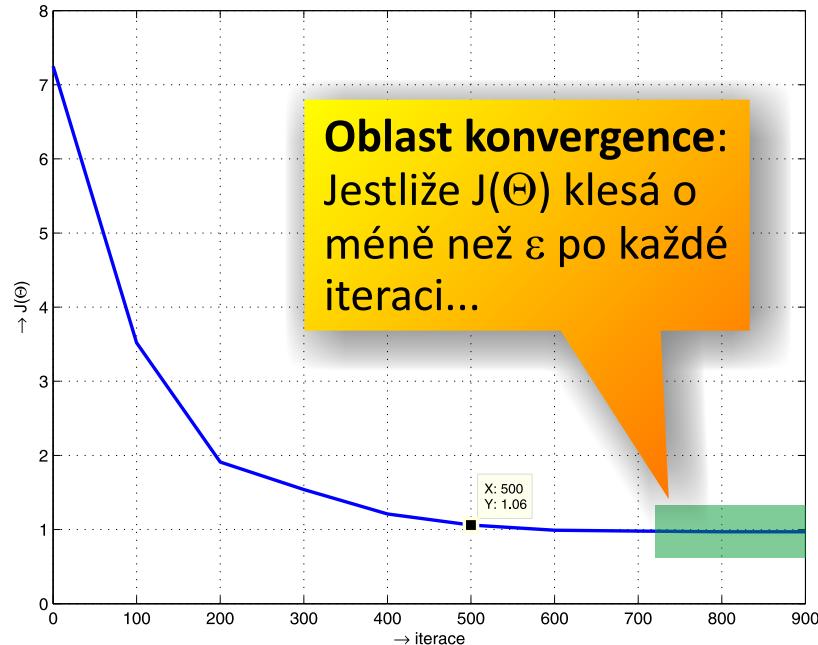
Aplikace gradientního sestupu

Debugging – jak zjistit, že GS pracuje správně?

GS je **minimalizace cenové funkce**, tj. po každé iteraci musí hodnota $J(\Theta)$ klesnout (pracuje-li GS správně):

Podmínka zastavení GS: Pokud se hodnota $J(\Theta)$ **zvýší oproti předchozí iteraci**.

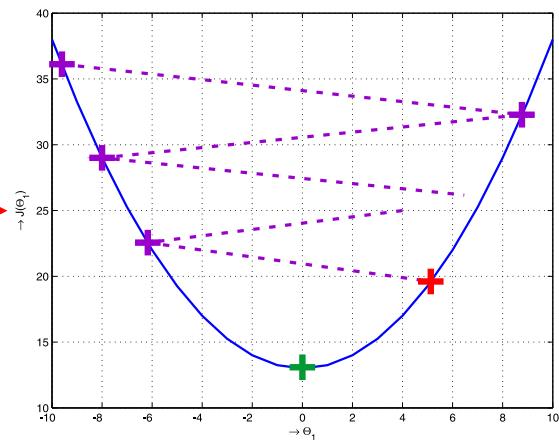
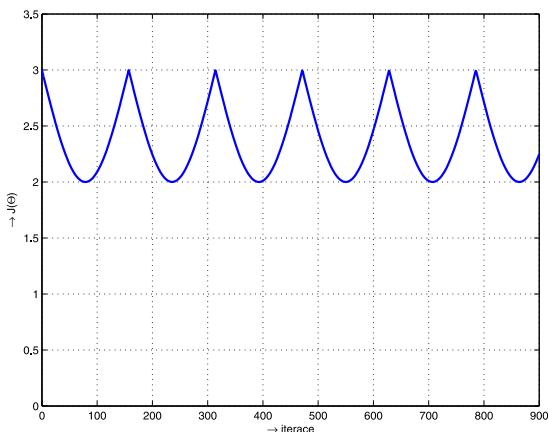
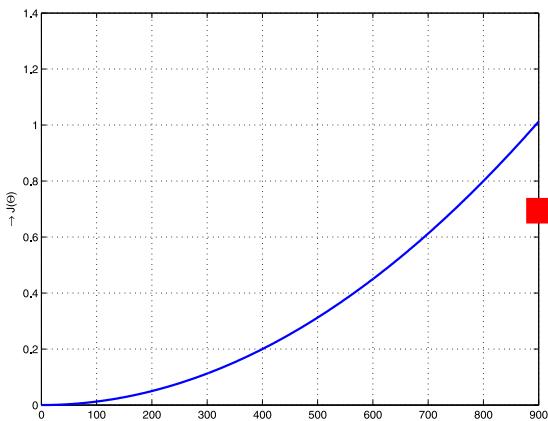
Podmínka konvergence: $\varepsilon = 10^{-3}$ (ideálně zvolit dle průběhu $J(\Theta)$...)





