Geostatistika Lekce 2

Table of Contents

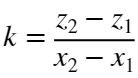
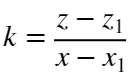
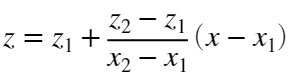
[Geostatistika Lekce 2](#MW_T_D63DC60A)   
 [Otázky](#MW_H_921AF0B8)   
[Lineární interpolace](#MW_T_D75DD866)   
 [Lineární interpolace v jednom bodě](#MW_H_0B3D79FE)   
 [Lineární interpolace ve více bodech](#MW_H_BA873BE6)   
[Metoda IDW](#MW_T_5DAD6EF4)   
 [Odvození](#MW_H_DD0D9CCD)   
 [Vlastnosti](#MW_H_FE32CD0A)   
 [Vliv velikosti okolí, anizotropie a rozložení bodů](#MW_H_802BE17F)

# Otázky

* Ve vzorci v knize 2.4 a 2.5 (již před 2.4) používáte index  a . Pokud jsem to správně pochopil, procházejí tyto indexy přes sousední prvky. Je nutné je potom rozlišovat? Zkusil jsem si vzorce cvičně přepsat (snad bez mnoha chyb) a po svém upravit, abych nakonec došel ke tvarům, které uvádíte v knize, ale použil jsem jen jeden index.
* Pokud to vyhovuje, budu vkládat některé přepsané kódy přímo do tohoto dokumentu, abych nemusel přikládat matlabovský soubor kvůli omezení SIS.
* Je nějaká konvence pro odsazování a komentování MATLAB funkcí případně skriptů? Po troše hledání se mi zdá, že to není nikde zcela jasně popsané.

Lineární interpolace

Mějme naměřené hodnoty  a  v bodech  a . Hledáme neznámou hodnotu v bodě , který leží mezi oběma danými body. Pro odhad hodnoty můžeme použít lineární interpolaci, kdy předpokládáme, že hledaná hodnota leží na spojnici (přímce) mezi body  a .

**Odvodíme** lineární závislost, do které poté můžeme dosadit hodnotu  z podobnosti trojúhelníků, Pro směrnici přímky platí, že  a zároveň . Nakonec vyjádřením dostaneme , což není nic jiného než směrnicová rovnice přímky.

**Poznámka** Ve statistice bychom takovou lineární závislost mezi nezávislou proměnnou  (*predictor*) a závislou proměnnou  (*response*) zapsali jako , kde  představuje hodnotu pro  (*y intercept*) a  je první regresní koeficient viz lineární regrese.

## Lineární interpolace v jednom bodě

Programový kód pro lineární interpolaci v jednom zadaném bodě je uložen v souboru linearInterpolation1.m s následujícím obsahem.

function z0 = linearInterpolation1(x1, z1, x2, z2, x0)

% Najde pomocí lineární interpolace hodnotu z0 pro dané x0

% ze dvou daných bodů (x1, z1) a (x2, z1).

z0 = z1 + (z2 - z1) / (x2 - x1) \* (x0 - x1);

end

**Úkol 1**. Do dokumentu vložte právě vytvořený obrázek interpolace

**Úkol 2.** Projděte si kód a pokuste se porozumnět jednotlivým příkazům. Kde to pokládáte za vhodné, vložte vysvětlující kometáře (začínají znakem %).

**Příklad 1.** Ve dvou bodech na ose  o souřadnicích  a  byly naměřeny hodnoty  a . Pomocí lineární interpolace najděte hodnotu  v bodě .

x1 = 1.0; z1 = 10.0;

x2 = 3.0; z2 = 20.0;

x0 = 1.5; z0 = linearInterpolation1(x1, z1, x2, z2, x0);

fprintf('%f', z0);

12.500000

Hodnoty si můžeme vykreslit do grafu, kde hledaný bod je červeně.

axis([0, 5, 0, 25]);

axis padded;

plot(x1, z1, 'bo', x2, z2, 'bo', x0, z0, 'ro', 'MarkerSize', 8);

grid on;

## Lineární interpolace ve více bodech

Programový kód pro lineární interpolace v  bodech je uložen v souboru linearInterpolationN.m s následujícím obsahem.

function zs = linearInterpolationN(x1, z1, x2, z2, xs)

% x1, z1 - zadaná souřadnice prvního bodu

% x2, z2 - zadaná souřadnice druhého bodu

% xs - zadané souřadnice v intervalu (x1, x2)

% zs - spočtené hodnoty pro interval (x1, x2)

n = length(xs);

zs = zeros(n);

for i = 1:n

zs(i) = linearInterpolation1(x1, z1, x2, z2, xs(i));

end

end

Vykreslíme si postupné interpolování pro jednotlivé body v poli xs a zs.

xs = [0.5: 0.1 :3.5];

zs = linearInterpolationN(x1, z1, x2, z2, xs);

hold on; % Nastavení překreslování grafu.

plot(x1, z1,'bo');

plot(x2, z2,'bo');

for i = 1 : length(zs)

