Programování pro 6. ročník
Metoda nejmenších čtverců

David Martinek

4. března 2015

Abstrakt

Metoda nejmenších čtverců je aproximační metoda používaná často pro aproximaci dat získaných měřením. U měřících úloh často víme, jaký tvar má mít výsledný graf, ale naměřené hodnoty jsou zatíženy chybou měření, proto nelze použít interpolaci. Naším cílem tedy je aproximovat naměřené hodnoty známou funkcí tak, aby byla co nejblíže naměřeným hodnotám.

1 Motivace

Měření je v technických oborech velmi častá činnost. Setkáte se s ním ve fyzice, elektrotechnice, strojním inženýrství, geografii, architektuře a v množství dalších oborů. Obvykle se měří vzájemná závislost dvou veličin. Při fyzikálním pokusu s tlakovým hrncem můžeme například měřit závislost tlaku na teplotě, při elektrotechnickém měření rezistoru můžeme měřit závislost proudu na napětí, a nakonec například v geografii můžeme měřit výšku (nebo úhel) v závislosti na vzdálenosti od referenčního bodu.

Výsledkem měření je vždy tabulka naměřených hodnot a následně graf, kde jsou zaneseny hodnoty *nezávislé proměnné* a odpovídající *závislé proměnné*. Nezávisle proměnná se vynáší na osu x a z našich příkladů by to byly hodnoty teplot, napětí a vzdálenosti. Závisle proměnná se vynáší na osu y, kam bychom v našich příkladech vynesli tlak, proud a výšku.

1.1 Aproximace versus interpolace

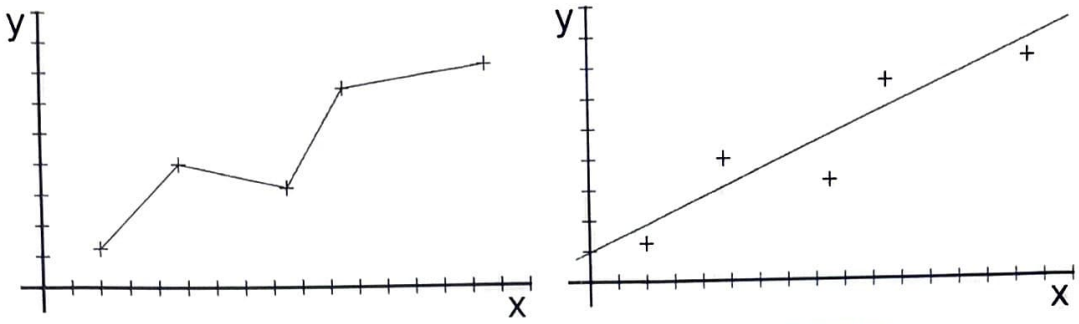
Protože nejsme schopni změřit spojitou závislost obou veličin, obvykle se spokojíme s dostatečným, ale konečným počtem změřených diskrétních bodů. Předpokládáme, že mezi těmito body závislost nějakým způsobem navazuje. To znamená, že není ani nespojitá („díry" či ostré skoky), ani mezi nimi nejsou různé nepravidelné špičky.

Na obrázku 1 jsou vidět dvě možná řešení, jak chybějící místa grafu doplnit. Varianta 1a se nazývá *interpolace*. Při ní požadujeme, aby interpolační funkce procházela všemi naměřenými body. Varianta 1b znázorňuje *aproximaci*, u níž se snažíme nahradit naměřené hodnoty pomocí známé matematické funkce. Už nevyžadujeme, aby naměřenými body procházela, ale musí se jim co nejvíce blížit.

Na obou obrázcích jsou použity lineární funkce. Jak interpolace, tak aproximace však může být realizována i pomocí hladkých křivek (polynomy vyšších řádů, trigonometrické funkce, logaritmus, a podobně).

