State of art pentru Echipa Sensor middleware

Bogdan Carina
Universitatea de Vest, Informatica aplicata

January 3, 2011

1 Senzor middleware

Middleware este software-ul care oferă o legătiră între două aplicaţii soft diferite şi pasează datele între ele. Middleware-ul oferă posibilitatea de a accesa datele aflate într-o bază de date să fie accesate printr-o alta. Se mai poate spune că middleware este software-ul de integrare de date. ObjectWeb defineşte middleware-ul ca fiind: "The software layer that lies between the operating system and applications on each side of a distributed computing system in a network."[1] În romană : "Stratul software care stă între sistemul de operare şi aplicaţii în fiecare parte a sistemului distribuit a reţelei. " Scopul priectului Sensor Middleware este de a defini şi implementa o arhitectură convenabilă care să răspundă cerinţelor impuse de construirea unei reţele wireless de senzori care să fie capabili să detecteze, să înregistreze şi să prelucreze date care vizează mărimi ale mediului cum ar fi: temperatura, curentul electric, consumul de gaz etc.

2 Arhitectura minimală

Abordarea proiectului nostru va avea în vedere trei probleme importante care constituie si nivelele arhitecturii modelului de snezor middleware:

1. Modelul hardware al senzorului
    - construcţia efectivă tehnică a senzorilor

[1] Krakowiak, Sacha. "What's middleware?".

- topologia senzorilor adică modelel arhitecturare de amplasare a senzorilor în cadrul locuinţei

2. Puntea între hardware şi software

   - dispozitivul de transformare a semnalelor analoage transmise de senzor în semnale binare adică date ce pot fi
   - prelucrate ulterior de un soft specializat
   - stocarea datelor într-o baza de date

3. Modelul software de prelucrare a datelor care presupune

   - agregarea datelor
   - serializarea datelor
   - clusterizarea datelor
   - generare de grafice
   - compararea datelor
   - detectarea punctelor de inflexiune
   - detectarea erorilor cauzate de preluarea datelor

Abordarea middleware pentru reţele de senzori a fost propusă pentru clusterizarea datelor, consumul de energie şi agregarea datelor şi a mecanismului de colectare a acestora. Modelul midleware pentru Reţelele Wireless de Senzori [2] poate fi abordat din două puncte de vedere: abordarea bazată pe bază de date sau abordarea bazată pe agent cea care satisface nevoile proiectului nostru. Reţeaua de senzori este un sistem reactiv care va răspunde schimbarilor de context bazate pe evenimente. Un agent este un cod special care se va plimba de-a lungul nodurilor [3] pentru a executa codul local pe baza evenimentului pe care il va primi. Modelul Sensor Ware [4] foloseşte abordarea bazată pe agenţi. în acest model interpretorul de date rulează pe fiecare nod care va detecta fiecare scimbare venită din exterior. O altă abordare bazată

[2] Wireless Sensor Network abreviat ca WSN

[3] Fiecare senzor care alcătuieţte reţeaua de senzori va fi denumit generic nod

[4] Boulis, C.C. Han, and M. B. Srivastava. Design and Implementation of a Framework for Programmable and Efficient Sensor Networks. In MobiSys 2003, San Francisco, USA, May 2003.

pe modelula gent este Dfuse [5] care este similară cu viziunile Sensor Ware. Aceste modele sunt implementate pe panouri embedded în special pentru probleme cum ar fi snezorii de luminiă şi temperatură.

3 Modele de construcţie a senzorilor

Vom prezenta câteva modele de senzori de temperatură care sunt deja dezvoltate împreună cu avantajele şi dezavantajele pe care le prezintă. în acest sens ne vom îndrepta atenţia spre senzorii de temperatură din seria TE-6100. Porţiunea senzitivă a acestor senzori este reprezentată de elemente de nichel şi silicon a căror rezistenţă variază odată cu schimbarea temperaturii. Modelele de senzori TE-6100 bazate pe fire de nichel sau tip de rezistentă bazate pe silicon conferă o rezistenţă variată pentru un controller. Când temperatura senzorului se modifică, rezistenţa senzorului se modifică în conformitate. Senzorul va arăta p schimbare pozitivă de rezistenţă în concordaţă cu temperatura. Senzorul de silicon se modifică 0.75% per grad Celsius cu o referinţă la rezistenţă de 1035 ohms la 256 grade Celsius. Senzorul de nichel se modifică 5.4 ohms per grad Celsius cu o referinţă la rezistenţă de 1000 ohms la 21 grade Celsius. Modelele asemănătoare dina ceastă categorie modelelor noastre de senzori sunt:

3.1 Modelul TE-6100-1

Acest model (figura de jos) este destinat pentru o paletă universală de aplicaţii, însă acurateţea şi consumul resurselor nu sunt prea avantajoase.