Aplikace gradientního sestupu

Volba míry učení α



GS nekonverguje: je třeba použít menší hodnotu α ...

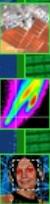
- pro dostatečně malé α klesá hodnota $J(\Theta)$ v každé iteraci
- je-li ale α příliš malé, GS je pomalý



Aplikace gradientního sestupu

Volba míry učení α – shrnutí

- k přesnému „vyladění“ α je **prakticky nezbytné** nechat si vykreslit průběh minimalizace $J(\Theta)$ a podle tvaru křivky volit α
- hodnotu α je vhodné zvyšovat po řádech, např.:
... → 0.001 → 0.01 → 0.1 → 1.0 → ...
- pro různé optimalizační úlohy se může hodnota konstanty α dost výrazně lišit – neexistuje „univerzální“ nastavení
- u „patologických“ funkcí raději volíme pomalou konvergenci prostřednictvím nízké hodnoty α , než nepředvídatelné chování při přestřelování optima





Polynomiální regrese

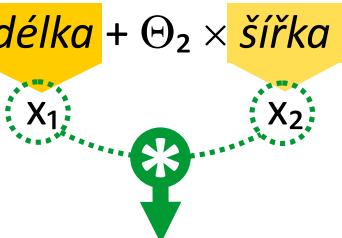
Motivační příklad

Odhad ceny pozemku podle jeho rozměrů:

Hypotéza má „naivní“ tvar $h_{\Theta}(x) = \Theta_0 + \Theta_1 \times \text{délka} + \Theta_2 \times \text{šířka}$



(snímek z www.mapy.cz)



plocha pozemku
– ta asi ovlivňuje
cenu pozemku víc,
než délka a šířka (?)

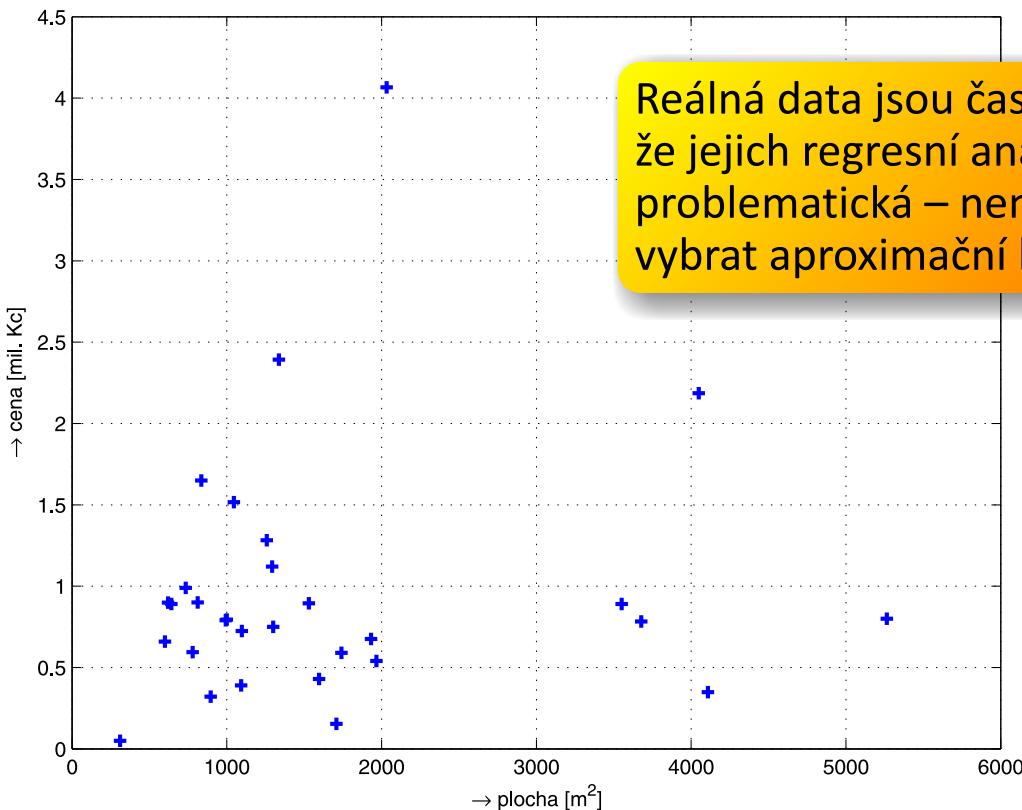
$$h_{\Theta}(x) = \Theta_0 + \Theta_1 \times \text{plocha}$$

