

Proiect

Identificarea sistemelor 2018

Organizare

Acest proiect este parte obligatorie a cursului de Identificarea Sistemelor, seria 2, linia de licență Ingineria Sistemelor, Universitatea Tehnică din Cluj-Napoca. Nota la proiect are o pondere de 30% în nota finală a disciplinei. Proiectul va fi efectuat în grupuri de câte doi studenți, și rezolvarea sa necesită aproximativ 20 ore / student, depinzând și de experiența fiecăruia în MATLAB. Fiecare grup va primi seturile sale de date separate. Pentru a le primi, formați cât mai repede grupuri și trimiteți un email profesorului Lucian Bușoniu, la adresa `lucian@busoniu.net`. Menționați numele celor doi membri ai grupului și adresele de email ale ambilor membri. Soluția proiectului va fi descrisă într-un scurt raport, scris în engleză sau în română. Un grup produce un singur raport.

Raportul va fi trimis prin email **în format PDF**, la adresa de mai sus. Nu uitați să menționați numele studenților din grup pe pagina de titlu. Raportul trebuie să includă listing-uri complete ale codului MATLAB dezvoltat (funcții și scripturi). Codul trebuie de asemenea atașat într-o arhivă ZIP aceluiași email în care trimiteți raportul.

Termenul limită pentru raportul și codul final este **13 ianuarie 2019, cel târziu la 24:00**. În plus, o versiune intermediară conținând raportul și codul pentru prima problemă trebuie trimisă până pe **18 noiembrie 2018, 24:00**. În caz că un grup întârzie cu trimiterea soluției, fiecare nouă zi de întârziere începută duce la pierderea a două puncte din nota maximă la proiect (de exemplu, dacă soluția finală este livrată în 15 ianuarie 2019 la ora 00:10AM, nota maximă devine 6 fiindcă a început deja a doua zi de întârziere). **Regulă esențială:** Este strict interzisă copierea de cod, text, sau rezultate de la colegi sau din alte surse cum ar fi website-uri etc. Nu se va tolera nici un fel de material copiat, și încălcarea acestei reguli duce automat și imediat la pierderea dreptului de participare la examen. Fiți așadar extrem de atenți!

Proiectul constă din două probleme. În prima problemă, polinomul este folosit pentru a modela comportamentul unei funcții necunoscute, iar a doua problemă tratează identificarea ARX neliniară a unui sistem dinamic necunoscut.

Partea 1. Modelarea unei funcții necunoscute

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 două variabile de intrare și una de ieșire. Va trebui dezvoltat un model pentru această funcție, folosind un aproximator polinomial. 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 următoarele câmpuri:

- O colecție de coordonate X pentru intrări, unde X este un *cell array* conținând doi vectori, fiecare vector $X\{1\}$, $X\{2\}$ conținând o grilă de n puncte pentru dimensiunea `dim` a intrării.
- O colecție de ieșiri corespunzătoare Y , reprezentată printr-o matrice de dimensiunea $n \times n$, unde $Y(i, j)$ este valoarea funcției f în punctul $(X\{1\}(i), X\{2\}(j))$ (afectată de zgomot).

Aproximatorul polinomial g trebuie să aibă gradul m configurabil. De exemplu, pentru primele câteva valori ale lui m , aproximatorul are forma:

$$m = 1, \quad \hat{g}(x) = [1, x_1, x_2] \cdot \theta = \theta_1 + \theta_2 x_1 + \theta_3 x_2$$

$$m = 2, \quad \hat{g}(x) = [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2] \cdot \theta = \theta_1 + \theta_2 x_1 + \theta_3 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \theta_6 x_1 x_2$$

$$m = 3, \quad \hat{g}(x) = [1, x_1, x_2, x_1^2, x_2^2, x_1^3, x_2^3, x_1 x_2, x_1^2 x_2, x_1 x_2^2] \cdot \theta$$

unde primele două polinoame au fost scrise și explicit pentru claritate.