1.2 Co tedy použít?

Co tedy pro doplnění naměřeného grafu použít? Interpolaci nebo aproximaci? Odpověď závisí na povaze měřených hodnot. Když se vrátíme k našim ukázkovým měřícím úlohám, u geografické úlohy



(a) Interpolace (b) Aproximace

Obrázek 1: Graf naměřených hodnot a ukázky lineární interpolace (a) a aproximace pomocí lineární funkce (b).

s měřením výšky by bylo logičtější použít interpolaci, u fyzikální a elektrotechnické úlohy zase aproximaci. Povrch země totiž nelze popsat pomocí jednoduché matematické funkce. Neočekáváme, že kopec bude mít tvar odpovídající například tvaru funkce sinus. Pro účel, který zde sledujeme, tedy získat představu o skutečném tvaru terénu, tedy interpolace poskytne lepší výsledky než aproximace.

U fyzikálních problémů, kam ostatně patří i elektrotechnické problémy, je naopak obvyklé, že vztah měřených veličin *je* popsán nějakým matematickým vztahem. Například při měření obyčejného rezistoru víme (nebo ověřujeme), že vztah mezi napětím a proudem je lineární. Jde nám o to zjistit, *jaký přesně* tento vztah je. V tomto případě tedy chceme zjistit parametry oné lineární funkce (tedy víme, že jde o lineární funkci, ale neznáme její sklon a posunutí).

V ideálním světě bychom i u těchto úloh vystačili s interpolací. Kdybychom uměli měřit přesně a život nám nekomplikovaly jevy jako náhodný šum, stačilo by nám propojit všechny naměřené body jedinou přímkou a bylo by hotovo. My ale přesně měřit neumíme. *Při měření vždy vznikají chyby.*

Tyto chyby mají naštěstí tendenci náhodně oscilovat kolem přesného řešení, proto pro doplnění grafu používáme aproximaci. Při ní se snažíme známý tvar funkční závislosti umístit tak, aby výsledná funkce byla *co nejblíže všem naměřeným hodnotám*. U polynomiálních funkcí „ladíme" jejich přesný tvar pomocí koeficientů, které ovlivňují jejich sklon a posunutí. Hledáním umístění se tedy myslí nalezení správných koeficientů aproximační funkce.

2 Princip metody nejmenších čtverců

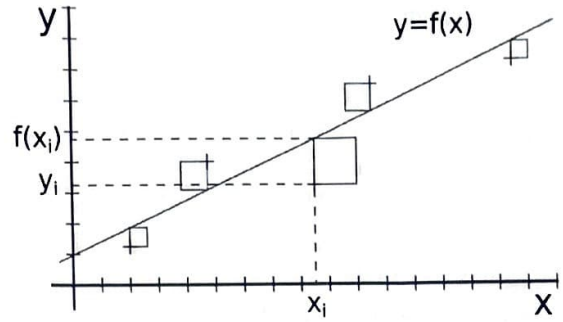
Abychom mohli metodu nejmenších čtverců použít, musíme splnit tyto dva požadavky:

1. Musíme mít naměřené hodnoty.
2. Musíme vědět, jakou funkcí chceme hodnoty aproximovat. PODMÍNKA APROXIMACE

Metoda nejmenších čtverců pak zajistí právě ten požadavek, aby zvolená aproximační funkce byla co nejblíže všem naměřeným hodnotám. K tomu, jak tento požadavek splnit, nás přivede následující úvaha.

Zvolená funkce bude nejblíže všem naměřeným bodům zároveň, když bude součet rozdílů mezi naměřenou hodnotou a hodnotou naší funkce nejmenší možný. Protože nevíme, kolik to je, vyjádříme si tedy tento součet jako funkci F.

Jaká matematická metoda slouží k hledání minima funkce? Odpověď je – první derivace. První derivace vrací v daném bodě x směrnici tečny k derivované funkci. Pokud má derivovaná funkce v bodě x minimum, je tečnou konstantní funkce (rovnoběžná s osou x) a derivace je zde nulová. Protože jsme



Obrázek 2: Čtverce použité při aproximaci pomocí lineární funkce.

si funkci F, u níž chceme najít minimum, stanovili jako součet vzdáleností mezi body, je zřejmé, že jde o lineární funkci. Problémem ovšem je, že taková funkce žádné detekovatelné minimum nemá. Co s tím?