Příznaky lze vytvářet
z hodnot pozorovaných
veličin...



Polynomiální regrese

Motivační příklad – ceny pozemků

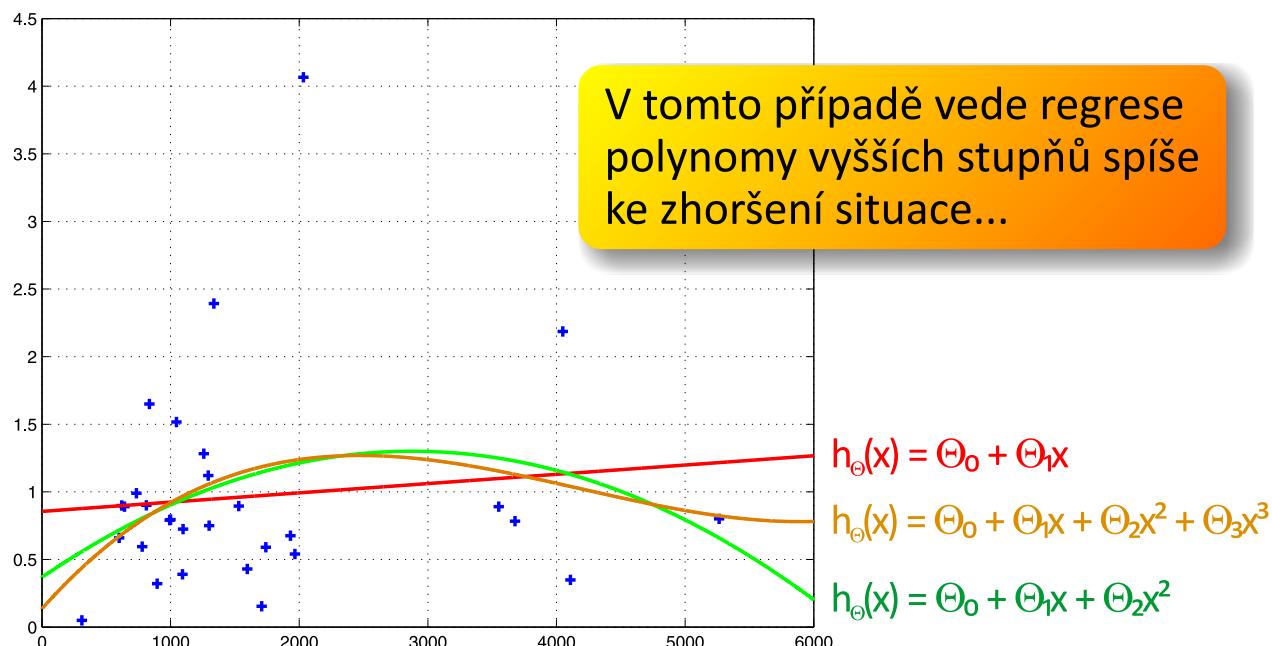


Reálná data jsou často taková, že jejich regresní analýza je problematická – není snadné vybrat aproximační křivku...