Odată ce gradul m este ales, antrenarea modelului constă din găsirea vectorului optim de parametri θ pentru care polinomul $g(x)$ se apropie cel mai mult de funcția țintă $f(x)$ pe setul de date de identificare – matematic, vectorul θ care minimizează suma (sau echivalent media) erorilor pătratice. Antrenarea se poate face cu metoda regresiei liniare, ținând cont de faptul că g este liniar în parametri (chiar dacă este neliniar în variabilele de intrare x_1, x_2). Detaliile despre această metodă se găsesc în curs, Partea 3: *Baze matematice*, secțiunile despre regresia liniară. Regresorii sunt în cazul nostru poliomiial puterile lui x_1 și x_2 , de ex. pentru $m = 2$ ei sunt $1, x_1, x_2, x_1^2, x_2^2, x_1 x_2$.

Cerințele pentru prima problemă sunt următoarele. Programați un aproximator polinomial cu grad configurabil. Încercați să antrenați aproximate de diverse grade, pentru a-l obține pe cel mai precis. Validarea trebuie efectuată permanent pe setul diferit de date pentru validare. Raportați eroarea medie pătratică pe ambele seturi de date (identificare și validare), și reprezentați grafic rezultatul dat de aproximator, comparat cu valorile reale ale funcției, pe cele două seturi de date. Discutați rezultatele, inclusiv alegerea gradului și erorile pe cele două seturi de date, ținând cont de discuția din curs despre alegerea modelului și supraantrenarea (en. *overfitting*) în regresie.

Chiar dacă MATLAB furnizează funcții deja implementate pentru regresia liniară cu polinoame, vi se cere să programați dvs. aproximatorul și procedura de regresie. Pentru inspirație puteți examina soluția dvs. pentru laboratorul 4, *Regresia liniară pentru aproximarea funcțiilor*.

Partea 2. ARX neliniar

Pentru a efectua această parte din proiect, veți avea nevoie de bazele despre modele ARX *liniare* din Partea 5 a materialului de curs, *Metode ARX*.

Se dă un set de date măsurat de la un **sistem dinamic** cu o intrare și o ieșire. Ordinul sistemului nu este mai mare de 3, și dinamica poate fi neliniară, în timp ce ieșirea poate fi afectată de zgomot. Vom dezvolta un model de tip cutie neagră pentru acest sistem, folosind o structură ARX neliniară de tip polinomial. Un al doilea set de date, măsurat de la același sistem, este furnizat pentru validarea modelului dezvoltat. Cele două seturi de date sunt furnizate într-un fișier MATLAB, respectiv în variabilele `id` și `val`, ambele obiecte de tip `iddata` din toolbox-ul de identificarea sistemelor. Reamintim că intrarea, ieșirea, și perioada de eșantionare sunt disponibile în câmpurile `u`, `y`, `Ts` ale acestor obiecte. În caz că toolbox-ul nu este instalat, aceleși seturi de date sunt furnizate și în format vectorial, `id_array` și `val_array`, fiecare dintre ele o matrice cu structura: valorile de timp pe prima coloană, intrarea pe a doua, și ieșirea pe ultima coloană.

Considerăm un model cu ordinele na , nb , și întârzierea nk , folosind aceeași convenție ca și funcția MATLAB `arx`. Modelul ARX neliniar are atunci structura:

$$\begin{aligned} \hat{y}(k) &= p(y(k-1), \dots, y(k-na), u(k-nk), u(k-nk-1), \dots, u(k-nk-nb+1)) \\ &= p(d(k)) \end{aligned} \quad (1)$$

unde vectorul de ieșiri și intrări întârziate este notat $d(k) = [y(k-1), \dots, y(k-na), u(k-nk), u(k-nk-1), \dots, u(k-nk-nb+1)]^T$, și p este un polinom de gradul m de aceste variabile.

De exemplu, dacă $na = nb = nk = 1$, atunci $d = [y(k-1), u(k-1)]^T$, și dacă luăm gradul $m = 2$, putem scrie explicit polinomul:

$$y(k) = ay(k-1) + bu(k-1) + cy(k-1)^2 + vu(k-1)^2 + wu(k-1)y(k-1) + z \quad (2)$$

unde a, b, c, v, w, z sunt coeficienți reali și parametrii modelului. De notat că modelul este neliniar, conține pătrate și produse între variabilele întârziate (spre deosebire de ARX standard, care ar conține doar termenii liniari în $y(k-1)$ and $u(k-1)$). Modelul (1) are o proprietate esențială: este liniar în parametri, ceea ce înseamnă că parametrii pot fi găsiți folosind metoda regresiei liniare.