Řešení tohoto problému napadlo jako prvního C. F. Gausse, který tuto metodu použil v roce 1795. Jestliže součet odchylek není k ničemu, proč nezkusit sečíst jejich druhé mocniny, tedy čtverce nad všemi odchylkami? Takto vznikne polynom druhého stupně, tedy kvadratická funkce, jejímž grafem je parabola. Protože součet čtverců vzdáleností bude vždy kladný, bude mít tato funkce vždy právě jedno minimum, které jsme schopni nalézt pomocí derivace.

Tato vlastnost je také důvodem, proč nemáme metodu nejmenších kvádrů, či jiných vícerozměrných těles. Grafy polynomů vyšších stupňů totiž nemají jediné minimum a buďto by pro tento účel byly nepoužitelné nebo zbytečně složité.

Úlohou metody nejmenších čtverců je tedy najít koeficienty zvolené aproximační funkce tak, aby součet čtverců odchylek od naměřených hodnot byl nejmenší možný.

3 Odvození pro lineární funkci

Metodě nejmenších čtverců pro lineární funkci se také říká *lineární regrese*. Při odvození řešení pro aproximaci lineární funkcí vycházejme z obrázku 2. Nejprve musíme vyjádřit funkci F reprezentující součet všech čtverců vzdáleností mezi skutečnou hodnotou $f(x_i)$ a naměřenou hodnotou y_i.

$$F = \sum_i (f(x_i) - y_i)^2 \tag{1}$$

Zvolená aproximační funkce je lineární, vyjádříme jí tudíž pomocí rovnice přímky.

$$f(x) = ax + b \tag{2}$$

Nyní musíme do rovnice 1 dosadit rovnici 2.

$$F = \sum_i (ax_i + b - y_i)^2 \tag{3}$$

Rozepíšeme-li pro názornost sumu, dostaneme stejnou rovnici ve tvaru

$$F = (ax_1 + b - y_1)^2 + (ax_2 + b - y_2)^2 + \cdots + (ax_n + b - y_n)^2 \tag{4}$$

Zamyslíme-li se nad touto rovnicí, zjistíme, že všechny hodnoty x_i a y_i jsou známé, protože to jsou naše změřené hodnoty. V této rovnici zůstávají neznáme pouze hodnoty koeficientů a a b. Funkce F je tedy funkcí dvou neznámých proměnných $F(a, b)$.

Tuto funkci nyní musíme derivovat. Fakt, že jde o funkci dvou neznámých nám trochu komplikuje situaci. Funkci nelze derivovat zároveň podle dvou proměnných. Budeme muset derivovat dvakrát, jednou podle proměnné a, podruhé podle proměnné b. Takovým derivacím se říká *parciální derivace*.

Když budeme derivovat podle proměnné a, stanou se pro tuto chvíli všechny ostatní proměnné konstantami a tak s nimi budeme zacházet i při derivování. Tuto derivaci označíme $F'_a(a,b)$, abychom znázornili, že nyní derivujeme podle proměnné a, zatímco b pro tuto chvíli považujeme za konstantu[1]. Až budeme derivovat podle proměnné b, bude to platit analogicky a derivaci označíme $F'_b(a,b)$.

To, co budeme derivovat je složená funkce. Máme zde druhou mocninu, která je aplikována na lineární funkci. Uplatníme tedy pravidla pro derivaci mocniny a složené funkce.

$$(x^a)' = ax^{a-1} \tag{5}$$

$$\left[f(g(x))\right]' = f'(g(x))g'(x) \tag{6}$$

Funkci F tedy nejprve derivujeme podle proměnné a a potom podle proměnné b.

$$F'_a(a,b) = 2(ax_1+b-y_1)\cdot x_1 + 2(ax_2+b-y_2)\cdot x_2 + \cdots + 2(ax_n+b-y_n)\cdot x_n \tag{7}$$

$$F'_b(b,b) = 2(ax_1+b-y_1)\cdot 1 + 2(ax_2+b-y_2)\cdot 1 + \cdots + 2(ax_n+b-y_n)\cdot 1 \tag{8}$$

Nyní budeme hledat minimum funkce F. To leží tam, kde je derivace rovna nule. Funkce vzniklé derivací tedy položíme rovny nule a tím nám vzniknou dvě rovnice o dvou neznámých. Nyní z těchto rovnic pomocí algebraických úprav osamostatníme neznámé a konstantní koeficienty, abychom byli schopni tuto soustavu vyřešit.

