Nelineární nejmenší čtverce

Tomáš Werner, 2011

Mějme m bodů v rovině, $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{R}^2$. Chceme najít kružnici se středem $\mathbf{c} \in \mathbb{R}^2$ a poloměrem r takovou, že součet čtverců vzdáleností bodů od kružnice je nejmenší.

Označte jako $\mathbf{x} = (\mathbf{c}, r) \in \mathbb{R}^3$ vektor parametrů kružnice. Nechť $\text{dist}(\mathbf{a}, \mathbf{x})$ je *vzdálenost se znaménkem (signed distance)* bodu $\mathbf{a}$ od kružnice s parametry $\mathbf{x}$. Tedy $|\text{dist}(\mathbf{a}, \mathbf{x})|$ je kolmá Eukleidovská vzdálenost bodu $\mathbf{a}$ od kružnice, přičemž pro $\mathbf{a}$ vně kružnice bude $\text{dist}(\mathbf{a}, \mathbf{x}) > 0$ a pro $\mathbf{a}$ uvnitř kružnice bude $\text{dist}(\mathbf{a}, \mathbf{x}) < 0$. Chceme minimalizovat funkci

$$f(\mathbf{x}) = \sum_{i=1}^{m} [\text{dist}(\mathbf{a}_i, \mathbf{x})]^2$$

Implementační úkoly

1. Implementujte funkci `d=dist(a,x)`, kde `a` je matice 2×1, `x` je matice 3×1, `d` je skalár.
2. Udělejte jednoduché grafické rozhraní s názvem `enter_points`, které dovolí naklikat body $\mathbf{a}_i$ a uložit je do souboru `points.mat`. Bude fungovat takto:
   - Po spuštění funkce se objeví prázdný obrázek se zafixovaným rozsahem souřadnicových os, přičemž tento rozsah se nebude v průběhu klikání měnit (použijte funkci `axis`).
   - Uživatel pak může naklikat libovolné množství bodů, přičemž končí klávesou 'enter' (použijte funkci `ginput`).
   - Na konci se body uloží do souboru (funkcí `save`).
3. Implementujte optimalizační algoritmus jako grafické rozhraní `x=fit_circle(method)`, kde `method` je řetězec se jménem optimalizační metody a `x` jsou konečné parametry kružnice. Funkcionalita rozhraní:
   - Funkce nejprve nahraje soubor `points.mat` a zobrazí body $\mathbf{a}_i$ jako žluté křížky.
   - Pak uživatel dvojím kliknutím zadá počáteční kružnici, přičemž první kliknutí určí střed $\mathbf{c}_0$ a druhé určí poloměr r_0. Vykreslí se počáteční kružnice s parametry $\mathbf{x}_0 = (\mathbf{c}_0, r_0)$ červenou barvou.
   - Poté se po každém stisknutí klávesy 'space' kružnice překreslí kružnicí s parametry $\mathbf{x}_k$, kde k je index iterace. V obrázku budou vždy body a jediná kružnice. V průběhu toho se do příkazového okna vypisuje aktuální hodnota $f(\mathbf{x}_k)$ (případně další údaje, navrhněte).
   - Po stisknutí klávesy 'enter' grafické rozhraní skončí.

   Takto tedy při postupném mačkání klávesy 'space' uvidíme jednotlivé iterace optimalizačního algoritmu v podobě měnící se kružnice. Implementujte nejméně dva optimalizační algoritmy: čistou (tedy s jednotkovou délkou kroku) Gaussovu-Newtonovu metodu (`method='GN'`) a Levenbergovu-Marquardtovu metodu (`method='LM'`).

Teoretické úkoly

1. Cílem tohoto úkolu je prozkoumat naši účelovou funkci. Jedná se o funkci tří proměnných, tedy ji nemůžeme přímo vizualizovat. Nicméně můžeme zkoumat její dvourozměrné řezy. Nakreslete v Matlabu graf funkce $f(\mathbf{x}) = f(\mathbf{c}, r)$ v závislosti na středu kružnice $\mathbf{c}$ při konstantním poloměru $r = 1$. Tedy kreslíme graf funkce $f(\mathbf{c}, 1)$, což je funkce dvou proměnných (c_1, c_2). K tomu můžeme použít příkazy `mesh, surf, contour, imagesc` apod. Nakreslete graf této funkce pro počet bodů $m = 1$, $m = 2$ a pak pro m rovné nějakému většímu číslu $m \geq 10$. Případně můžete funkci nakreslit pro jednu dimenzi (tj. body jsou skaláry, $\mathbf{c}, \mathbf{a}_j \in \mathbb{R}$). Odpovězte: Je funkce f všude diferencovatelná? Má jedno nebo více lokálních minim?
2. Diskutujte, jaký algoritmus je vhodný na minimalizaci funkce $f(\mathbf{x})$ a proč. Čím více myšlenek a argumentů uvedete, tím lépe.
3. Najděte takovou dvojici počátečních parametrů kružnice $\mathbf{x}_0$, aby algoritmus inicializovaný každými těmito parametry skončil v různých lokálních minimech. Přemýšlejte, jaký soubor bodů je pro to nejvhodnější. Vykreslete body a dvě výsledné kružnice. Diskutujte, které z těchto lokálních minim je lepší.

   Dokážete vypozorovat (neformálně), jak závisí výsledné lokální optimum příp. rychlost konvergence na souboru bodů $\mathbf{a}_i$ a na počáteční kružnici $\mathbf{x}_0$? Divergoval Gaussův-Newtonův algoritmus někdy?

Výstupem bude implementace v Matlabu s implementačními úkoly a písemná zpráva s teoretickými úkoly.

Poznámka: Export obrázku z Matlabu do zprávy je nejlepší dělat příkazem

```
print -depsc jmeno.eps
```

který obrázek exportuje do formátu *Encapsulated Postscript*. Ten pak již převedete např. do PDF. Přímý export z Matlabu do PDF nedoporučujeme, jsou s ním problémy.