

Wojciech Myszka

Laboratorium 3 ver. 28 z drobnymi
modyfikacjami!

2015-11-30 10:07:45 +0100

1. Laboratorium 3: Aproksymacja

1.1. Wstęp

Podstawowym problemem interpolacji jest to, że stara się przeprowadzić funkcję przez wszystkie dane, które posiadamy (węzły). Ma to sens tylko i wyłącznie wtedy, gdy dane są dokładne i niezaburzone. Gdy jest inaczej — powinniśmy myśleć o jakimś ich uśrednieniu.

1.2. Aproksymacja

Aproksymacja to taka metoda „przybliżania” danych, w której zadaną funkcję stara się tak poprowadzić, żeby była **jak najbliżej** posiadanych punktów.

Osobną kwestią jest ustalenie co to znaczy „jak najbliżej”? Jeżeli mamy zestaw danych pomiarowych (par) (x_i, y_i) , $i = 1, 2, \dots, N$, szukamy takiej funkcji $f(x)$ aby:

$$Q = \sum_{i=1}^N (f(x_i) - y_i)^2 \rightarrow \min!$$

Tak postawione zadanie jest bardzo trudne — minimalizacja polega na wybraniu funkcji takiej, żeby... Znacznie prościej jest rozwiązywać zadanie następujące. Niech $f(x) = g(x, a)$ gdzie a jest wektorem parametrów $a = (a_1, a_2, \dots, a_M)$, $M \leq N$

$$Q = \sum_{j=1}^N (g(x_j, a) - y_j)^2 \rightarrow \min!$$

Teraz zadanie optymalizacji jest łatwiejsze — musimy wybrać wektor liczb. Kolejne uproszczenie polega na rozważaniu zadanie liniowego względem parametrów:

$$g(x, a) = \sum_{j=1}^M a_j \varphi_j(x),$$

a zadanie optymalizacji wygląda tak:

$$Q = \sum_{i=1}^N \left(\sum_{j=1}^M a_j \varphi_j(x_i) - y_i \right)^2 \rightarrow \min!$$

Jego rozwiązanie jest stosunkowo proste — wystarczy wyliczyć pochodne cząstkowe $\frac{\partial Q}{\partial a_j}$ i rozwiązać układ równań:

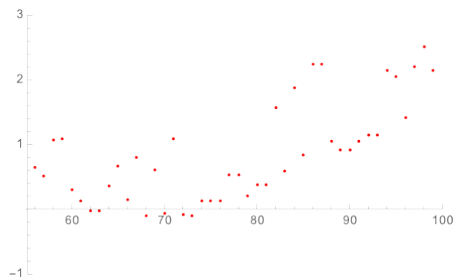
$$\frac{\partial Q}{\partial a_j} = 0; \quad j = 1, 2, \dots, M$$

Zadanie dalej się upraszcza gdy przyjąć, że funkcja $\varphi_j(x) = x^j$.

1.3. Aproksymacja a interpolacja

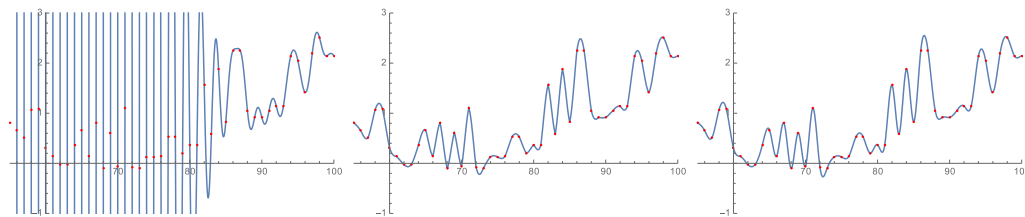
W przypadku zadania interpolacji żądamy, aby funkcja interpolująca przeszła przez wszystkie punkty (węzły interpolacyjne).

Poniżej przedstawiam zestaw punktów (pomiarów temperatury termometrem IR).



Rysunek 1.1. Kilka punktów uzyskanych z pomiarów temperatury

Krzywe interpolacyjne mogą wyglądać tak jak na kolejnym rysunku. Czerwonymi kropkami zaznaczone są węzły interpolacji. Zwracam uwagę, że różnica między interpolacją Hermite'a a krzywymi sklejanymi nie jest specjalnie wielka. Niepokojąco natomiast wyglądają różnice pochodnych — pochodna interpolacji splajnami sześciennymi jest gładka.



Rysunek 1.2. Przykłady interpolacji (od lewej wielomian Newtona, sklejanym Hermita i sklejanym trzeciego stopnia)

1.4. Aproksymacja — Mathematica

Do realizacji aproksymacji wykorzystać można w Mathematici funkcję **Fit**. Jej wywołanie jest następujące:

Fit[*data*, *funcs*, *vars*]

gdzie *data* to zestaw danych (par punktów), *funcs* funkcja lub wektor funkcji którymi przybliżamy. Na przykład: $\{1, x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}, x^{11}, x^{12}\}$, *vars* — zmienna lub zmienne niezależne.