$$2(ax_1+b-y_1)\cdot x_1 + 2(ax_2+b-y_2)\cdot x_2 + \cdots + 2(ax_n+b-y_n)\cdot x_n = 0 \tag{9}$$

$$2(ax_1+b-y_1) + 2(ax_2+b-y_2) + \cdots + 2(ax_n+b-y_n) = 0 \tag{10}$$

Nyní se můžeme zbavit dvojek vynásobením rovnic hodnotou $\frac{1}{2}$. Také musíme roznásobit závorky.

$$ax_1^2+bx_1-x_1y_1+ax_2^2+bx_2-x_2y_2+\cdots+ax_n^2+bx_n-x_ny_n = 0 \tag{11}$$

$$ax_1+b-y_1+ax_2+b-y_2+\cdots+ax_n+b-y_n = 0 \tag{12}$$

Nyní vytkneme proměnné a zbytek převedeme na druhou stranu.

$$a(x_1^2+x_2^2+\cdots+x_n^2)+b(x_1+x_2+\cdots+x_n) = x_1y_1+x_2y_2+\cdots+x_ny_n \tag{13}$$

$$a(x_1+x_2+\cdots+x_n)+bn = y_1+y_2+\cdots+y_n \tag{14}$$

Zkráceně tedy můžeme soustavu zapsat takto:

$$a\sum_i x_i^2 + b\sum_i x_i = \sum_i x_iy_i \tag{15}$$

$$a\sum_i x_i + bn = \sum_i y_i \tag{16}$$

Tuto soustavu lze nyní řešit známým způsobem buďto odvozením jedné proměnné z jedné rovnice a dosazením do druhé nebo například Gaussovou eliminační metodou.

Pokud bychom si zvolili pro aproximaci polynomiální funkci jiných řádů, bude odvození probíhat principiálně podobným způsobem. Zde jsme aproximovali pomocí polynomu 1. řádu, hledali jsme dva koeficienty, tudíž jsme prováděli dvě parciální derivace a nakonec jsme řešili soustavu dvou rovnic o dvou neznámých. Pokud bychom pro aproximaci použili polynom n-tého řádu, budeme hledat $n+1$ koeficientů, provádět $n+1$ parciálních derivací a řešit soustavy n rovnic o n neznámých.

[1] Zejména u fyzikálních úloh se můžete setkat s jiným značením parciálních derivací $F'_a(a,b) \equiv \frac{\partial F}{\partial a}(a,b)$. Obě tyto notace znamenají totéž. V anglické literatuře se lze též setkat s označením $F'_b(a)$ s významem derivace podle proměnné a, zatímco b nyní zůstává konstantou.

4 Implementační poznámky

Při implementaci této metody v jazyce C si musíme ujasnit několik věcí. Předně nebudeme programovat samotné odvozování. Ještě před samotným programováním musíme zvolit, jakou aproximační funkci použijeme. Teprve poté můžeme provést odvození (na papír) a odvozené řešení naprogramovat.

Dále si musíme ujasnit, jaké jsou vstupy a výstupy tohoto algoritmu. Vstupem určitě bude tabulka naměřených hodnot. Potřebujeme znát souřadnice bodů, tedy hodnoty x_i a y_i. Tyto hodnoty budou uloženy v polích. Pole s hodnotami x_i bychom si mohli ušetřit, pokud by bylo dáno, že hodnoty y_i byly měřeny v přesně definovaných rozestupech hodnot x_i. My se ale budeme držet obecnějšího řešení, kdy tento předpoklad neplatí.

Naměřené hodnoty by bylo možné ukládat i do dvojrozměrného pole, případně si pro dvojici hodnot vytvořit strukturu reprezentující souřadnice bodu a pracovat s polem struktur. Tyto varianty však začnou mít cenu až u aproximace pomocí polynomiálních funkcí vyšších řádů. Pro lineární regresi si vystačíme se dvěma jednorozměrnými poli.

Výstupem našeho algoritmu budou hodnoty koeficientů lineární funkce a a b.

