

Analýza dat: lineární regrese, detekce shluků

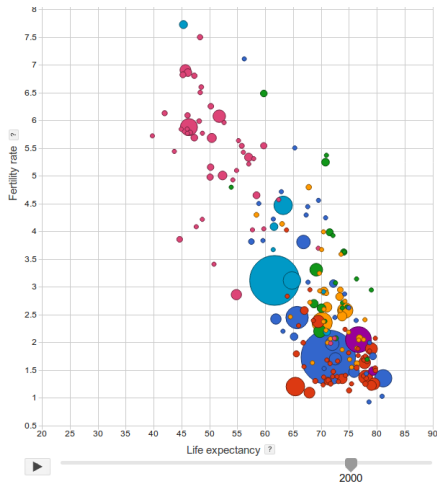
Radek Pelánek

IV122, jaro 2013

Úvodní poznámky

- rozbor dvou konkrétních technik
 - lineární regrese
 - detekce shluků (k -means)
- průběžně ilustrace obecných principů z analýzy dat, pravděpodobnosti, strojového učení, ...

Reálná data: délka života, porodnost

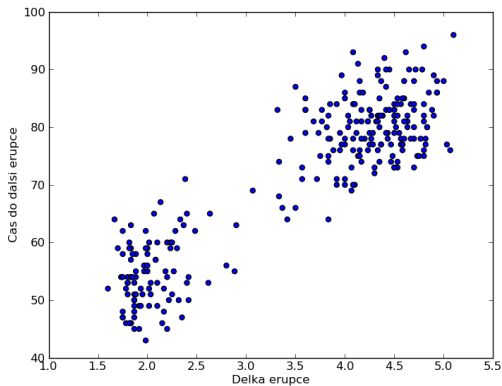


Google Public Data / World Bank

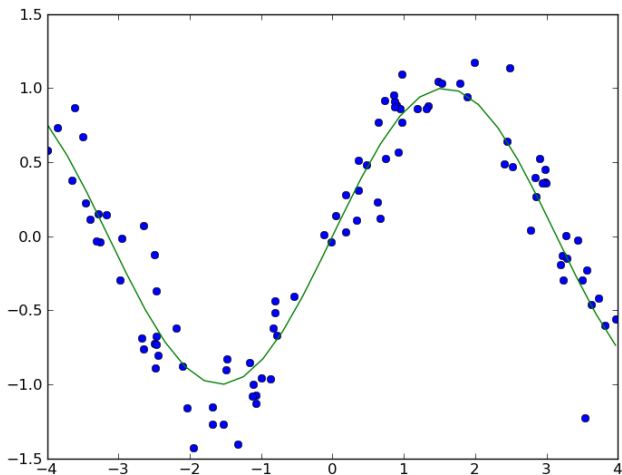
Reálná data: Old Faithful



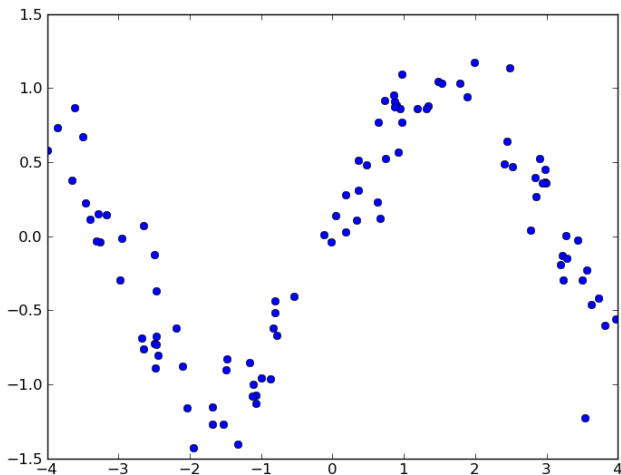
Zdroj: Wikipedia



Simulovaná data: generování



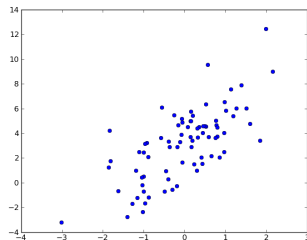
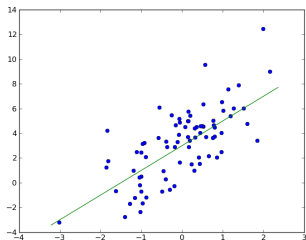
Simulovaná data: vstup pro analýzu



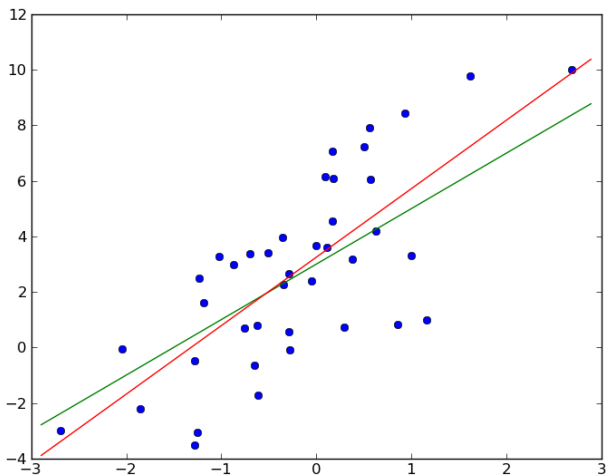
Simulovaná data

- (syntetická data)
- zvolíme „správné řešení“
- vygenerujeme data: „správné řešení“ + náhodný šum
- náhodný šum \sim normální distribuce
- algoritmu pro analýzu dat dáme pouze vygenerovaná data
- výsledek algoritmu můžeme porovnat se správným řešením
- dobrý přístup pro pochopení, ladění implementace, nastavení parametrů

