

Compararea predictiilor de crestere a fiabilitatii software-ului: retelele neuronale si recalibrare parametrica

Lect. Marian CRISTESCU

Catedra de Informatica Economica, Universitatea "Lucian Blaga" Sibiu

Software-reliability engineering (SRE) stems from the needs of software users. The day-to-day operation of our society is increasingly more dependent on software-based systems and tolerance to failures of such systems is decreasing. Since, software reliability will depend on how software is used, software usage information is an important part of reliability evaluation. This includes information on the environment in which software is used, as well as the information on the actual frequency of usage of different functions (or operations, or features) that the system offers. The usage information is quantified through operational profiles.

Keywords: error, fail, fiability, software engineering, profile, neural network.

Introducere

Odata cu dezvoltarea dimensiunilor si complexitatii software-ului precum si datorita necesitatii obtinerii unei fiabilitati superioare, asociate cu necesitatea unei procesari software eficiente, face ca estimarea fiabilitatii software-ului sa dobândeasca o importanta crescuta. Estimările legate de fiabilitatea software-ului se realizeaza în urmatorul mod:

La începutul perioadei de testare, erorile sunt descoperite mai usor si astfel *TBF* este relativ mic. Odata cu eliminarea erorilor din software, *TBF* creste. Pentru estimarea fiabilitatii software-ului sunt folosite, de obicei, doua metode:

- predictia în termeni variabili;
- predictia ca punct final.

Prima metoda prevede unul sau mai multe puncte plasate pe axa timpului; cea de-a doua estimeaza doar punctul când graficul se stabilizeaza. Înca de la începutul anilor '70 s-au dezvoltat diferite modele pentru estimarea fiabilitatii software-ului, cu numeroase variante. Acestea sunt dependente de timp, fiind modele statistice ale familiilor de curbe care corespund setului de date. Puterea de predictie a modelelor variaza si este adesea limitata la un tip sau altul de date. Un utilizator este nevoit sa treaca printr-un stagiu de încercari pentru a gasi modelul care se potriveste cel mai bine datelor respective. Acest lucru se datoreaza

faptului ca metodele de predictie au fost dezvoltate în medii de programare specifice iar modelele au fost asociate datelor ca familii de curbe specifice.

Puterea de predictie a modelelor parametrice a fost comparata. Când se aplica modelele parametrice, parametrii modelului se estimeaza pornind de la un set de date initial redus. Acest proces (care este numit si model de calibrare), este foarte important pentru o predictie corecta deoarece este necesar sa se gaseasca "parametrii modelului care coordoneaza setul de date" cât mai aproape de realitate, însa la început, setul de date este în mare parte necunoscut. Odata cu realizarea oricarei predictii a seriilor de timp, o mica eroare în estimările initiale mareste drastic eroarea pentru predictia finala. Aceasta se datoreaza faptului ca eroarea în parametrii estimati initial produce o curba diferita fata de cea corespunzatoare valorilor datelor actuale, când acestea sunt observate mai târziu în proces. De exemplu când parametri initiali corespund unei curbe care are ramura initiala ascendenta cu o panta diferita sau cu o curbura diferita, atunci aceste lucruri ar putea determina ca predictia finala sa difere de ceea ce se întâmpla în realitate.

O metoda de a depasi aceasta problema este sa se reestimeze parametrii modelului, recalculând parametrii modelului initial

odata ce noile date devin disponibile. Recalcularea parametrilor se realizeaza pe parcursul testarii si numai atata timp cat sunt gasite greseli. Ca si pentru modelele parametrice, este necesara o faza de testare a erorilor pentru determinarea modelului potrivit. În continuare este necesara o curba corespunzatoare pentru aproape fiecare noua categorie de date, urmata de o actualizare a predictiei. Cu toate ca puterea de predictie este mare procesul este anevoios si necesita calcule aditionale, la momente nedefinite si dificil de inteles de catre neinitiati, facând metoda mai de graba nepopulara pentru scopuri practice. Ca masura a eficacitatii predictiei, se folosesc distantele Kolmogrov.

O metoda alternativa, retelele neuronale, îmbunatatesc modelele parametrice de fiabilitate. Retelele neuronale sunt constructii matematice care pot produce optimizarea graficului prin minimizarea erorilor, iar aplicarea lor necesita un efort relativ scazut. Retelele neuronale necesita doua faze:

- antrenamentul (învatarea);
- simularea.

