Universidade Federal do Paraná
Programa de Pós-Graduação em Geologia

GEOL7048: Tópicos Especiais em Geologia Exploratória II

Métodos semiquantitativos

Saulo P. Oliveira

Departamento de Matemática, Universidade Federal do Paraná

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Aula 22

- Gridagem/interpolação

- Interpolação inversa

Gridagem/interpolação

Gridagem (*griding*) é a formatação de um dado irregularmente espaçado para uma malha (*grid*) regular, para fins de processamento e visualização.

Gridagem/interpolação

Gridagem (*griding*) é a formatação de um dado irregularmente espaçado para uma malha (*grid*) regular, para fins de processamento e visualização.

A gridagem normalmente supõe alguma forma de interpolação.

Gridagem/interpolação

Gridagem (*griding*) é a formatação de um dado irregularmente espaçado para uma malha (*grid*) regular, para fins de processamento e visualização.

A gridagem normalmente supõe alguma forma de interpolação.

Dizemos que uma função $f(x)$ interpola os pares ordenados

$$(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)$$

se for verdade que

$$\begin{cases} f(\boldsymbol{x}_1) & = & y_1 \\ f(\boldsymbol{x}_2) & = & y_2 \\ \vdots & & \vdots \\ f(\boldsymbol{x}_n) & = & y_n, \end{cases}$$

observando que $\boldsymbol{x}_i = x_i$, $\boldsymbol{x}_i = (x_{1,i}, x_{2,i})$ ou $\boldsymbol{x}_i = (x_{1,i}, x_{2,i}, x_{3,i})$.

Interpolação1D

Em 1D, a $f(x)$ interpola os pontos

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

se

$$\begin{cases} f(x_1) & = & y_1 \\ f(x_2) & = & y_2 \\ \vdots & & \vdots \\ f(x_n) & = & y_n. \end{cases}$$

Interpolação1D

Em 1D, a $f(x)$ interpola os pontos

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

se

$$\begin{cases} f(x_1) & = & y_1 \\ f(x_2) & = & y_2 \\ \vdots & & \vdots \\ f(x_n) & = & y_n. \end{cases}$$

Tipos mais comuns:

- interpolação polinomial
- interpolação polinomial por partes (ex: splines)
- interpolação racional (Padé)
- interpolação trigonométrica

Interpolação polinomial 1D

Buscamos

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

tal que

$$\begin{cases} f(x_1) & = & y_1 \\ f(x_2) & = & y_2 \\ \vdots & & \vdots \\ f(x_{n+1}) & = & y_{n+1} \end{cases}$$

Interpolação polinomial 1D

Buscamos

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

tal que

$$\begin{cases} f(x_0) & = & y_0 \\ f(x_1) & = & y_1 \\ f(x_2) & = & y_2 \\ \vdots & & \vdots \\ f(x_n) & = & y_n \end{cases}$$

Interpolação polinomial 1D

Buscamos

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

tal que

$$\begin{cases} a_n x_0^n + a_{n-1} x_0^{n-1} + \ldots + a_2 x_0^2 + a_1 x_0 + a_0 & = & y_0 \\ a_n x_1^n + a_{n-1} x_1^{n-1} + \ldots + a_2 x_1^2 + a_1 x_1 + a_0 & = & y_1 \\ a_n x_2^n + a_{n-1} x_2^{n-1} + \ldots + a_2 x_2^2 + a_1 x_2 + a_0 & = & y_2 \\ \vdots & & \vdots \\ a_n x_n^n + a_{n-1} x_n^{n-1} + \ldots + a_2 x_n^2 + a_1 x_n + a_0 & = & y_n \end{cases}$$

Interpolação polinomial 1D

Buscamos

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

tal que

$$\begin{bmatrix} x_0^n & x_0^{n-1} & \ldots & x_0^2 & x_0 & 1 \\ x_1^n & x_1^{n-1} & \ldots & x_1^2 & x_1 & 1 \\ x_2^n & x_2^{n-1} & \ldots & x_2^2 & x_2 & 1 \\ & & \vdots & & & \\ x_n^n & x_n^{n-1} & \ldots & x_n^2 & x_n & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Interpolação polinomial 1D

Buscamos

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

tal que

$$\begin{bmatrix} x_0^n & x_0^{n-1} & \ldots & x_0^2 & x_0 & 1 \\ x_1^n & x_1^{n-1} & \ldots & x_1^2 & x_1 & 1 \\ x_2^n & x_2^{n-1} & \ldots & x_2^2 & x_2 & 1 \\ & & \vdots & & & \\ x_n^n & x_n^{n-1} & \ldots & x_n^2 & x_n & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