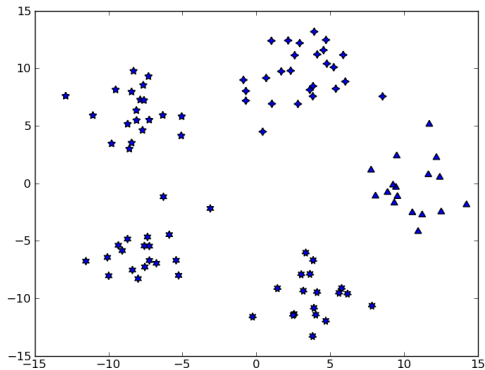
Simulovaná data: lineární regrese



Lineární regrese



Simulovaná data: Detekce shluků



- k dispozici data pro lineární regresi a detekci shluků
- zkuste najít „co nejlepší“ přímku / rozdělení na shluky
 - 1 co to znamená „co nejlepší“?
 - 2 jak hledat?
- zkuste vymyslet ...
žádný Google, Wikipedie, studijní materiály

Která přímka je nejlepší?

- hledáme co nejlepší přímku $ax + b$
- minimalizace „sumy čtverců chyb“ (sum of squared error)

$$SSE = \sum_{i=1}^n (y_i - (ax_i + b))^2$$

- proč zrovna tato funkce?

Která přímka je nejlepší?

- hledáme co nejlepší přímku $ax + b$
- minimalizace „sumy čtverců chyb“ (sum of squared error)

$$SSE = \sum_{i=1}^n (y_i - (ax_i + b))^2$$

- proč zrovna tato funkce?
- pragmaticky: dobře se s tím pracuje
- teoreticky: nejlepší vysvětlení dat při předpokladu normálního šumu

Metoda maximální věrohodnosti

- maximum likelihood estimation
- jaká je věrohodnost (likelihood) dat, pokud jsou generována přímkou $ax + b$?

$$L = \prod_i p(x_i, y_i) = \prod \mathcal{N}(ax_i + b, \sigma^2)(y_i)$$

- hledáme a, b tak, abychom maximalizovali
- vezmeme logaritmus (monotónní operace, zachovává maximum)
- maximalizovat L je to stejné jako minimalizovat sumu čtverců:

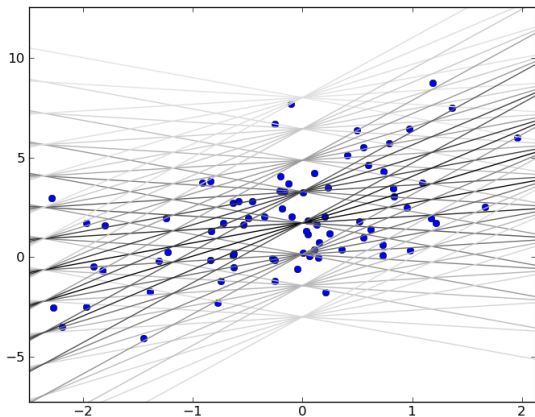
$$SSE = \sum_{i=1}^n (y_i - (ax_i + b))^2$$

Jak najít přímku minimalizující SSE?

- analytické řešení „vzorečkem“ – ideální řešení, tady funguje, u složitějších problémů však nikoliv
- pro ilustraci:
 - „grid search“ – hrubá síla
 - gradient descent – postupné vylepšování

Grid search

8 hodnot b , 7 hodnot a ; stupeň šedi \sim SSE



Analytické řešení

$$SSE = \sum_{i=1}^n (y_i - (ax_i + b))^2$$

- hledáme minimum vzhledem k a , b
- parciální derivace musí být 0

$$\frac{SSE}{\partial a} = 2 \sum_{i=1}^n -y_i x_i + ax_i^2 + x_i b = 0$$

$$\frac{SSE}{\partial b} = 2 \sum_{i=1}^n -y_i + ax_i + b = 0$$

Analytické řešení

Po algebraických úpravách dostaneme:

$$a = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} = r_{xy} \frac{s_y}{s_x}$$

$$b = \bar{y} - a\bar{x}$$

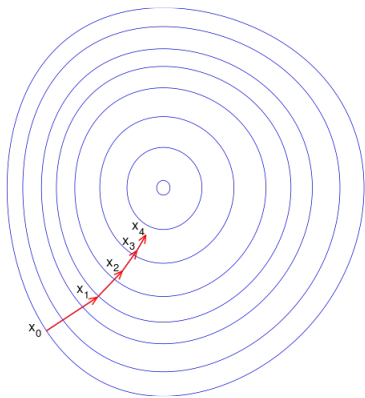
r_{xy} – korelační koeficient

s_x, s_y – standardní odchylka x, y

Metoda největšího spádu

- gradient descent
- „hladová“ metoda
 - začneme s iniciálním odhadem parametrů
 - iterativně zlepšujeme
- snažíme se o co největší lokální zlepšení = úprava hodnot parametrů ve směru spádu (gradient)
- parametr „learning rate“: velikost skoku ve směru gradientu
 - příliš malý – pomalé
 - příliš velký – nestabilní (nekonverguje)