Faza de învățare corespunde, în acest caz, cu înscrierea în grafic, iar faza de simulare corespunde estimarilor functionale pentru un anumit numar de etape temporale avansate. Ambii algoritmi de antrenament si simulare sunt specifici pentru tipul de retea neuronală care este folosit. Software-ul comercial ofera, de obicei, posibilitatea de a alege rețeaua neuronală cu o varietate de algoritmi de antrenament si simulare ca parte din pachet. Utilizatorul nu este nevoit sa aprofundeze mai mult de atât aspectele teoretice ale rețelilor neuronale. În ciuda existentei unui numar mare de algoritmi pentru rețelele neuronale, doar câțiva sunt compatibili pentru predictia fiabilitatii software-ului.

Experiment de comparare

Pentru a compara predictiile de fiabilitate a software-ului, obtinute folosind metoda recalibrării, cu predictiile realizate cu ajutorul rețelilor neuronale sunt necesare următoarele:

- cele doua tehnici sa fie aplicate aceluasi set de date;
- folosirea aceluasi etalon pentru aprecierea predictiilor.

Pentru a evita repetitia, distanta Kolmogrov este folosita ca etalon obisnuit al predictiei în experimentul de comparare. Pentru a obtine distantele Kolmogrov pentru metodele de predictie bazate pe rețele neuronale, a fost dezvoltat un set sistematic de experimente. În completare, au fost efectuate alte câteva experimente sistematice pentru a testa sensibilitatea fata de alti factori.

Comparatiile au fost realizate pentru predictii de tip "termeni variabili (tendinta)" si "punct final". Predictiile de tip "termeni variabili" sunt importante deoarece reprezinta cazul general si sunt mai dificil de predictat decât cele de tip "pasul urmator". Predictia de tip "tendinta" este realizata prin extrapolarea în viitor a unei secvente de valori, pe baza unui numar de observatii existente. Aceasta include predictii de tip "pasul urmator" si "punct final" pentru a-si atinge scopul; testarea seturilor de date. Este mai semnificativa observarea tendintelor decât evenimentul generat de un singur punct final, deoarece nu se poate sti când este atins punctul final (când cea din urma eroare este detectata si fixata). Scopuri realiste în legatura cu punctele finale sunt: fixarea unei limite pentru o parte din sistemul de programe considerata fara erori sau o limita de timp pentru testare, dar totusi cele mai bune predictii se obtin prin cunoasterea tendintelor.

Pregatirea datelor

În predictia fiabilitatii software-ului, domeniile de interes sunt reprezentate de prevederea celui mai probabil progres al procesului de detectare a erorilor precum si de îndepartarea acestora. Date referitoare al fiabilitate, cum este *TBF*, pot indica o mare variatie a valorilor observate datorita naturii testarii sau esecurilor de punere în practica. Datele sunt adesea grupate în jurul unei valori medii. Datele grupate sunt

dupa aceea combinate si folosite pentru a se înscrie într-un model parametric.

Construirea mediilor si a functiilor cumulative au un efect oarecum uniformizator, lucru care face probabila convergenta curbei cea mai corespunzatoare procesului, dar cu cheltuielile implicate de pierderea datelor.

Detalii tehnice ale modelelor bazate pe rețele neuronale

Compatibilitatea unui tip de retea neuronală este specifica scopului propus. Unele tipuri lucreaza mai bine decât altele într-o situatie data. În [KAR1], se afirma ca rețelele FFN si Elman sunt mai potrivite pentru a fi utilizate în evaluarea fiabilitatii software-ului, ambele tipuri de rețele fiind folosite ca rețele cu întârziere, în cadrul carora:

- intrarea a fost o fereastră a timpilor normalizati de aparitie a defectarilor succesive;
- iesirea a fost numarul cumulat de defecte.