Budeme pracovat s desetinnými čísly s dvojitou přesností. Tento algoritmus realizujeme jako funkci `linRegrese`.

```
1  void linRegrese(double x[], double y[], unsigned int n, double *a, double *b)
2  {
3    double sumx = 0.0;
4    double sumx2 = 0.0;
5    double sumy = 0.0;
6    double sumxy = 0.0;
7    for(unsigned int i = 0; i < n; ++i)
8    {
9      // výpočet průběžných sum
10   }
11
12   // řešení soustavy lineárních rovnic například zavoláním funkce pro
13   // Gaussovu eliminační metodu
14
15   *a = // výsledek řešení soustavy
16   *b = // výsledek řešení soustavy
17 }
```

Ve výše uvedeném náčrtu řešení si všimněte parametrů funkce. Kromě polí s naměřenými hodnotami `x` a `y` je potřeba předat funkci i velikost polí `n`. Výsledné hodnoty koeficientů `a` a `b` jsou pochopitelně předávány odkazem, abychom s výsledky mohli dále pracovat.

Samotné řešení předpokládá výpočet čtyř součtů (sum) přes celá pole `x` a `y`. Nebylo by však dobrým řešením, kdyby se ve funkci vyskytly čtyři samostatné cykly pro každý součet. Všechny čtyři součty lze provést v rámci jediného cyklu. V tomto případě to bude přehlednější řešení, méně náchylné k chybám.

Nakonec připomínám, že je potřeba vypočítat opravdu všechny čtyři sumy. Pokud vás napadlo, že `sumxy` se dá efektivněji spočítat jako `sumxy = sumx * sumy`, pak si zkuste tento vztah rozepsat na papír. Suma součinů opravdu není rovna součinu sum.

4. 3. 2025

$f(x) : a$

Odvození pro absolutní funkci

$$F = \sum_i \left(f(x_i) - y_i \right)^2$$

$$F = \sum_i \left(a - y_i \right)^2$$

$$F'_a = \sum_i 2 \cdot (a - y_i) \cdot 1$$

POLOŽÍM DERIVACI NULE

$$\sum_i 2 \cdot (a - y_i) = 0 \quad /:2$$

$$\sum_i (a - y_i) = 0$$

$$na - \sum_i y_i = 0$$

$$a = \frac{\sum_i y_i}{n}$$

⇒ ARITMETICKÝ PRŮMĚR

Odvození pro lineární funkci

$$f(x) = ax_i + b$$

$$F = \sum_i \left(f(x_i) - y_i \right)^2$$

$$F = \sum_i \left(ax_i + b - y_i \right)^2$$

2 neznámé ⇒ 2 rovnice

$$F'_a(a,b) = \sum_i 2 \cdot (ax_i + b - y_i) \cdot 1$$

$$\sum_i 2 \cdot (ax_i + b - y_i) = 0 \quad /:2$$

NEDOKONČENÉ!