A matriz acima (matriz de Vandermonde) é sempre inversível

Interpolação polinomial 1D

Buscamos

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

tal que

$$\begin{bmatrix} x_0^n & x_0^{n-1} & \ldots & x_0^2 & x_0 & 1 \\ x_1^n & x_1^{n-1} & \ldots & x_1^2 & x_1 & 1 \\ x_2^n & x_2^{n-1} & \ldots & x_2^2 & x_2 & 1 \\ & & \vdots & & & \\ x_n^n & x_n^{n-1} & \ldots & x_n^2 & x_n & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

A matriz acima (matriz de Vandermonde) é sempre inversível

Consequência: $f(x)$ existe e é único.

Interpolação polinomial 1D

No matlab/octave: `polyfit` e `polyval`

Exemplo:

```
>> n=5;
>> xo = linspace(-4,4,n+1);
>> yo = rand(1,n+1);
>> P = polyfit(xo,yo,n)
>> x=linspace(-4,4,100);
>> plot(xo,yo,'o',x,polyval(P,x))
```

Interpolação polinomial 1D

No matlab/octave: `polyfit` e `polyval`

Exemplo:

```
>> n=5;
>> xo = linspace(-4,4,n+1);
>> yo = rand(1,n+1);
>> P = polyfit(xo,yo,n)
>> x=linspace(-4,4,100);
>> plot(xo,yo,'o',x,polyval(P,x))
```

OBS: o cálculo torna-se instável quando `n` é grande.

Interpolação por partes 1D

Quando o número de pontos é grande (> 10), convém usar interpolação por partes:

```
y = interp1(xo,yo,x,'method')
```

Interpolação por partes 1D

Quando o número de pontos é grande (> 10), convém usar interpolação por partes:

```
y = interp1(xo,yo,x,'method')
```

Exemplo:

```
>> n=15;
>> xo = linspace(-4,4,n+1);
>> yo = rand(1,n+1);
>> x = linspace(-4,4,100);
>> y = interp1(xo,yo,x,'spline');
>> plot(xo,yo,'o',x,y)
```

Interpolação por partes 1D

Quando o número de pontos é grande (> 10), convém usar interpolação por partes:

```
y = interp1(xo,yo,x,'method')
```

Exemplo:

```
>> n=15;
>> xo = linspace(-4,4,n+1);
>> yo = rand(1,n+1);
>> x = linspace(-4,4,100);
>> y = interp1(xo,yo,x,'spline');
>> plot(xo,yo,'o',x,y)
```

Outros métodos:

`'nearest','previous','next','linear','pchip'`

Interpolação por partes 1D

Quando o número de pontos é grande (> 10), convém usar interpolação por partes:

```
y = interp1(xo,yo,x,'method')
```

Exemplo:

```
>> n=15;
>> xo = linspace(-4,4,n+1);
>> yo = rand(1,n+1);
>> x = linspace(-4,4,100);
>> y = interp1(xo,yo,x,'spline');
>> plot(xo,yo,'o',x,y)
```

Outros métodos:

`'nearest','previous','next','linear','pchip'`

Sugestão: testar `xo = sort(rand(1,n+1));`

```
x = linspace(xo(1),xo(n+1),100);
```

Interpolação por partes 2D

O comando para interpolação 2D é semelhante ao comando 1D:

```
y = interp2(xo,yo,zo, x,y ,'method')
```

Interpolação por partes 2D

O comando para interpolação 2D é semelhante ao comando 1D:

```
y = interp2(xo,yo,zo, x,y ,'method')
```

Exemplo:

```
>> to = linspace(-4,4,10);
>> [xo,yo] = meshgrid(to);
>> zo = rand(10);
```

Interpolação por partes 2D

O comando para interpolação 2D é semelhante ao comando 1D:

```
y = interp2(xo,yo,zo, x,y ,'method')
```

Exemplo:

```
>> to = linspace(-4,4,10);
>> [xo,yo] = meshgrid(to);
>> zo = rand(10);
>> t = linspace(-4,4,50);
>> [x,y] = meshgrid(t);
>> z = interp2(xo,yo,zo,x,y,'spline');
```

Interpolação por partes 2D

O comando para interpolação 2D é semelhante ao comando 1D:

```
y = interp2(xo,yo,zo, x,y ,'method')
```