Powyższy zestaw jednomianów w różnych potęgach można łatwo wygenerować automatycznie:

```
funcs = Table [x^i, {i, 0, 10}]
```

```
{1, x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^10}
```

i dalej:

```
funkcja1 = Fit[dane[[All, 2]], funcs, x]
```

w wyniku dostajemy współczynniki wielomianu:

$$\begin{aligned} & -8.003515809018834x^{10} + 1.1265995371896338x^9 \\ & -6.645297813196042x^8 + 2.1252203010076843x^7 \\ & -3.981375997713063x^6 + 4.408253297181539x^5 \\ & -0.0000278354x^4 + 0.00093783x^3 - 0.0165472x^2 + 0.22931x - 1.59072S \end{aligned}$$

Korzysta się z otrzymanej funkcji aproksymacyjnej dosyć łatwo, na przykład:

```
Plot[funkcja1, {x, 0, 288}, PlotStyle -> Blue]
```

albo

```
ff1[x_] = funkcja1;
```

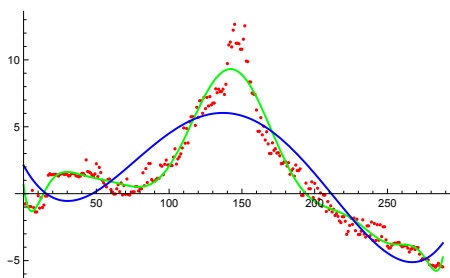
i

```
Plot[ff1[x], {x, 0, 288}]
```

W przypadku bardziej skomplikowanych zadań wykorzystać można również funkcje **FindFit** (funkcja aproksymująca nie musi liniowo zależeć od parametrów), **LinearModelFit** (tylko dla modeli liniowych) i chyba najogólniejszą: **NonlinearModelFit**.

Na poniższej ilustracji przykład aproksymacji dobowych zmian temperatury z termometru IR wielomianami stopnia 12 (zielony) i 4 (niebieski)).

Jak widać — cały problem sprowadza się do wyboru odpowiedniej funkcji aproksymacyjnej.



Rysunek 1.3. Aproksymacja danych wielomianami różnego stopnia: niebieski — 4, zielony — 12

1.5. Zadanie do wykonania

Wybrać jakiś przebieg dobowy i przybliżyć go za pomocą jakiejś sprytniej funkcji (która dobrze będzie oddawała istotę zmienności przebiegu).

Uwagi:

1. Dane otrzymane z pomiarów zawierają bardzo duże wartości współrzędnych x . Może to stanowić problem podczas aproksymacji. Stanowczo więc zalecam przesunięcie czasu do zera (to znaczy pierwszy pomiar dokonywany jest w chwili 0, a następne co 300 sekund). Można to osiągnąć tak:

```
dane = Import[AVERAGE300.dat];
xmin = [Min[dane[[All, 1]]];
dane[[All, 1]] = dane[[All, 1]] - xmin
```

Można też porównać otrzymane wyniki z wynikami aproksymacji tylko wartości y (x przyjmuje wartości 1,2,...):

```
funkcja1 = Fit[dane[[All, 2]], funks, x]
```

2. Jeżeli chodzi o wybór funkcji — sugeruję zacząć od wielomianów. Teoretycznie, im wyższy stopień wielomianu — tym przybliżenie lepsze. Tylko nie wiadomo czy sensowniejsze. Ambitni mogą wymyślić jakąś funkcję nieliniową lub złożyć z kawałków (patrz tutorial).

1.6. Matlab

Możliwości matlaba w zakresie aproksymacji wydają się być mniejsze. Toolbox Curve Fitting zawiera funkcję o nazwie `fit` i wywołaniu:

```
fit(x,y,fitType)
```

x i y to dane wejściowe. jako `fitType` podać należy łańcuch znaków określający rodzaj aproksymacji. Możliwości opisuje dokumentacja. Są tam wielomiany do stopnia 9 i parę innych funkcji.

Najprostsze użycie (korzystające z dostarczonych z matlabem danych przykładowych) wyglądać może tak:

```
load census;
f=fit(cdate,pop,'poly2')
plot(f,cdate,pop)
```

1.7. Aproksymacja a regresja

Wyobraźmy sobie, że mamy n pomiarów x_i jakiegoś parametru i chcemy zaaproksymować je wartością stałą \bar{x} . Chcielibyśmy, aby ta stała była jak najbliższa wszystkim pomiarom. Interesuje nas zatem taki problem:

$$Q = \sum_{i=1}^n (x_i - \bar{x})^2 \rightarrow \min!$$

czyli szukamy takiej wartości \bar{x} , która minimalizuje Q . Policzymy więc pierwszą pochodną $\frac{dQ}{d\bar{x}}$ (będziemy przyrównywać ją do zera):

$$\frac{dQ}{d\bar{x}} = \sum_{i=1}^n 2(x_i - \bar{x}) = 2 \sum_{i=1}^n x_i - 2n\bar{x} = 0$$

zatem

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i.$$

Wzór ten przypomina nam znany ze statystyki wzór na średnią.

Nie od rzeczy będzie wspomnieć, że aproksymacja ma bardzo wiele wspólnego ze znaną ze statystyki regresją. W pewnym sensie jest to to samo (choć nie należy mówić tego głośno) — w przypadku regresji jest cała otoczka związana z probabilistyką (w szczególności zakłada się, że x_i są to obserwacje pewnej **zmiennej losowej** X , a to bardzo silne założenie — mówi ono, o tym, że istnieje rozkład prawdopodobieństwa zmiennej losowej X). Można w takim przypadku pokazać, że wyliczona wartość \bar{x} ma pewne pożądane właściwości — wraz ze wzrostem n , \bar{x} zmierza do wartości średniej rozkładu (jest estymatorem wartości średniej) i, że jest to estymator nieobciążony.

W technice wykorzystuje się średnią do „polepszania” wyników pomiarów. Zakładamy, że wartość a mierzona jest z pewnym addytywnym błędem, czyli: $x_i = a + \zeta_i$; zaburzenia ζ_i są niezależnymi realizacjami obserwacji pewnej zmiennej

losowej Z o średniej 0. Zatem wyliczając $\sum_{i=1}^n x_i$, po dokonaniu odpowiednio wielu pomiarów „odkryć” możemy prawdziwą wartość a .

Podobne interpretacje można zaprezentować również dla innych zadań, w których stosujemy aproksymację.

1.8. Instrukcja w postaci jednego pliku...

... jest również dostępna.