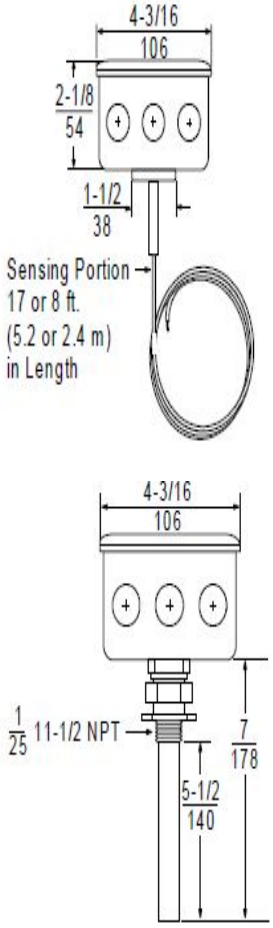
3.2 Modelul TE-6100-3

Acest model (figura de jos) este destinat pentru controlul temperaturii şi memorarea acesteia. Modelul hardware include un element cu două cabluri de nichel, unul care s e intinde de-a lungul a două cilindre concentrice şi altul care este izolat termic şi care este destinat producerii unu lag de timp la schimbări drastice de temperatură.

În modelul nostru

În modelul senzorului nostru schimbările drastice de temperatură nu vor

[5] R. Kumar, M.Wolenetz, B. Agarwalla, J. Shin, P. Hutto, A. Paul, and U. Ramachandran. DFuse: A Framework for Distributed Data Fusion. SenSys, 2003.



fi ignorate de hardware, ci înregistrate ca date, iar ulterior în procesul de prelucrare de date, pe baza unor marje de erori, vor fi excluse din calcul.

4 Cerinţele arhitecturii pentru reţelele de senzori

Programabilitatea

în funcţie de tipul de senzor -staşionar sau mobil reţeaua care facilitează comunicarea între senzorul fizic efectiv si maşina care va prelua datele trebuie să îndeplinească anumite cerinţe. Senzorii staţionari vor comunica prin pro-

tocoale preexistente si caracteristice tipului de senzor ales, pe când senzorii mobili vor avea o reţea de comunicare ad-hoc. Trebuie luat în considerare şi pericolul de deconectare a senzorului de la reţea, lungimea de bandă mică, limitarea aprovizionării cu energie a senzorilor, etc. în conseciinţă reţeaua de senzori trebuie să suporte funcţionalitate de programare, operaţii asincrone şi o arhitectură bazată pe componente. Există asemenea limbaje orientate eveniment care ar putea fi folosite pentru situaţiile în care apar operaţii de deconectare şi anume: NesC [6] şi galsC [7] Însă programarea fiecărui senzor este dificilă, mai ales dacă există reţele de senzori cu numeroase noduri. Noile paradigme de programare, asemenea celor abordate în proiectul nostru, îţi îndreaptă atenţia spre funcţionalităţi specifice la nivel de aplicaţie şi nu la nivel de nod. Soluţiile middleware eficiente pot ascunde complexitatea în ceea ce priveşte configurarea individuală a nodurilor bazată pe capabilităţile arhitecturii software şi hardware.

Adaptabilitatea

Multe aplicaţii de senzori au nevoie să işi poată schimba comportamentul în funcţie de starea mediului în care se află. De exemplu senzorii de lumină vor trebui să funcţioneze în două moduri diferite şi anume în regim de zi şi de noapte. Acest lucru este natural deoarece senzorii nu vor trebui să detecteze lumina zilei ci doar acea lumină venită de la o sursă artificială. în caul în care senzorul nu dispune de memorie suficientă pentru comutare de context, adică de regim de funcţionare, atunci avem două posibilităţi de a rezolva această problemă. Prima ar fi că datele vor trebui trimise unui server pentru lurea acestei decizii şi serverul va fi cel care va dicta senzorului regimul corespunzător de funcţionare. A doua opţiune ar fi ca senzorul să aibe posibilitatea de a hotărî singur ce regim să urmeze urmînd să işi downloadeze singur algoritmul necesar funcţionării care este disponibil pe serverul de decizie. Deoarece am convenit să păstrăm luarea decizilor şi programarea la nivel de aplicaţie şi nu la nivel de nod proiectul nostru va aborda prima variantă.

[6] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D.Culler. The nesC Language: A Holistic Approach to Networked Embedded Systems. In Proceedings of Programming Language Design and Implementation (PLDI), June 2003.

[7] Elaine Cheong, Jie Liu galsC: A Language for Event-Driven Embedded Systems http://ptolemy.eecs.berkeley.edu

Scalabilitate

Majoritatea implementărilor de senzori profită de localizarea spaţială. De exemplu, în cazul senzorilor de detectare a miscării sau a temperaturii, informaţia provenită de la nodurile mai apropiate este evaluată şi pe baza ei decizile ulteriaore vor fi făcute.