$$\sum_i (ax_i + b - y_i) = 0$$

$$a \sum_i x_i^2 + b \sum_i x_i = \sum_i y_i x_i$$

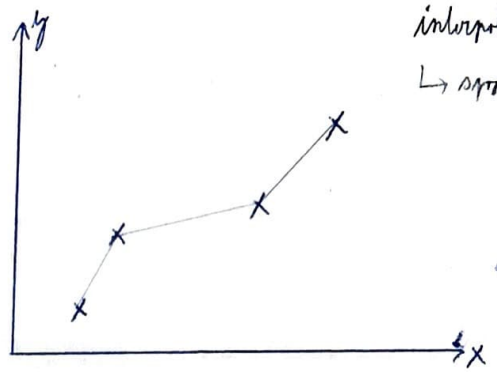
$F'_b(a,b)$

$$a \sum_i x_i + bn = \sum_i y_i$$

26. 2. 2025

Metoda nejmenších čtverců

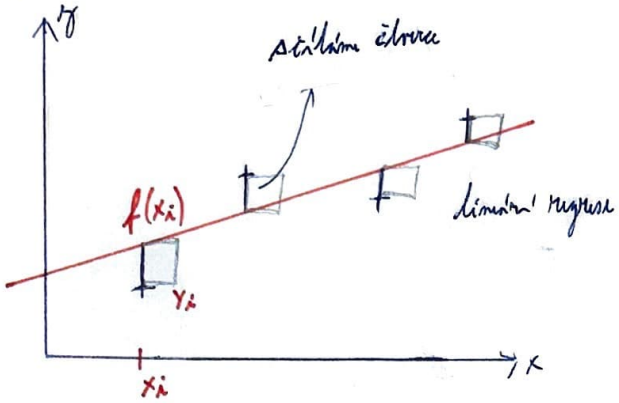
x									
y									

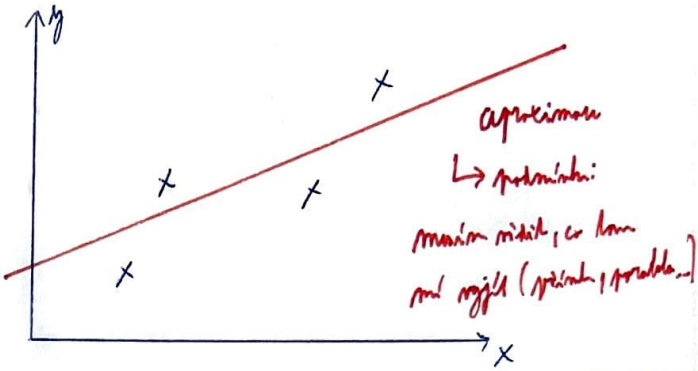


interpolace
↳ spojení bodů

- lineární regrese ⇒ metoda nejmenších čtverců pro přímky

$$y = ax + b$$





$S_{\square} = a \cdot a = a^2$

- i v prostoru se body promítají do čtverců
  ↳ x^3 ⇒ nejde

$$(x^a)' = ax^{a-1}$$

$$[f(g(x))]' = f(g(x)) \cdot g'(x)$$

$$F = \sum_{i=0}^{n} (f(x_i) - y_i)^2$$

$$F(a,b) = \sum_i (ax_i + b - y_i)^2$$

$ax_i + b$ neznámé (x_i, y_i známe z tabulky)

$$F'_a(a,b) = \sum_i 2 \cdot (ax_i + b - y_i) \cdot x_i$$

$$F'_b(a,b) = \sum_i 2 \cdot (ax_i + b - y_i) \cdot 1$$

- derivuju podle a (b = konstanta)
- derivuju podle b (a = konstanta)

$$\sum_i 2 \cdot (ax_i + b - y_i) \cdot x_i = 0$$

$$\sum_i 2 \cdot (ax_i + b - y_i) \cdot 1 = 0$$

$$\left(\sum ax_i^2 + \sum b x - \sum x_i y_i = 0\right)$$

$$\sum ax_i^2 + \sum bx = \sum x_i y_i$$

$$\sum ax_i + \sum b = \sum y_i$$

$\sum b$ běží $\Rightarrow n \cdot b$

```
float konstantniRegrese(const float y[], int n) {
        float sumy = 0.0;

        for (unsigned int i = 0; i < n; ++i) {
        sumy += y[i];
        }

        return sumy / n;
}

void linearniRegrese(const float x[], const float y[], int n, float* a, float* b) {
        float sumx = 0.0;
        float sumx2 = 0.0;
        float sumy = 0.0;
        float sumxy = 0.0;

        for (int i = 0; i < n; ++i) {
        sumx += x[i];
        sumx2 += x[i] * x[i];
        sumy += y[i];
        sumxy += x[i] * y[i];
        }

        Tmatice m;

        m.prvek[0][0] = sumx2;
        m.prvek[0][1] = sumx;
        m.prvek[0][2] = sumxy;
        m.prvek[1][0] = sumx;
        m.prvek[1][1] = n;
        m.prvek[1][2] = sumy;

        vyresSoustavu(&m);

        *a = m.prvek[0][2];
        *b = m.prvek[1][2];
}

int main(void) {
        float a, b;
        linearniRegrese(TAB_X, TAB_Y, TAB_LEN, &a, &b);
        printf("a = %f, b = %f\n", a, b);
        return 0;
}
```