Polynomiální regrese

Regresy ceny polynomy různých stupňů



Pro výpočet používáme stále stejný aparát – substituujeme:

$$h_{\Theta}(x) = \Theta_0 + \Theta_1 x + \Theta_2 x^2 + \Theta_3 x^3 \rightarrow h_{\Theta}(x) = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + \Theta_3 x_3$$



Polynomiální regrese

Škálování u polynomů vyšších stupňů

Použijeme-li při výpočtu polynomiální regrese substituci:

$$x_1 = x = (\text{plocha})$$

$$x_2 = x^2 = (\text{plocha})^2$$

$$x_3 = x^3 = (\text{plocha})^3$$

nic jiného nám nezbývá,
chceme-li využít aparát
lineární regrese...

pak je **nutné** upravit pomocí škálování rozsahy hodnot, jelikož

- plocha $\in (0, 6000)$, pak nutně
- $(\text{plocha})^2 \in (0, 36000000)$, a dále
- $(\text{plocha})^3 \in (0, 2.16 \times 10^{11})$

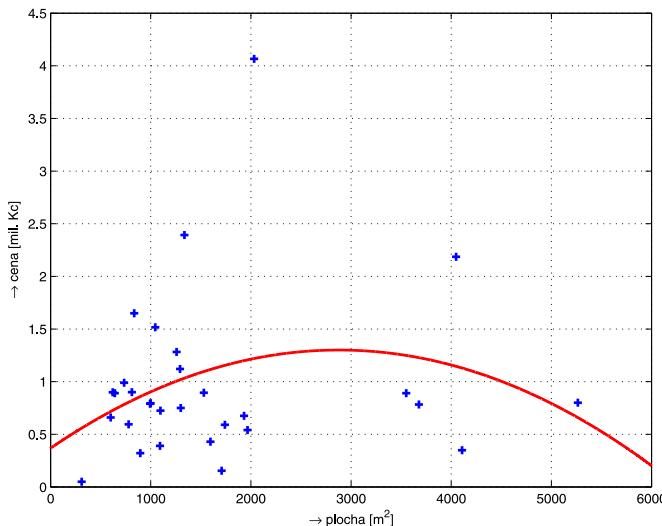
➔ nenaškálované hodnoty (tj. hodnoty s významně nestejným rozsahem) by drasticky ovlivnily výkon GS – docházelo by ke kličkování, přestřelování, atd.



Polynomiální regrese

Výběr příznaků

V příkladu s cenou pozemků je regrese polynomem 2. stupně, tj. kvadratickou funkcí, **zcestná** – ceny pozemků jistě nezačnou od jisté plochy (zde 2878 m^2) klesat...



Můžeme volit **jiné funkce**, jejichž tvary nám vyhovují – stačí upravit tvar hypotézy použité pro výpočet GS, např.:

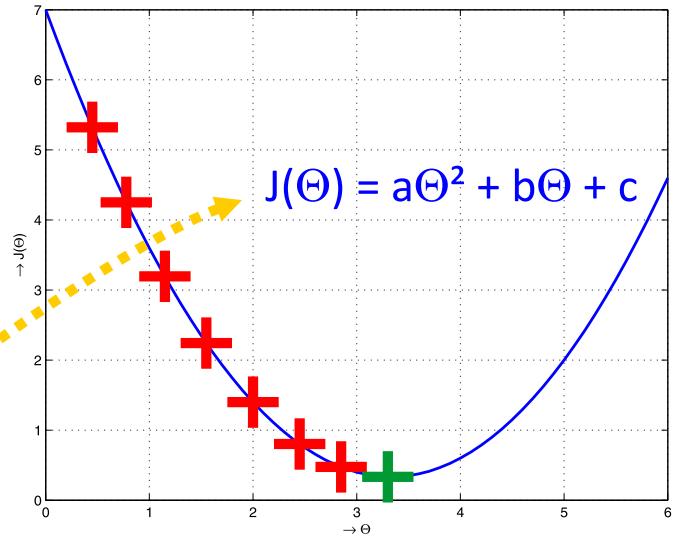
$$h_{\Theta}(x) = \Theta_0 + \Theta_1 x + \Theta_2 \sqrt{x}$$



