

## Aproksymacja funkcji metodą najmniejszych kwadratów

### Teoria

Interpolacja polega na znajdowaniu krzywej przechodzącej przez wszystkie węzły. Zdarzają się jednak sytuacje, w których dane te mogą być obarczone błędem, np. gdy są one odczytywane z pomiarów. W takim wypadku bardziej interesujące jest zachowanie ogólnego charakteru krzywej, która te punkty przybliży niż przeprowadzanie jej przez wszystkie węzły. Do tego zadania wykorzystywana jest aproksymacja metodą najmniejszych kwadratów. W aproksymacji metodą najmniejszych kwadratów mamy z góry zadaną bazę funkcji za pomocą których chcemy przybliżyć funkcję poszukiwaną postaci:

$$F(x) = c_1 f_1(x) + c_2 f_2(x) + \dots + c_n f_n(x) \quad (1)$$

dla zadanego zbioru punktów  $(x_j, y_j)$ ,  $j = 1, \dots, m$ . Jak wynika z tego wzoru, poszukiwana funkcja  $F(x)$  jest liniową kombinacją funkcji bazowych  $f_i(x)$ ,  $i = 1, \dots, n$ . Funkcje bazowe mogą być dowolne, dopóki są liniowo niezależne.

Niewiadomymi w tak postawionym zadaniu są współczynniki  $c_i$ .

Ponieważ, w ogólnym przypadku, tak zadana funkcja poszukiwana nie będzie pokrywała się ze wszystkimi węzłami. W zależności od doboru współczynników  $c_i$  może lepiej przybliżać pewne węzły a gorzej inne. Należy zatem dobrać odpowiednie kryterium porównawcze. Możemy obliczyć pionową odległość funkcji od każdego z węzłów:

$$r_j = y_j - F(x_j) \quad (2)$$

Ale ponieważ tak zapisane  $r_j$  może być zarówno dodatnie jak i ujemne, to proste zsumowanie tych wartości nie nadaje się do porównywania różnych krzywych. Lepszym wyborem jest użycie  $r_j^2$ , wtedy:

$$\rho = \sum_{j=1}^m r_j^2 \quad (3)$$

jest wartością, której minimum będzie określało krzywą najlepiej dopasowaną do węzłów.

### Dopasowanie prostej do zbioru punktów

Zacznijmy od najprostszego przypadku polegającego na przybliżeniu zbioru punktów krzywą. Poszukiwana funkcja ma zatem postać:

$$F(x) = c_1 x + c_2 \quad (4)$$

a jej bazą są funkcje:

$$f_1(x) = x \quad f_2(x) = 1$$

Wyprowadzone wcześniej wartości  $r_j$  i  $\rho$  możemy zatem wyrazić jako:

$$r_j = y_j - F(x_j) = y_j - (c_1 x_j + c_2)$$

$$\rho = \sum_{j=1}^m r_j^2 = \sum_{j=1}^m [y_j - (c_1 x_j + c_2)]^2$$

Naszym celem jest zminimalizowanie funkcji  $\rho$ . Jedynymi niewiadoamymi w tym wypadku są wartości  $c_1$  i  $c_2$ . Możemy zapisać zatem:  $\rho = \rho(c_1, c_2)$ . Warunkiem koniecznym minimum jest zerowanie się pierwszych pochodnych, zatem:

$$\frac{\partial \rho}{\partial c_1} = 0 \quad \text{i} \quad \frac{\partial \rho}{\partial c_2} = 0$$

Wykonując różniczkowanie i podstawiając warunek na minimum otrzymujemy:

$$\frac{\partial \rho}{\partial c_1} = \sum_{j=1}^m \frac{\partial}{\partial c_1} [y_j - (c_1 x_j + c_2)]^2 = \sum_{j=1}^m 2(-x_j)[y_j - (c_1 x_j + c_2)] = 0$$

$$\frac{\partial \rho}{\partial c_2} = \sum_{j=1}^m \frac{\partial}{\partial c_2} [y_j - (c_1 x_j + c_2)]^2 = \sum_{j=1}^m 2(-1)[y_j - (c_1 x_j + c_2)] = 0$$

Po uporządkowaniu możemy to zapisać w postaci macierzowej:

$$\begin{bmatrix} \sum x_j^2 & \sum x_j \\ \sum x_j & m \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \sum x_j y_j \\ \sum y_j \end{bmatrix} \quad (5)$$

Dla dwóch wektorów

$$\mathbf{x} = [x_1, x_2, \dots, x_m] \quad \text{ i } \quad \mathbf{y} = [y_1, y_2, \dots, y_m]$$

możemy zdefiniować iloczyn wektorowy jako:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{j=1}^m x_j y_j \quad (6)$$