O retea neuronală consta din mai multe nivele ale nodurilor de calcul. Pe scurt, o retea cu întârziere considera ca date de intrare o secventa, de marime fixa, a celor mai recente valori ale functiei si apoi le foloseste pentru a calcula (a predicta) o valoare viitoare. O fereastră de intrare, sau o fereastră simpla, este subsetul $A(t-w), \dots, A(t-1), A(t)$ al seriilor de intrare A , folosite în procesul de antrenare pentru estimarea elementului $A(t+1)$. În timpul fazei de antrenare, parametrii seriilor de timpi sunt determinati printr-un proces de optimizare, utilizând o initializare aleatoare prin rețele neuronale. Mecanismul de antrenament consta în etajarea succesiva a ferestrei prin intermediul partii cunoscute (setul de antrenament) a seriilor si ajustarea parametrilor pâna când rețeaua neuronală reproduce acest lucru într-un mod satisfactor (realizeaza potrivirea cea mai buna). Aceasta functie este apoi folosita în generarea secventei de predictie.

Rețelele Elman lucreaza oarecum diferit. Într-un FFN, curgerea semnalului se face

intrare spre iesire, în timp ce în rețeaua recurentă Elman iesirea unor noduri HL (noduri recurente) este reîntoarsa înapoi la intrari.

Predictia în procesul de executie

Predictiile realizate în timpul executiei sunt împartite în trei grupe mari:

1. Pentru a studia sensibilitatea predictiei fata de marimea ferestrei si HL pentru rețelele FFN iar pentru dublura sa, fata de numarul nodurilor recurente în rețelele Elman;
2. Pentru a studia predictiile de tip punct final;
3. Pentru a studia efectul lungimii perioadei asupra mediei timpilor de executie.

Concluzii:

Compararea empirica a performantelor predictive a doua metode diferite de estimare a fiabilitatii software-ului: "rețele neuronale" si "recalibrarea pentru modelele parametrice", face sa se desprinda urmatoarele concluzii:

- ambele metode pretind ca prezic la fel sau chiar mai bine decât modelele parametrice conventionale care au fost folosite, cu rezultate limitate pâna în prezent;
- fiecare metoda aplica propria ei scala de estimare, ceea ce împiedica o comparatie directa iar pentru a se putea realiza o comparatie trebuie folosita o scala de predictibilitate obisnuita si seturi de date obisnuite;
- rețelele neuronale sunt mai usor de folosit decât metoda recalibrării, si reprezinta o metoda de estimare cel puțin la fel de buna;
- estimările realizate de metoda rețelilor neuronale sunt îmbunătățite, în continuare, prin "combinarea" datelor cu o medie aritmetica activa, în schimb metodele traditionale folosesc mediile grupurilor de date;
- estimările rețelilor neuronale nu depind de modelele cunoscute anterior.

Notatii

AE – eroarea medie (Average Error);

AB – tendinta erorii medii (Average Bias error)

FFN – retea încarcata în prealabil (**F**eed – **F**orward **N**et)

HL – nivel ascuns (**H**idden **L**ayer)

KD – distanta Kolmogrov (**K**olmogrov **D**istance)

NN – retele neuronale artificiale (**A**rtificial **N**eural **N**etworks)

TBF – timpul dintre esecuri (**T**ime **B**etween **F**ailures)

WS – dimensiunea ferestrei (**W**indow **S**ize)

Bibliografie

[ABD1] – Abdel-Ghaly A., Chan P.Y., Littlewood B., ‘*Evaluation of Competing Software Engineering*’, vol. SE-12, pp. 950-967, Sept. 1996

[FAR1] - Farr W.H., Smith D.O., ‘*Statistical Modeling and Estimation of Reliability Functions for Software User’s Guide*’, NAVSWC TR-84-373, Revision

2, Naval Surface Weapon Center, March 1999

[IAN1] - Iannino A., s.a. ‘*Criteria for Software Reliability Model Comparisons*’, in IEEE Transactions on Software Engineering, vol. SE-10, nr.11, 1984, p.687-691

[KAR1] - Karunanithi N., Whitely D., Malaya K., ‘*Prediction of Software Reliability Using Connectionist Models*’, IEEE Transactions on Software Engineering, vol. 18, no.7, July 1992, p.563-574;

[LBL1] - Lu M., Brocklehurst S., and Littlewood B., ‘*Combination of predictions obtained from different software reliability growth models*’, In Proceeding 10th Software Reliability Symposium, June 25-26, 1992;

[LYU10] – Lyu M., Nikora A., ‘*Applying reliability models more effectively*’, IEEE Software, July 1992.