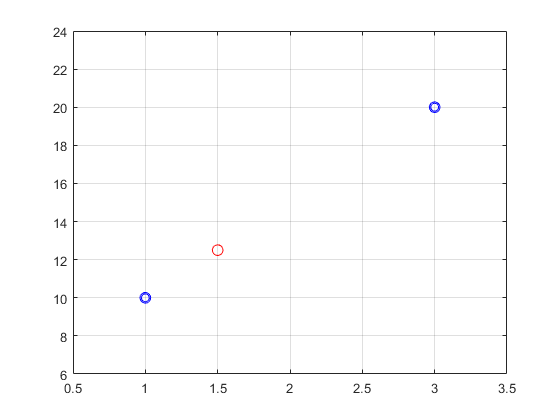
plot(xs(i), zs(i),'b--');

pause(0.1) % Nastavení zdržení animace v sekundách.

end

grid on;

hold off % Zruš nastavení, jinak platí pro další vyhodnocení buňky.



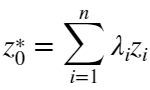
Metoda IDW

(učebnice GPI na str. 17 až 23)

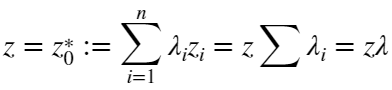
Metoda *Inverse Distance Weighting* (IDW) neboli *metoda inverzních vzdáleností* je založena na prostorová interpolační metoda založená na stanovení hodnoty hledaného bodu pomocí váženého průměru vzdálenosti okolních bodů.

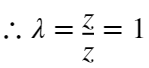
# Odvození

Mějme bod  jehož hodnotu  chceme odhadnout (hvezdička značí odhad) a v jeho okolí body  o známých hodnotách  (GPI obr. 2.1).



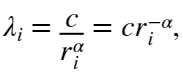
Pokud hodnota všech bodů v okolí stejná tj. , požadujeme, aby odhadnutá hodnota byla také stejná tj. .





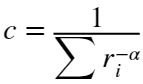
Požadavek aby součet vah  byl roven jedné budeme nazývat jako normovací podmínku.

Jednotlivé váhy pak položíme jako

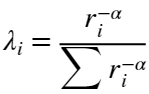


kde  je vzdálenost -tého bodu  od bodu , mocnina  a  je *škálovací koeficient,* odvodíme z normovací podmínky



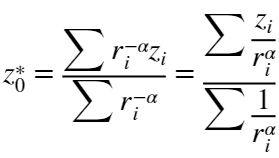


potom pro váhy dostaneme



(GPI 2.4)

a dosazením do získáme výdledný vzorec



(GPI 2.5)

Vetšinou první počítáme váhy, protože můžeme výpočet vah porovnávat s dosaženým výsledkem pro různě volené mocniny .

# Vlastnosti

* Charakteristickou vlastností IDW je tendence k průměru.
* Jde o přesnou interpolační metody tzn. že plocha prochází datovými body.
* Extrémní hodnoty tj. maxima a minima plochy jsou v datových bodech.
* Pokud se pohybujeme uvnitř intervalu lze lineární interpolaci chápat jako speciální případ IDW s hodnotou  (IDW1), to však již neplatí vně intervalu! Pro  shoda LI s IDW již neplatí, např. pro  (IDW10) již dostáváme výsledky podobné jako při metodě nebližšího souseda (*nearest neighbors method).*

**Příklad 2.** Ve dvou bodech na ose  o souřadnicích  = 1 a byly naměřeny hodnoty  a . Pomocí interpolace metodou IDW s koeficientem (mocninou)  najděte hodnotu veličiny  v bodě  = 1.5.

**Úkol 3.** Načrtněte situaci (tužka-papír) a zkuste nejprve přijít na řešení bez následujícího návodu .

**Řešení 3**. Zjistím sumu převrácených hodnot jednotlivých vdáleností  a spočtem jednotlivé normalizované váhy ,vyjde nám

x1 = 0.5;

z1 = 10.0;

x2 = 3.0;

z2 = 20.0;

x0 = 1.5;

r1 = abs(x0 - x1);

r2 = abs(x0 - x2);

R = (1/r1) + (1/r2);

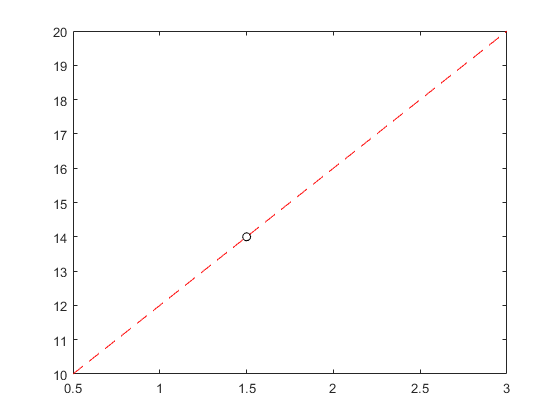
l1 = (1/r1)/R;

l2 = (1/r2)/R;

assert(l1 + l2 == 1.0); % Součet vah musí být jedna.

