

# Identificarea sistemelor – Laborator 2

## Regresia liniară pentru aproximarea funcțiilor

### Organizare

Acest laborator se rezolvă independent de către fiecare student. Doar în situația în care există mai mulți studenți decât calculatoare, *cu acordul explicit al profesorului de laborator pentru fiecare grup*, studenții se pot grupa câte doi la un calculator.

Soluția constă din cod Matlab. Dezvoltați acest cod într-un singur script Matlab. Dacă aveți nevoie de funcții, acestea pot fi locale în script, vezi funcții locale în scripturi Matlab.

Regulile generale pentru laboratoare, inclusiv termenele limită pentru predarea lor și procedura de validare, sunt descrise pe site și nu vor fi așadar repetate aici; reamintim doar că soluțiile vor fi examinate atent, atât direct de către asistenții de laborator cât și folosind unelte automate, pentru a detecta soluțiile copiate fie de la colegi fie de la generatoare automate de cod cum ar fi ChatGPT. Doar după ce laboratorul a fost validat de către asistent, puteți încărcați soluția dvs. pentru testul final de plagiarism aici:

<https://www.dropbox.com/request/YjHEHfTg711yt9wPPBm9>

Încărcați o singură data, un singur fișier .m, denumit exact după următorul model:

L2\_ROgs\_NumePrenume.m

unde g este grupa și s semigrupa, urmate de numele și prenumele dvs. De exemplu, L2\_RO31\_PopAlex.m. Dacă ați lucrat în grup cf. procedurii de mai sus, încărcați un singur fișier cu ambele nume ale studenților din grup. Fișierele duplicate, nonstandard, denumite în mod necorespunzător sau care corespund unor soluții nevalidate încă de profesorul de laborator nu vor fi luate în considerare.

### Descrierea laboratorului

În acest laborator vom lucra la aproximarea funcțiilor folosind regresia liniară cu aproximatoare polinomiale, vezi secțiunea de *Regresie liniară* din suportul de curs, *Baze matematice*.

Se dă un set de date de intrare-ieșire, unde ieșirea este generată de o funcție necunoscută, neliniară dar statică. Ieșirea este afectată de zgomot, pe care-l vom presupune aditiv, Gaussian, și de medie zero. Funcția are o variabilă de intrare și tot una de ieșire. Va trebui dezvoltat un model pentru această funcție. Un al doilea set de date, generat de aceeași funcție, este furnizat pentru validarea modelului dezvoltat. Cele două seturi sunt furnizate într-un fișier de date MATLAB, conținând câte o structură (tip de date MATLAB) pentru fiecare set. Structura pentru antrenarea modelului este numită `id`, iar cea pentru validare `val`. Fiecare din aceste structuri conține datele de intrare în vectorul `X`, și cele de ieșire corespunzătoare în vectorul `Y`.

Fiecărui student  $i$  se alocă de către profesor un index pentru setul de date. Apoi, studentul descarcă fișierul de date Matlab ce formează baza laboratorului de pe pagina cursului.

Răspundeți următoarelor cerințe:

- Reprezentați grafic datele de identificare, pentru a vă forma o idee despre forma funcției.
- Creați un aproximator polinomial de gradul  $n - 1$ , unde  $n$  este numărul de parametri sau funcții de bază. Valoarea lui  $n$  trebuie să fie ajustabilă în cod. De notat că există un parametru în plus pentru termenul constant din polinom, și de aceea gradul este doar  $n - 1$ . De exemplu, pentru  $n = 4$  polinomul este de gradul 3 și expresia aproximatorului este dată de:

$$\hat{g}(x) = \theta_1 + x\theta_2 + x^2\theta_3 + x^3\theta_4$$

Codul dvs. trebuie să funcționeze pentru orice valoarea a lui  $n$ , configurabilă printr-o variabilă.

- Construiți un sistem de ecuații liniare pentru regresia liniară, folosind datele de identificare. Utilizați reprezentarea matriceală explicată la curs. Rezolvați acest sistem folosind operatorul Matlab de împărțire matriceală la stânga, “\” (sau, ca o alternativă, funcția `linsolve`). Calculați eroarea medie pătratică pe datele de identificare.
- Validați modelul obținut pe setul diferit de date de validare: calculați valorile approximate și din acestea eroarea medie pătratică. Reprezentați grafic funcția aproximată pentru setul de intrări de validare.
- Reglați numărul  $n$  de funcții de bază pentru performanțe cât mai bune, încercând valori de exemplu până la 25. Performanța va fi tot timpul evaluată folosind MSE pe setul diferit de validare, pentru a evita supraantrenarea. Creați un grafic al valorilor MSE în funcție de  $n$ , și găsiți punctul în care MSE este minimală.

Graficele pe care le obțineți vor fi similare celor exemplificate în figura de mai jos (evident, forma funcției și valoarea erorii pot fi diferite pentru setul dvs. de date). Este normal, și de așteptat, ca eroarea pe setul de date de validare să fie mare pentru  $n$  prea mic (subantrenare, aproximatorul nu este suficient de flexibil pentru a modela funcția), și pentru  $n$  prea mare (supraantrenare, aproximatorul este prea flexibil și începe să modeleze zgomot; în acest caz eroarea de validare crește în timp ce eroarea de identificare continuă să scadă).