De notat că forma liniară ARX este un caz special al formei generale (1), obținut prin alegerea gradului $m = 1$, ceea ce duce la:

$$\hat{y}(k) = ay(k-1) + bu(k-1) + c$$

și în plus impunând ca termenul liber să fie $c = 0$ (fără această condiție, modelul se numește afin).

Cerințele sunt descrise în cele ce urmează. Programați o funcție care generează un model ARX neliniar de tip polinomial, pentru ordine na, nb , întârzieri nk , și gradul m configurabile. Programați de asemenea procedura de regresie pentru identificarea parametrilor, și utilizarea modelului cu intrări noi. De notat că această utilizare trebuie să se poată face în două moduri:

- Predicție (cu un pas înainte), care folosește valorile reale ale ieșirilor întârziate ale sistemului; în exemplul (2), la pasul k s-ar aplica ecuația (2) folosind variabilele $y(k-1)$ și $u(k-1)$ în partea dreaptă a egalității.
- Simulare, în care ieșirile precedente ale sistemului nu sunt disponibile, și ca atare nu se pot folosi decât ieșirile anterioare ale modelului însuși; în exemplu, $y(k-1)$ ar fi înlocuit cu valoarea simulată precedent $\hat{y}(k-1)$ în partea dreaptă a ecuației (2)

Identificați un astfel de model ARX neliniar folosind setul de date de identificare, și validați-l pe setul de validare. Alegeți atent ordinele modelului și întârzierea, precum și gradul polinomului. Pentru a simplifica procedura de căutare, puteți lua $na = nb$. Raportați eroarea de predicție și eroarea de simulare atât pentru setul de identificare, cât și pentru cel de validare (folosiți eroarea medie pătratică). Includeți un grafic reprezentativ al calității modelului pentru cele două seturi, în simulare și pentru predicție. Discutați rezultatele, incluzând calitatea modelului pe cele două seturi de date.

Prezentarea de mai sus este suficientă pentru implementarea metodei, dar pentru detalii tehnice puteți studia de ex. următoarele articole:

1. H. Peng et al., *RBf-ARX model-based nonlinear system modeling and predictive control with application to a NOx decomposition process*, Control Engineering Practice 12, paginile 191–203, 2007. Modelul este explicat în secțiunile 2.1-2.2, dar folosește în loc de polioname funcții de bază radiale reglabile în poziție și lățime.
2. L. Ljung, *System Identification*, Wiley Encyclopedia of Electrical and Electronics Engineering, 2007. Disponibil sub formă de raport tehnic cu numărul LiTH-ISY-R-2809. Vezi secțiunea 4 pentru modele neliniare, care din nou folosesc funcții de bază.

Programare în MATLAB și remarci adiționale

Faceți un efort pentru a produce un cod MATLAB compact și elegant, evitând folosirea buclor `for`, `while` și a instrucțiunilor `if-then-else` pentru operații care pot fi efectuate mai simplu folosind vectori sau matrici. Indicii în această direcție găsiți căutând conceptul de “vectorizare” (en. *vectorization*) în documentația MATLAB. Nu exagerați însă cu vectorizarea: dacă programul este mai clar sau ușor de înțeles cu bucle sau instrucțiuni `if`, folosiți-le.

Dacă sunteți mai puțin familiari cu programarea în MATLAB, următoarele idei vă pot fi de folos. Scrieți `doc` în linia de comandă pentru accesarea documentației. Un nod util la început este *Getting Started with MATLAB*. Vor fi de folos de asemenea și *Matrices and Arrays*, *Programming Basics*, și *Plotting Basics*.

Este de preferat ca fiecare grup să lucreze la proiect pe laptopul lor, și ca acest laptop să fie același pe toată durata proiectului. Se evită astfel incompatibilitățile între versiunile de MATLAB instalate pe calculatoarele dvs. și cele ale Universității. Dacă acest lucru nu este posibil pentru dvs., contactați profesorul pentru a discuta situația.