z0 = l1 \* z1 + l2 \* z2;

plot([x1, x2], [z1, z2], 'r--', x0, z0, 'blacko');

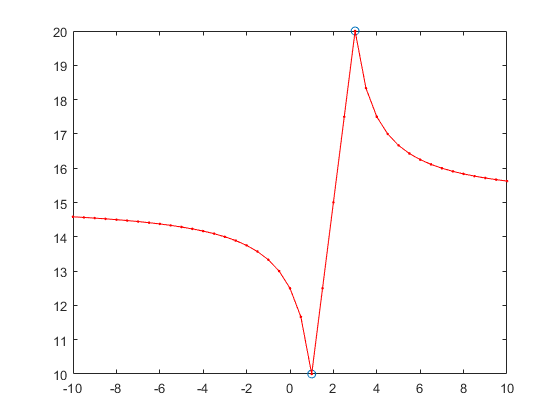


ans = 1×41

14.5833 14.5652 14.5455 14.5238 14.5000 14.4737 14.4444 ⋯

**Úkol 4.** Vložte do dokumentu výsledný obrázek z funkce `intpol\_IDW`a zkuste vysvětlit, proč se výsledky interpolace na krajích oblasti blíží průměru dat.

intpol\_IDW([1 3], [10 20], [-10:0.5:10], 1)

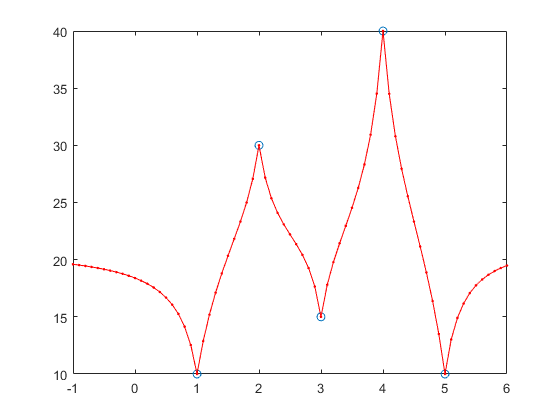


IDW je metoda, jejíž charakteristickou vlastností je tendence k průměru. Pro zadaný interval se k tomuto průmeru, pokud jsme vně, rychle blížíme, narozdíl např od lin. interpolace a jiných metod . Vysvělení tohoto chování je v GPI na straně 21.

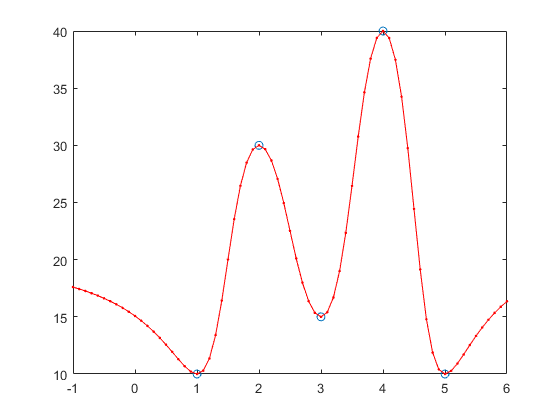
**Poznámka:** Zkratka NaN znamená NOT A NUMBER a může se objevit např. při dělění nulou.

**Úkol 5.** Napište, která volba mocniny se jeví jako nejvhodnější? Druhá mocnina se jeví vizuálně jako nejlepší.

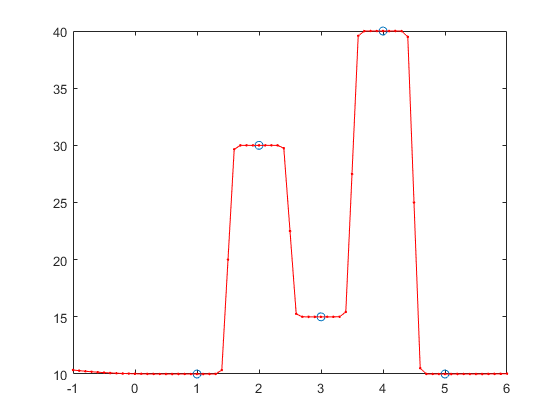
intpol\_IDW([1 2 3 4 5],[10 30 15 40 10],[-1:0.1:6], 1);



intpol\_IDW([1 2 3 4 5],[10 30 15 40 10],[-1:0.1:6], 2);



intpol\_IDW([1 2 3 4 5],[10 30 15 40 10],[-1:0.1:6], 10);



**Úkol 6.** Pročtení pasáží v knize, studování a přepisování poznámek a příkladů mi zabralo asi 8h, z části asi také proto, že jsem si stále ještě osahával MATLAB.

*“In theory, there’s no difference between theory and practice. In practice, there is.” Walter Savitch*

# Vliv velikosti okolí, anizotropie a rozložení bodů

*Bude zřejmě příště...*