Dinamism moştenit

Noduri noi adăugate şi eliminate din cuibul de senzori vor creea o schimbare în topologia snezorilor. De aceea trebuie să avem un mecanism pentru aplicaţie pentrua determina dinamic nodurile şi servicii suport.

Priorităţi de timp real

Reţelele de senzor sunt utilizate pentru amonitoriza fenomene de timp real şi nu alte informaţii stocate. Acest lucru pune problema signării priorităţilor în timpul rulării şi cum să menţinem ordinea naturaşă de timp real. în reţelele de senzori prioritătiile trebuie să fie explicit specificate. Prioritatea unui mesaj depinde de contexul sistemului. De aceea, priorităţile mesajelor trebui să fie asignate la runtime de către middleware şi ar trenui să fie bazate pe context. Managementul fuziunii senzorilor

Aplicaţiile implicate în detectarea miscării folosesc un număr mare de senzori pentrua identifica un obiect particular. Trebuie astfel întreprins un manager pentru fuziunea senzorilor care să suporte fuziunea evenimentelor de nivel jos (cum ar fi temperatura mică şi nivele de lumină) pentru a prelucra evenimente de nivel înalt cum ar fi detectarea obiectelor folosind diverşi algoritmi de fuziune. Acest manager trebuie să fie capabil să selecteze rolul cel mai bun pentru modelele disponibile şi să aloce roluri nodurilor din reţea.

Manager de context

Aplicaţiile cu senzori necesită adaptări în funcţie de schimbările comportamentului şi trebuie să se poată adapta la un context anume în care trebuie să lucreze. Contextul poate fi definit ca un set de stări de schimbare care vor conduce la o stare particulară a reţelei.

5 Achiziţionarea datelor de la senzor

Achiziţionarea datelor este procesul de acumularea semnalelor care măsoară condiţii fizice din lumea reală şi de conversie a reultatelor preluate în valori digitale numerice care ulterior pot fi manipulate de calculator. Sistemele de achiziţie a datelor de la senzori(abreviate DAS sau DAQ), în mod tipic, convertesc forme ondulatorii analogice în valori digitale menite prelucrării computerizate. Componentele unui astfel de sistem sunt:

- Senzorii care convertesc parametrii fizici în semnale electrice
- Circuite de condiţionări ale semnalelor pentru convertirea semnalelor de la sezori într-o formă care va putea fi convertită în valori digitale.
- Convertoare analog spre digital, care vor converti semnalele senzorilor în valori digitale

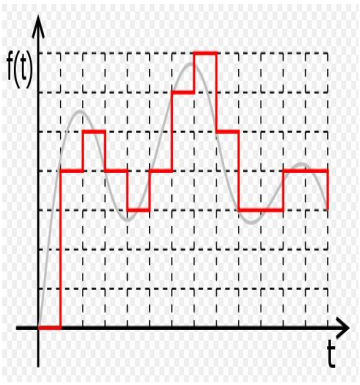
Aplicaţiile de achiziţie de date sunt controlate de programe soft dezvoltate folosind limbaje de programare diferite, în cazul nostru Java. Pentru a accesa si controla data preluată de hardware-ul de achiziţionare se va putea folosi un API care va putea rula pe diverse sisteme de operare.

Achiziţionarea datelor va începe cu fenomenul fizic sau proprietatea fizică care urmează a fi măsurată, cum ar fi temperatura, intensitatea luminii, consumul de energie electrică, consumul de gaz etc. Indiferent de mărimea fizică măsurată, starea fizică care urmează a fi măsurată trebuie să fie mai întâ transformată într-o formă unică care va putea fi preluată de sistemul de achiziţie. Această transformare va fi reaşizată de snezorul efectiv.

Abilitatea de receptare de date a sistemului depinde de tipul de senzor care va detecta variaţia proprietăţii fizice respective. DAQ necesită numeroase tehnici de condiţionare pentru semnalempentru a modifica numeroase semnale electrice în curent electric pentru a fi digitalizate utilizând Convertorul analog spre digital (Analog-to-digital converter (ADC)).

Semnalele pot fi digitale, numile semnale logice, sau analoage. Condiţionarea semnalelor ar putea fi necesară dacă semnalele preluate nu sunt inteligibile pentru hardware-ul DAQ utilizat. Semnalul va trebui fie amplificat, filtrat, fie demodulat.