Exemplo:

```
>> to = linspace(-4,4,10);
>> [xo,yo] = meshgrid(to);
>> zo = rand(10);
>> t = linspace(-4,4,50);
>> [x,y] = meshgrid(t);
>> z = interp2(xo,yo,zo,x,y,'spline');
>> plot3(xo(:),yo(:),zo(:),'o')
>> hold on
>> mesh(x,y,z)
>> hold off
```

Interpolação inversa

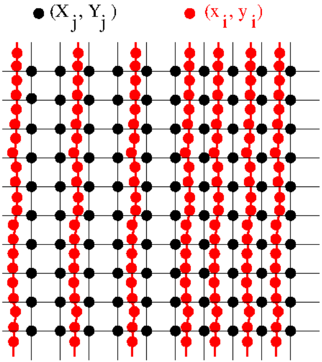
Ideia: Conhecidos os dados $\{x_i, y_i, z_i\}_{i=1}^{m}$ e o grid $\{X_j, Y_j\}_{j=1}^{n}$ desejado,

Encontrar

$$f(x,y) = \sum_{j=1}^{n} w_j(x,y) f_j$$

tal que

$$f(x_i, y_i) = z_i \quad 1 \leq i \leq m$$



Funções-peso: $w_j(x,y) = \dfrac{\exp(-br_j^2)}{\sum_{l=1}^{m} \exp(-br_l^2)}, \; r_j^2 = (x - X_j)^2 + (y - Y_j)^2$

L. Guo et al (2012). Gridding aeromagnetic data using inverse interpolation. Geophys. J. Int., 189, 1353-1360

Interpolação inversa

Sistema linear $A\boldsymbol{x} = \boldsymbol{b}$:

$$\begin{cases} f(x_1, y_1) & = & z_1 \\ f(x_2, y_2) & = & z_2 \\ \vdots & & \vdots \\ f(x_m, y_m) & = & z_m \end{cases}$$

L. Guo et al (2012). Gridding aeromagnetic data using inverse interpolation. Geophys. J. Int., 189, 1353-1360

Interpolação inversa

Sistema linear $A\boldsymbol{x} = \boldsymbol{b}$:

$$
\left\{
\begin{array}{ccc}
w_1(x_1, y_1)f_1 + w_2(x_1, y_1)f_2 + \cdots + w_n(x_1, y_1)f_n & = & z_1 \\
w_1(x_2, y_2)f_1 + w_2(x_2, y_2)f_2 + \cdots + w_n(x_2, y_2)f_n & = & z_2 \\
\vdots & & \vdots \\
w_1(x_m, y_m)f_1 + w_2(x_m, y_m)f_2 + \cdots + w_n(x_m, y_m)f_n & = & z_n
\end{array}
\right.
$$

L. Guo et al (2012). Gridding aeromagnetic data using inverse interpolation. Geophys. J. Int., 189, 1353-1360

Interpolação inversa

Sistema linear $A\boldsymbol{x} = \boldsymbol{b}$:

$$
\begin{bmatrix}
w_1(x_1, y_1) & w_2(x_1, y_1) & \cdots & w_n(x_1, y_1) \\
w_1(x_2, y_2) & w_2(x_2, y_2) & \cdots & w_n(x_2, y_2) \\
 & & \vdots & \\
w_1(x_m, y_m) & w_2(x_m, y_m) & \cdots & w_n(x_m, y_m)
\end{bmatrix}
\begin{bmatrix}
f_1 \\ f_2 \\ \vdots \\ f_n
\end{bmatrix}
=
\begin{bmatrix}
z_1 \\ z_2 \\ \vdots \\ z_m
\end{bmatrix}
$$

L. Guo et al (2012). Gridding aeromagnetic data using inverse interpolation. Geophys. J. Int., 189, 1353-1360

Interpolação inversa

Sistema linear $A\boldsymbol{x} = \boldsymbol{b}$:

$$\begin{bmatrix} w_1(x_1, y_1) & w_2(x_1, y_1) & \cdots & w_n(x_1, y_1) \\ w_1(x_2, y_2) & w_2(x_2, y_2) & \cdots & w_n(x_2, y_2) \\ & & \vdots & \\ w_1(x_m, y_m) & w_2(x_m, y_m) & \cdots & w_n(x_m, y_m) \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}$$

Regularização:

$$\begin{bmatrix} A \\ \lambda L \end{bmatrix} \boldsymbol{x_\lambda} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix}, \qquad L(x_i, y_i) \approx \frac{\partial^2 f}{\partial x^2}(x_i, y_i) + \frac{\partial^2 f}{\partial y^2}(x_i, y_i)$$

L. Guo et al (2012). Gridding aeromagnetic data using inverse interpolation. Geophys. J. Int., 189, 1353-1360