Korzystając z tej definicji możemy zapisać układ (5) następująco:

$$\begin{bmatrix} \langle \mathbf{x}, \mathbf{x} \rangle & \langle \mathbf{1}, \mathbf{x} \rangle \\ \langle \mathbf{1}, \mathbf{x} \rangle & \langle \mathbf{1}, \mathbf{1} \rangle \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \langle \mathbf{x}, \mathbf{y} \rangle \\ \langle \mathbf{1}, \mathbf{y} \rangle \end{bmatrix} \quad (7)$$

gdzie:

$$\mathbf{1} = [1, 1, \dots, 1]$$

Z tej postaci łatwo już wyliczyć nieznanne współczynniki  $c_i$ . Algorytm ten można rozszerzyć na większą liczbę funkcji bazowych lub zastosować inne podejście.

#### Metoda przededefiniowanego układu równań

Rozważając ponownie problem znalezienia prostej przybliżającej  $m$  węzłów możemy zapisać zbiór równań:

$$\begin{aligned} c_1 x_1 + c_2 &= y_1 \\ c_1 x_2 + c_2 &= y_2 \\ &\dots \\ c_1 x_m + c_2 &= y_m \end{aligned}$$

lub w postaci macierzowej:

$$\mathbb{A} \mathbf{c} = \mathbf{y} \quad (8)$$

gdzie:

$$\mathbb{A} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_m & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Ponieważ w ogólnym przypadku wszystkie węzły nie będą leżały na jednej prostej, to równania te nie będą spełnione, zatem wektor residuów

$$\mathbf{r} = \mathbf{y} - \mathbb{A}\mathbf{c} \quad (9)$$

będzie niezerowy. W tej metodzie minimalizacji podlega kwadrat normy wektora residuów:

$$\rho = \|\mathbf{r}\|_2^2 = \mathbf{r}^T \mathbf{r} \quad (10)$$

Przekształcając otrzymujemy:

$$\begin{aligned} \rho = \|\mathbf{r}\|_2^2 = \mathbf{r}^T \mathbf{r} &= (\mathbf{y} - \mathbb{A}\mathbf{c})^T (\mathbf{y} - \mathbb{A}\mathbf{c}) \\ &= \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbb{A} - \mathbf{c}^T \mathbb{A}^T \mathbf{y} + \mathbf{c}^T \mathbb{A}^T \mathbb{A} \mathbf{c} \\ &= \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbb{A} \mathbf{c} + \mathbf{c}^T \mathbb{A}^T \mathbb{A} \mathbf{c} \end{aligned}$$

Warunkiem koniecznym minimum jest zerowanie się pierwszych pochodnych, zatem:

$$\frac{\partial \rho}{\partial \mathbf{c}} = 0$$

Wykonując różniczkowanie otrzymujemy:

$$\frac{\partial \rho}{\partial \mathbf{c}} = -2\mathbb{A}^T \mathbf{y} + 2\mathbb{A}^T \mathbb{A} \mathbf{c}$$

Korzystając z warunku na minimum otrzymujemy:

$$0 = -2\mathbb{A}^T \mathbf{y} + 2\mathbb{A}^T \mathbb{A} \mathbf{c}$$

lub inaczej

$$(\mathbb{A}^T \mathbb{A}) \mathbf{c} = \mathbb{A}^T \mathbf{y} \quad (11)$$

Są to tzw. *równania normalne* aproksymacji metodą najmniejszych kwadratów.

Dopasowanie metodą najmniejszych kwadratów dla dowolnej bazy funkcji

W ogólnym przypadku doboru bazy funkcji  $f_i(x)$ ,  $i = 1, \dots, n$  będziemy mieli:

$$y = F(x) = c_1 f_1(x) + c_2 f_2(x) + \dots + c_n f_n(x)$$

Niewiadomymi będą tu współczynniki  $c_i$ . Celem jest znalezienie takich współczynników  $c_1, \dots, c_n$ , które minimalizują kwadraty pionowych odległości funkcji  $F(x)$  od węzłów  $(x_j, y_j)$ ,  $j = 1, \dots, m$ . Dla tak postawionego zadania możemy zapisać układ równań:

$$\begin{aligned} c_1 f_1(x_1) + c_2 f_2(x_1) + \dots + c_n f_n(x_1) &= y_1 \\ c_1 f_1(x_2) + c_2 f_2(x_2) + \dots + c_n f_n(x_2) &= y_2 \\ &\dots \\ c_1 f_1(x_m) + c_2 f_2(x_m) + \dots + c_n f_n(x_m) &= y_m \end{aligned}$$

lub w postaci macierzowej:

$$\mathbb{A}\mathbf{c} = \mathbf{y}$$

gdzie:

$$\mathbb{A} = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \dots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \dots & f_n(x_2) \\ \vdots & \vdots & \dots & \vdots \\ f_1(x_m) & f_2(x_m) & \dots & f_n(x_m) \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Dochodzimy do podobnej postaci układu jak w poprzednim podrozdziale różniącym się tylko definicją macierzy  $A$ . Naszym zadaniem jest zminimalizowanie wartości  $\rho = \|\mathbf{y} - A\mathbf{c}\|_2^2$ . Postępując w przedstawiony wcześniej sposób otrzymujemy równania normalne postaci:

$$(A^T A)\mathbf{c} = A^T \mathbf{y}$$

Wyznaczając wektor  $c$  z tego równania otrzymujemy nieznanne współczynniki.

Skrypt:

```
function [c,R2,rout] = fitnorm(x,y,basefun)
% Funkcja aproksymuje grupe wezlow (x_i,y_i), i=1,...,m metoda
% najmniejszych kwadratow
%
% Dane wejsciowe:   x,y       = wektory zawierajace wspolrzedne wezlow
%                  basefun = (string) nazwa pliku ktory wylicza wartosci
%                  macierzy A. Kolumny A sa wartosciami funkcji
%                  bazowych dla wspolrzednych x wezlow.
%
% Dane wyjsciowe:  c = wektor wspolczynnikow poszukiwanej funkcji
%                  R2 = (opcjonalnie) wspolczynnik korelacji
%                  r = (opcjonalnie) wektor residuow

if length(y)~= length(x); error('x i y maja rozne wymiary'); end

A = feval(basefun,x(:)); % Macierz wspolczynnikow ukkladu przeddefiniowanego
c = (A'*A)\(A'*y(:));   % Rozwiazuje rownanie normalne, y(:) jest kolumna
if nargin>1
    r = y(:) - A*c;      % Residua w wezlach
    [m,n] = size(A);
    R2 = 1 - (m-1)/(m-n-1)*(norm(r)/norm(y-mean(y)))^2;
    if nargin>2, rout = r; end
end
```

Zadanie:

Wyznacz aproksymację punktów:

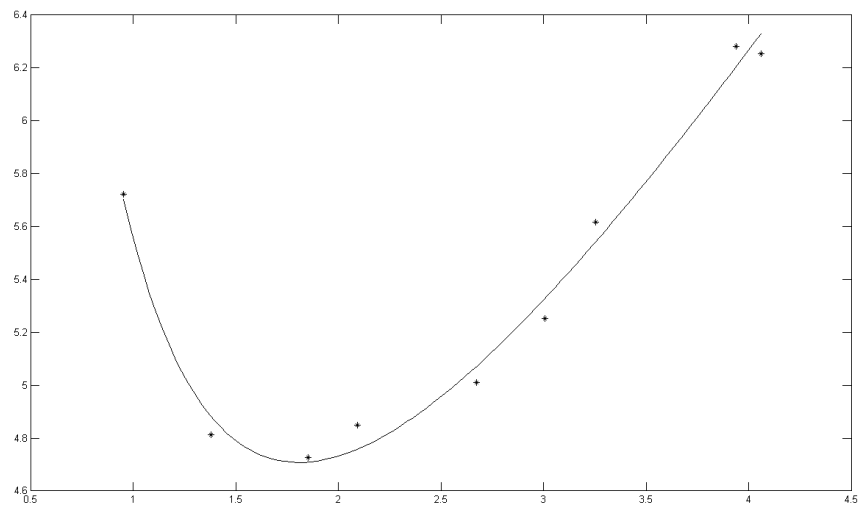
(0.955, 5.722), (1.38 4.812), (1.854, 4.727), (2.093, 4.850), (2.674, 5.011),  
 (3.006, 5.253), (3.255, 5.617), (3.940, 6.282), (4.060, 6.255)

Używając jako funkcji bazowych:

$$\frac{1}{x}, \quad x, \quad x^2$$

Rozwiązanie w programie MATLAB:

```
clc;
x = [0.955 1.38 1.854 2.093 2.674 3.006 3.255 3.940 4.060];
y = [5.722 4.812 4.727 4.850 5.011 5.253 5.617 6.282 6.255];
fun1 = inline('1./x x');
c = fitnorm(x,y,fun1);
xf = linspace(min(x),max(x))';
Af = feval(fun1,xf);
yf = Af*c;
plot (x,y,'*',xf,yf,'-');
```



Spis używanych komend:

<code>inline(expr)</code>	Tworzy funkcję zawartą w <code>expr</code>
---------------------------	--