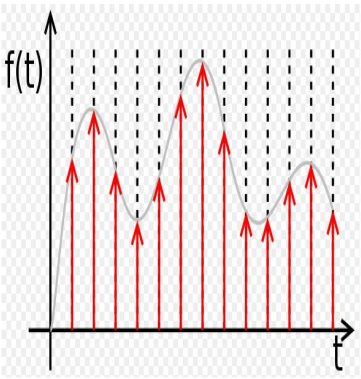
Conversia analog spre digital



Un convertor analog la digital (analog-to-digital converter ADC) este un dispozitiv care va converti o cantitate continuă la un număr digital discret. Tipic ADC-ul este un dispozitiv electronic care va converti un input analog de un anumit voltaj sau curent într-un număr digital proporţional cu magnitudinea care defineşte curentul sau voltajul. Output-ul digital foloseşte o numeroase scheme de codificare diferite. Un ADC poate fi folosit să măsoare valori izolate, însă în cazul nostru se vor folosi semnale dependente de timp şi le va transforma în secvenţe digitale. Rezultatul va fi o cantitate de date măsurată atât în timp, cât şi în valoare. Semnalele pot fi încadrate în numeroase categorii, însă distincţia cea mai comună se realizează între semnalele discrete şi cele continue în funcţie de funcţiile matematice peste care sunt definite. Semnale discrete de timp sunt deseori referite ca şi serii de timp. Semnalele continue de timp sunt referite ca simplu semnale continue.

Un semnal discret sau un semnal discret de timp este o serie de timp care reprezintă o secvenţă de anumite cantităţi. Cu alte cuvinte, reprezintă o serie de timp care este o funcţie definită peste un domeniu de numere discrete. Fiecare valoare a secvenţei s e numeşte eşantion sau probă. Spre deosebire de semnalele continue, un semnal discret nu este o funţie de un argument continuu, chiar dacă se poate ca semnalul să fi fost obţinut prin colectarea unui smenal continuu. Atunci când un semnal discret corespunde unor valori uniforme de timp, va exista o rată de colectare. Rata de colectare nu va fi prezentă în secvenţa de date, deci va trebui să s e atribuie o variabilă separată.

Semnalul digital utilizat va fi un semnal dintr-o serie discretă de timp pentru care amplitudine a si timpul sunt discrete şi care va lua un singur ste de valori discrete şi de asemene a input-ul şi output-ul sunt discrete.



Seria de date utilizată va fi o secvenţă de puncte de date, măsurate la valori de timp succesive şi uniforme. Intervalul nostru de timp între măsurători va fi constant de 6 secunde.

Serile de date au o ordine temporală naturală. Acest lucru le afce distincte de alte probleme de analiză a datelor în care nu există o ordine naturală a observărilor. Un model de serii de date va reflecta în general faptul că observările apropiate ca timp vor fi şi mai apropiate decât observările realizate într-un timp viitor. şi mai mult, modelele serilor de date vor face uz de ordonarea într-o singură direcţie astfel încât valorile pentru o perioadă dată de timp va fi exprimată ca fiind derivată din valori din trecut şi nu valori din viitor.

6 Prelucrări software ale serilor de date

6.1 Clusterizarea datelor provenite de la senzori

Analiza bazată pe clusterizare este procedeul de a găsi grupuri de similarităţi într-o serie de date. Deoarece evenimentele vizate pentru prelucarre sunt prea individuale, ele vor fi grupate în anumite grupuri bazate pe anumite trăsături similare. Un termen necesar aici este *clasificarea* procesul de asignare a unui nou item sau observaţie la locul său potrivit într-un set predefinit de clase de categorii.

Clusterizarea serilor de date cu variabile multiple are ca scop gruparea seturilor de date care au caracteristici similare. Aceste grupuri pot fi analizate ulterior în detaliu pentru a face descoperiri în miezul caracteristicilor comune a fiecărui grup de date.

Clusterizarea folosind factori de similaritate

Factorii de similaritate pot fi folosiţi în locul distanţei Euclidiene(abordare veche) pentru a măsura similaritatea între două seturi de date multivariate X_1 şi X_2. Krzanowski [8] a dezvoltat o metodă pentru măsurarea similarităţii a două seturi de date folosind factorul de similaritate PCA care se va calcula folosind componenta k cea mai mare a fiecărui set de date. Componentele principale (PC) sunt vectorii matricei de covarianţă a setului de date multivalori. Factorul de similaritate S_{PCA} este definit ca :

$$S_{PCA} =^{\Delta} \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{k} cos^2\theta_{ij}$$

unde k este numărul selectat din ambele seturi de date, θ_{ij} este unghiul dintre al i-lea PC a lui X_1 şi al j-mea PC alui X_2. Numărul de PC-uri, k, poate fi ales astfel încât PC-ul k să descrie cel puţin 95% varianţă în fiecare set de date.