Normální rovnice

Analytický aparát lineární regrese

Gradientní sestup –
iterační algoritmus
minimalizace $J(\Theta)$



Normální rovnice (Normal Equation) –
analytický postup minimalizace cenové funkce $J(\Theta)$;
známe-li její tvar a umíme-li derivovat, pak lze pozici minima
zjistit výpočtem...



Normální rovnice

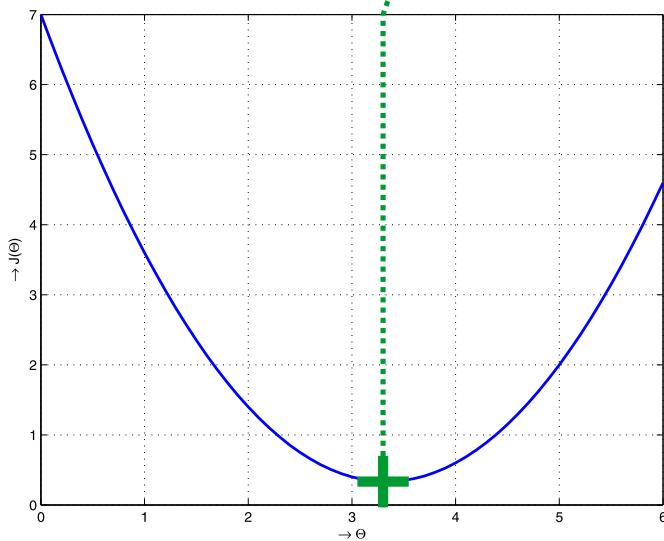
Výpočet v 1D

$$J(\Theta) = a\Theta^2 + b\Theta + c, \Theta \in \mathbb{R}$$

Jak zjistit pozici minima? **Derivací** podle Θ :

$$\frac{dJ(\Theta)}{d\Theta} = \frac{d}{d\Theta} a\Theta^2 + b\Theta + c = \boxed{2a\Theta + b = 0} \rightarrow \Theta = -\frac{b}{2a}$$

Položíme-li 1. derivaci
cenové funkce rovnu 0,
najdeme pozici optima.





Normální rovnice

Výpočet v nD

Cenová funkce závisí na n parametrech Θ_j :

$$\Theta \in \mathbb{R}^{n+1} : J(\Theta) = J(\Theta_0, \Theta_1, \dots, \Theta_n) = \frac{1}{2m} \sum_{i=1}^m \left(h_\Theta(x^{(i)}) - y^{(i)} \right)^2$$

Je třeba vyřešit pro každé Θ_j :

$\frac{\partial J(\Theta)}{\partial \Theta_j} = \dots = 0$, tj. 1. parciální derivace položené rovno 0,
abychom dostali polohu optima vzhledem
ke všem souřadnicím...

PROBLÉM: Je-li n opravdu velké, je výpočet normální rovnice nezvládnutelný (*Unfeasible*), protože matice přeurčené soustavy má velikost $m \times (n + 1)$ a během výpočtu se užívá v součinu se svou transpozicí, tj. $n \times n \dots$ viz dále





Normální rovnice

Výpočet v n D pomocí přeurečené soustavy

Vezměme motivační příklad s odhadem velikosti obuvi:

Data je nutno doplnit o vektor

x_0 , jehož složky jsou jednotky

→ výpočet aparátem LA

$$X = \begin{bmatrix} 1 & 161 & 24 & 47 \\ 1 & 165 & 21 & 50 \\ 1 & 170 & 31 & 55 \\ 1 & 170 & 19 & 51 \\ 1 & 173 & 35 & 73 \\ 1 & 175 & 27 & 86 \\ 1 & 180 & 18 & 90 \\ 1 & 185 & 29 & 79 \\ 1 & 190 & 43 & 82 \\ 1 & 195 & 26 & 103 \end{bmatrix} \quad y = \begin{bmatrix} 36 \\ 38 \\ 39 \\ 41 \\ 41 \\ 42 \\ 44.5 \\ 43 \\ 45 \\ 46 \end{bmatrix}$$

	1	2	3	4
1	1	161	24	47
2	1	165	21	50
3	1	170	31	55
4	1	170	19	51
5	1	173	35	73
6	1	175	27	86
7	1	180	18	90
8	1	185	29	79
9	1	190	43	82
10	1	195	26	103

x_1 výška [cm] x_2 věk [rok] x_3 hmot. [kg] y vel. bot [čís.]

výpočet koeficientů Θ

$$\Theta = (X^T X)^{-1} X^T y$$



Normální rovnice

Výpočet v n D pomocí přeurečené soustavy

TM

příznaky pozorovaných vzorků

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}, i \in \langle 1, m \rangle$$

odpovědi učitele

$$\mathbf{y} = \begin{bmatrix} y^{(i)} \\ y^{(i)} \\ y^{(i)} \\ \vdots \\ y^{(i)} \end{bmatrix} \in \mathbb{R}, i \in \langle 1, m \rangle$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)} \end{bmatrix}$$

$\Theta = (X^T X)^{-1} X^T y$

výpočet koeficientů Θ



Normální rovnice

Výpočet pomocí MATLABu/Octave

$$\Theta = (X^T X)^{-1} X^T y$$

```
>> pinv(X' * X) * X' * y'
```

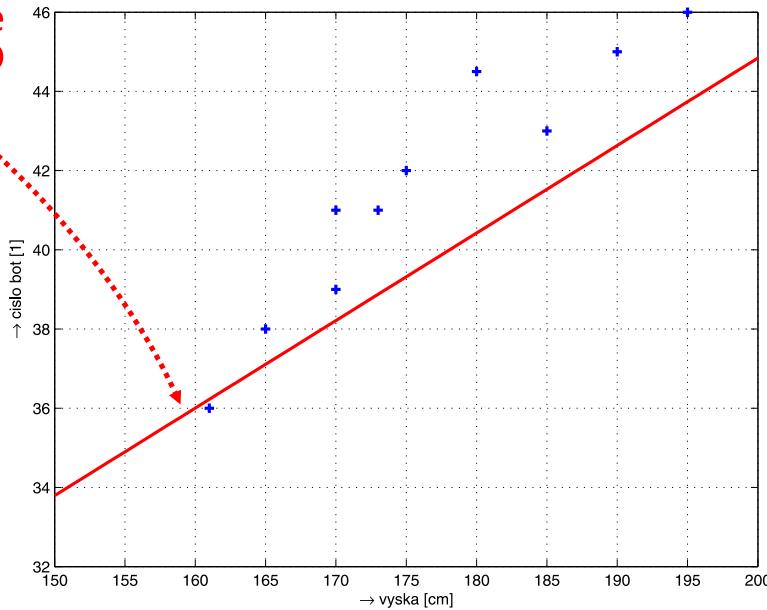
```
ans =
```

0.6430
0.2210
-0.0449
0.0439

```
>> |
```

Zdá se, že aproxi-
mace přímkou
není příliš přesná
– ale jedná se o
projekci

projekce
do 2D





Gradientní sestup vs normální rovnice

Kdy zvolit který způsob?

Máme-li m vzorků v trénovací množině, každý o n složkách:

Gradientní sestup

- je potřeba zvolit α
- musí proběhnout mnoho iterací
- funguje dobře i tehdy, je-li n **velmi velké** (10^6)

Normální rovnice

- není potřeba volit α
- neiteruje se
- je třeba násobit a invertovat velké matice ($O(n^3)$)
- velmi pomalý postup, je-li n **velmi velké** (10^6)