Gradient descent: intuitive



Wikipedia

Gradient descent pro lineární regresi

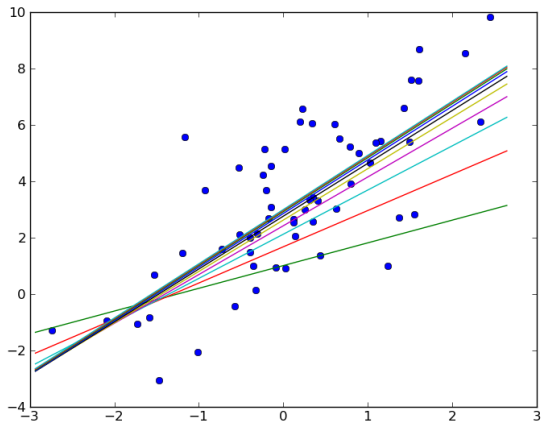
$$SSE = \frac{1}{2} \sum_{i=1}^n (y_i - (ax_i + b))^2$$

gradient:

$$\frac{SSE}{\partial a} = - \sum_{i=1}^n x_i (y_i - (ax_i + b))$$

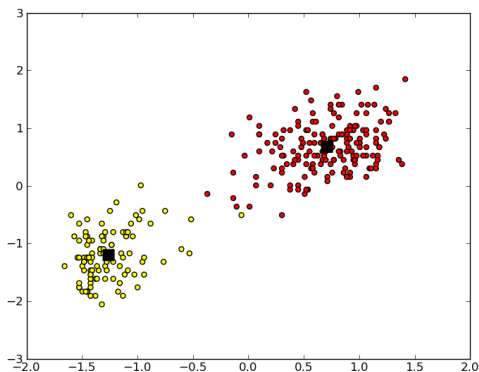
$$\frac{SSE}{\partial b} = - \sum_{i=1}^n (y_i - (ax_i + b))$$

Gradient descent demo



Detekce shluků (clustering)

Old Faithful



Cíl shlukování

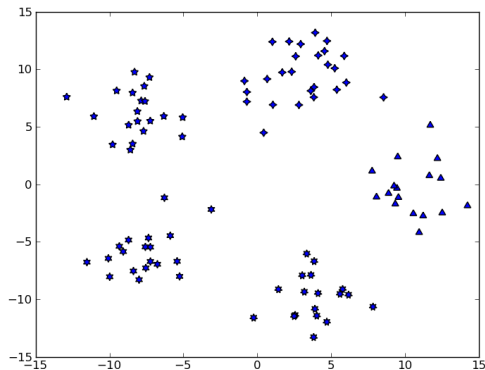
shlukování obecně:

- minimalizovat vzdálenosti v rámci shluku
- maximalizovat vzdálenosti mezi shluky

konkrétně např: minimalizace sum čtverců vzdáleností od středů shluků

pozn. normalizace (standardizace) dat (z-skóre), viz např. Old Faithful data

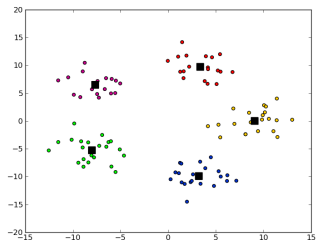
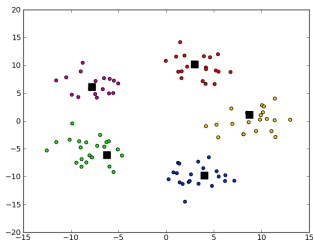
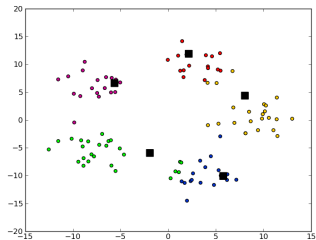
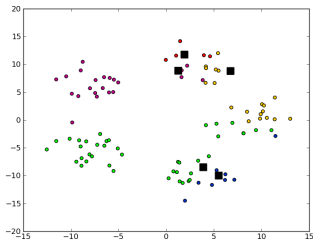
Detekce shluků – simulovaná data



Algoritmus k -means

- vyber k „středů shluků“
- opakuj:
 - každý bod přiřaď do toho shluku, jehož střed je nejbližší
 - aktualizuj polohu středů – těžiště bodů přiřazených do shluku

Algoritmus k -means: ukázka



Algoritmus k -means – poznámky

- hladová metoda
- lokální optima
- role inicializace
- opakované spuštění