Algoritmul propus este cel care foloseşte K mijloace de factori de similaritate

Fiind date Q seturi de date, $X_1, ..., X_q, ..., X_Q$, trebuie să fie clusterizate(încuibate) în K clustere.

1. Fie al j-lea set de date reprezentat în al i-lea set de date numit $X_j^{(i)}$. Se va computa setul de date agregat $X_i (i = 1, ..., K)$, pentru fiecare din cele K clustere ca fiind:

   $$X_i = [(X_1^{(i)})^T ... (X_j^{(i)})^T ... (X_{Q_i}^{(i)})^T]^T$$

   unde Q_i este numărul de seturi de date în X_i. Să se consemneze că $\sum_{i=1}^{K} Q_{i=Q}$.

2. Se va calcula disimilaritatea între setul de date X_q şi fiecare din cele K clustere agregate $X_i, i = 1, ..., K$ ca fiind:

   $$d_{i,q} = 1 - SF_{i,q}$$

[8] Oxford Statistical Science Books, W. Krzanowski, Principle of Multivariate Analisys

unde $SF_{i,q}$ este factorul de similaritate între al q-lea set de date şi al i-lea cluster. Apoi se va lăsa setul de date agregat X_i să fie setul de date de referinţă. Setul de date X_q va fi asignat cluster-ului căruia îi este cel mai puţin nesimilar şi anume clusterul cu valoarea cea mai mică a lui $d_{i,q}$. Se va repeta pasul pentru toate Q clustere.

3. Se va calcula media nesimilarităţilor a fiecărui set de date după formula:

$$J(K) = \frac{1}{Q} \sum_{i=1}^{K} \sum_{X_q \in X_i} d_{i,q}$$

4. Dacă valoarea lui $J(K)$ s-a schimbat dinaintea primei iteraţii atunci ne vom întoarce la pasul 2, altfel stop.

6.2 Compararea seriilor de date pentru sisteme de senzori

În statistici , corelaţia şi dependenţa sunt elemente dintr-o clasă largă de relaţii statistice între două sau mai multe variabile aleatoare sau de date, de valori observate. Există două metode de comparare a serilor de date după cu urmează: Măsura cea mai cunoscută de dependenţă între două cantităţi este coeficientul Pearson sau "de corelaţie Pearson . Acesta se obţine prin împărţirea covariantă dintre cele două variabile la produsul abaterilor standard . Coeficientul de corelaţie a populaţiei ρ, X, Y dintre două variabile aleatoare X şi Y cu valorile aşteptate $\mu\mu X$ şi Y şi abaterile standard σX şi σY, este definit astfel:

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\rho_X \rho_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\rho_X \rho_Y}$$

unde E este valoarea aşteptată, cov înseamnă covarianţă şi corr fiind o notaţie alternativă utilizată pe scară largă pentru corelaţie Pearson. Corelaţia Pearson este definită numai dacă ambele deviaţii standard sunt finite şi ambele dintre ele sunt nenule. Este un corolar al inegalităţii Cauchy-Schwarz deoarece această corelaţie nu poate depăşi 1 în valoare absolută . Coeficientul de corelaţie este simetric: $Corr(X,Y) = Corr(Y,X)$.

Corelaţia Pearson este 1 în cazul unei relaţii liniare perfecte pozitive (în creştere), -1 în cazul unei relaţii liniare perfecte negative (de scadere) şi are

valori cuprinse între -1 şi 1 în toate celelalte cazuri care indică gradul de dependenţă liniară între variabile. Cu cat coeficientul este mai aproape de a fi -1 sau 1, cu atât mai puternică este corelaţia dintre variabile.

Coeficientul Pearson utilizat în compararea seriilor de date pentru sisteme de senzori

Fiind date serii de timp care reprezintă consumul de curent electric se doreşte o comparare low-end a utilizatorilor sau respectiv a claselor de utilizatori. Pentru aceasta ne putem folosi de coeficientul Pearson de dependenţă liniară (deoarece graficele de consum sunt compuneri de funcţii liniare). Pentru a aplica coeficientul Pearson asupra unor seturi de date, acestea trebuie întâi normalizate.

Prin normalizarea seturilor de date se înţelege:

1. Ambele seturi să aibă acelaşi număr de elemente
2. Ambele seturi să fie reprezentate pe aceeaşi scară
3. Eliminarea Zgomotului

Algoritmul pentru generarea coeficientului Pearson

1. Vom avea în vedere două seturi de date S_1 şi S_2, abaterile standard a celor două seturi d_x şi d_y şi medile aritmetice m_1 si m_2 ale valorilor fiecărui set de date se va calcula abaterea standard a setului 1 de date apoi a setului 2 de date conform formulei:

$$\sigma = \sqrt{\frac{1}{N}\sum_{N}^{i=1}(x_i - \mu)^2}$$

   unde σ este abaterea standard, N este mărimea setului de valori, x_i este valoarea considerată, μ este media valorilor setului.

2. Apoi se va calcula media aritmetică a tuturor valorilor setului 1 de date, respectiv setul 2 de date vom lua ca iterator o variabilă i care va începe la 0 şi se va incrementa cu o unitate cât timp variabila i va fi mai mică decât numărul de valori din setul de date la fiecare pas se va calcula raportul dintre diferenţa dintre valoarea extrasă din set şi

media setului respectiv şi deviaţia standard a setului. Acest procedeu se va repeta pentru fiecare set. La fiecare pas vom avea o variabilă rezultat care va păstra radicalul produsului operaţiei calculate pentru setul 1 respectiv setul 2.

3. La sfârşit se va face împărţirea unităţii la prodului sumei rezultatelor de la fiecare pas cu mărimea setului în cauză.

A doua metodă se numeşte **Circle Matching**

Circle matching se bazează pe algoritmi primitivi de comparare de imagini.

Algoritmul utilizat primeşte ca date de intrare 2 grafice de consum ce reprezintă seturile de date de comparat şi returnează o valoare între 0 si 100 ce reprezintă procentajul de similaritate. Algoritmul funcţionează pe următoarele principii:

Fie graficul de consum A roşu şi graficul de consum B albastru Circle matching va detecta pentru fiecare punct al graficului A- puncte albastre în proximitatea sa (proximitate determinată de o constantă alfa, aleasă în funcţie de mărimea setului de date)

Fiecare punct ce indeplineşte condiţia de proximitate incrementează un contor; Algoritmul returnează valoarea: *Contor/NumărTotalPuncte(A+B)*

6.3 Detectarea anomaliilor din fluxul de date a senyorilor de mediu

Unii senzori operează în condiţii grele, iar datele trimise de aceştia către reţelele de comunicare, pot deveni eronate. Senzorii insitu sunt acei senzori care sunt localizati în mediul în care monitorizeaza. Tipurile de erori pot fi cauyate chiar de aceştia, de transmiterea unor date eronate, sau de comportamentul ocazional al sistemului, lucruri care sunt de interes pentru comunităţiile de ştiinţă. Aşadar avem nevoie de o asigurare şi un control în mod automat a calităţii datelor(QA/QC). Acest lucru este foarte util atunci când dorim să facem o analiză mai detaliată a datelor, care la prima vedere pot părea anormale sau pentru care trebuie să executam cu totul alte acţiuni(de exemplu cazul unui dezastru natural). Aplicaţia cere ca datele anormale să fie verificate în timp aproape real, deci trebuie să fie rapid şi efectuat incremental pentru a putea ţine pasul cu rata colectării de date.

Pentru această metodă, se propune un senzor de simulare, ale cărui măsurători vor fi comparate cu măsurătorile senzorului actual. Astfel, vom clasifica o dată ca fiind normală sau anormală pe baza diferenţei dintre anticipările modelului şi măsurătoarea senzorului. Acest studiu dezvoltă o metodă de detecţie a anomaliilor în timp real, care angajează un model de data-driven unidimensional a fluxului de date şi un interval de predicţie (PI), calculat dintr-un istoric recent.

Datele vor fi clasificate ca fiind normale în funcţie de aserţiunea că rezultatul va fi PI sau nu. Această metodă nu cere niciun exemplu preclasificat de date, ci se potriveste unui volum larg de date şi permite o evaluare de date incrementală rapidă, din momentul în care devine disponibilă.

Algoritmul pentru detectarea anomaliilor din serile de date

Acest studiu propune o metodă de redundanţă analitică pentru detectarea anomaliilor, care foloseşte o fereastră mobilă de q măsurători ale senzorilor (sau valorile pe care le aşteptăm de la aceştia):

$$D^t = x_{t-q+1}, ..., x_t$$

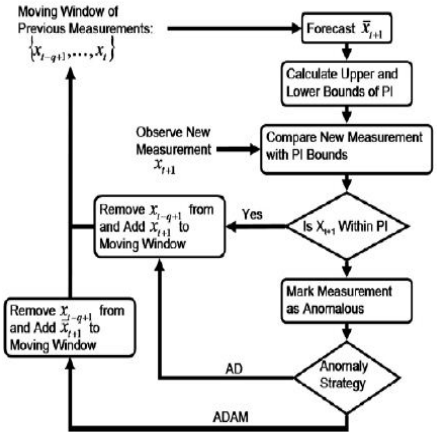
pentru a clasifica următoarele secvenţe de măsurători. Măsurătoarea va fi clasificată ca fiind anormală dacă deviază semnificativ de la valoarea anticipată făcută cu un pas înainte. Apoi metoda umple geamul cu q-ul cel mai recent şi se continuă clasificarea cu următoarea măsurătoare primită de la senzor. Avem următorii paşi începând de la timpul t:

Înaintea valorilor calculate folosind D^t ca dată de intrare. Apoi metoda umple geamul cu q-ul cel mai recent şi se continuă clasificarea cu urmatoarea măsuratoare luată de la senzor. Avem urmatorii paşi (începând de la timpul t):

1. Metoda de anticipare cu un pas înainte ia ca input D^t pentru a anticipa x_{t+1} valoare aşteptată de la senzorul măsurat la timpul $t + 1$;

2. Calculează legăturile din intervalul de anticipare cu probabilitatea p, realizând un rang

3. Comparăm măsurarea de la timpul $t+1$ cu (2) . Dacă datele nu aparţin intervalului rangului de la (2) atunci sunt considerate ca fiind anormale.



4. (a) Sub strategia de detecţie a anomaliilor si a atenuării (ADAM), daca măsuratoarea este clasificată ca fiind anormală modifică D^t ştergând x_{t-q+1} din spatele ferestrei şi adăugând x_{t+1} în faţa ferestrei pentru a crea D^{t+1}

5. (b)Daca măsurătoarea este clasificată ca fiind normală, modifică D^t stergând x_{t-q+1} din spatele ferestrei şi adăugând x_{t+1} în faţa ferestrei pentru a crea D^{t+1}.

6. repeta paşii 1-4.

6.4 Numărarea elementelor distincte dintr-un flux de date

În bazele de date, o problemă fundamentală o reprezintă numărarea elementelor diferite dintr-un tabel. Avem nevoie de un algoritm de optimizare a datelor primite într-un anumit timp şi spaţiu. Acest algoritm este considerat eficient dacă foloseşte foarte puţin spaţiu, dacă are unul sau cât mai puţine cicluri şi dacă procesează fiecare element introdus într-un timp cât mai scurt. Presupunem că avem un şir de elemente $a_1, ...a_n$, n aparţine domeniului $|m| = 1, ..., m$. $F_0 = F_0(a)$(cu alte cuvinte momentul frecvenţei 0) reprezintă numărul elementelor distincte care apar în secvenţă. Algoritmul care aproximează F_0 este considerat eficietn dacă foloseşte mai mulţi $(1/\epsilon, logn, logm)$ biţi de memorie, unde $1 \pm \epsilon$ este un factor în cadrul căruia F_0 trebuie să fie aproximat.

Algorithm	Spaţiu	Timp/element
1, Thm. 1	$O(1/\varepsilon^2 \cdot \log m)$	$\tilde{O}(\log m)$
2, Thm. 2	$\tilde{O}(1/\varepsilon^2 + \log m)$	$\tilde{O}(1/\varepsilon^2 \cdot \log m)$
3, Thm. 3	$\tilde{O}(1/\varepsilon^2 + \log m)$	$\tilde{O}(\log m)$ [amortizat]

Vom prezenta trei algoritmi cu diferite echilibrări timp-spaţiu pentru aproximarea lui F_0. Algoritmii vor fi condiţionaţi de ϵ si $logm$(surprimând dependenţa de n). de exemplu dacă $m < n$ este mult mai avantajos să avem algoritmi ai căror legături depind de $logm$ şi nu de $logn$, iar dacă $m > n$ putem realiza un mic truc utilizând o dispersie a datelor, care reduce descrierea fiecărui element în curs ($O(logn)$). De asemenea presupunem că există $\epsilon_0 < 1$ în aşa fel încât acurateţea parametrului ϵ dat algoritmului este cel mult ϵ_0.

Algoritmul 1
Acest algoritm alege la întâmpalre o funcţie $h : |m| - > [0, 1]$. Apoi aplică funcţia la fiecare element din a şi menţine valoarea $v = min_j^n h(a_j)$. În final estimarea va fi $F_0 = 1/v$. Apoi, mai luăm o funcţie $h : |m| - > [0, 1]$, dar menţinem elementele a_i la $t = O(1/\epsilon^2)$ pe care funcţia h le va evalua la cea mai mică valoare t. Estimarea $F_0 = 1/v$. , v fiind valoarea t cea mai mică.

Teorema 1 Avem un algoritm care pentru oricare ϵ, $\delta > 0$,(ϵ, δ) aproximează F_0 folosind $O(1/\epsilon^2 * logm * log(1/\delta))$ biţi de memorie şi $O(1/\epsilon^2 * logm * log(1/\delta))$ timp de procesare per element.
Demonstraţie: Luăm o pereche independentă de funcţie $h > |m| - > |M|$, unde $M = m^3$ este injectivă pe elementele lui a. Cu probabilitatea cel puţin $1 - 1/m$.

Algoritmul 2 Scopul acestui algoritm este să definească o cantitate care poate fi aproximată într-un model de flux de date şi care poate fi folositp pentru aproximarea lui F_0.

Teorema 2 Avem un algoritm care pentru oricare ϵ, $\delta > 0$,(ϵ, δ) aproximează F_0 folosind $O(1/\epsilon^2 * logm * log(1/\delta))$ biţi de memorie şi $O(1/\epsilon^2 * logm * log(1/\delta))$ timp de procesare per element.
Demonstraţie: Avem $b_1, ..., b_{F_0}$ care denotă F_0 elemente distincte din

fluxul a, iar B denotă setul $b_1, ..., b_{F_0}$. Considerăm o funcţie complet întmplătoare $h > |m| - > |R|$ şi definim $r = Pr_h[h^{-1}(0) \cap B \neq null]$. Mai întâi arătăm că dacă R şi !! sunt constante multiple una pentru cealaltă, atunci aproximând r-ul este un bun mod de a aproxima F_0.

6.5 Aproximare de grafic detectarea punctelor critice

Interpolarea lineară este o metodă de a potrivi graficele generate ca si curbe în mod normal folosind polinoame lineare. Este o metodă des utilizată în matematică, mai ales analiza numerică şi numeroase aplicaţii mai ale sceve bazate pe grafice o folosesc. Interpolarea lineară este o formă simplă de interpolare.

Interpolarea lineară între două puncte date Fie punctele date prin coordonatele (x_0, y_0) şi (x_1, y_1), interpolarea lineară este linia dreaptă trasată între aceste puncte. Pentru o valoare x din intervalul (x_0, x_1), valoare lui y de-a lungul liniei drepte este dată de ecuaţia:

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0}$$

care poate fi derivată geometric ca şi în imagine a de mai jos. Rezolvarea acestei acuaţii în y, care este valoare necunoscută în x, ne dă:

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} = \frac{(x - x_0)y_1 + (x_1 - x)y_0}{x_1 - x_0}$$

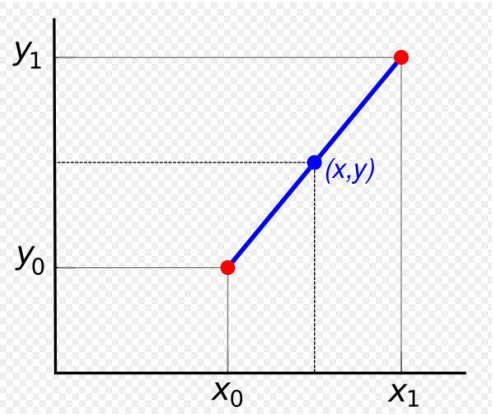
care este formula interpolării lineare din intervalul (x_0, x_1). În afara acestui interval formula este identică cu extrapolarea lineară.

Interpolarea unui set de date

Interpolarea unui set de puncte de date $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$ este definită ca fiind concatenarea interpolării lineare între fiecare pereche de puncte a setului de date. Rezultatul va fi o curbă continuă, cu o derivată discontinuă, totuşi de clasă diferită C^0.

Deseori interpolarea lineară este folosită pentru aproximarea unei valori a unei funcţii f folosind două valori cunoscute a acelei funcţii la fiecare din cele două puncte. Eroarea acestei aproximări este definită ca:

$$R_T = f(x) - p(x)$$



unde p denotă interpolarea lineară polinomială definită mai jos:

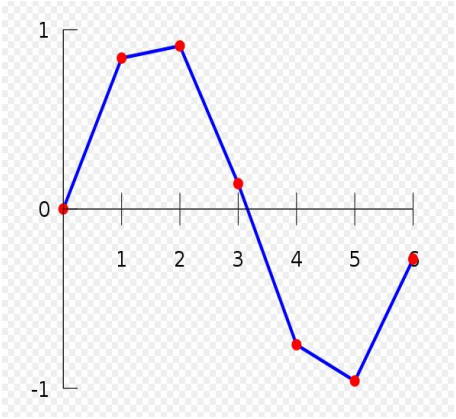
$$p(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

Se poate defini folosind teorema lui Rolle [9] că dacă f are o a doua derivată continuă, eroarea este limitată în:

$$|R_T| \leq \frac{(x_1 - x_0^2)}{8} max_{x_0 \leq x \leq x_1} |f^{||}(x)|$$

După cum s e vede aproximarea dintre cele două puncte pe o funcţie definită devine din ce în ce mai rea cu cea de-a doua derivată a funcţiei acre este aproximată.

[9]Studiu aprofundat la: http://en.wikipedia.org/wiki/Rolletheorem


1
0
-1
1
2
3
4
5
6