

PËRDORIMI I MODELIT TË FSHEHTË TË MARKOVIT NË NJOHJEN E GJUHËS SË FOLUR

*QIRICI O., MELO XH.

Universiteti i Tiranës, Fakulteti i Shkencave të Natyrës, Departamenti i Informatikës

e-mail: olti.qirici@fshn.edu.al

Përmbledhje

Artikulli në vijim paraqet një trajtim përmbledhës të njërës prej metodave më të përdorura në trajnimin e sistemeve të njohjes së gjuhës së folur. Këto sisteme, me zanafillë në një vektor vlerash të gjeneruara nga një proces prodhimi koeficientësh të quajtur MFCC, trajnojnë një sistem peshash, që bëjnë të mundur njohjen e fonomeve të folura. Të gjitha këto algoritme vendosen në zinxhirë dhe ekzekutohen në mënyrë të njëpasnjëshme. Rezultatet e marra nga trajnimi i sistemeve me këtë metodë, ndonëse prodhojnë rezultate shumë të mira krahasuar me metoda të tjera, përsëri lënë për të dëshiruar. Shpesh herë kjo metodë has vështirësi në identifikimin e sinjaleve të ardhura nga pikëtakimi i dy fonemave të ndryshme. Përpjekje të vazhdueshme bëhen në vijim për të gjetur metoda të tjera që të japin rezultate më optimale.

Abstract

The following paper shows a summary of one of the most important training methods for speech recognition systems. These systems, starting with an array generated from a coefficients production process called MFCC, train a weights system which makes possible the recognition of spoken phonemes. All these algorithms are executed in a chain. The results obtained from the systems trained through this method, even though producing very good results in comparison to many other methods, still need further improvements. This method has sometimes problems while trying to detect signals produced from the joining point of two different phonemes. Continuous efforts are being made to obtain a better method which would eventually produce better results.

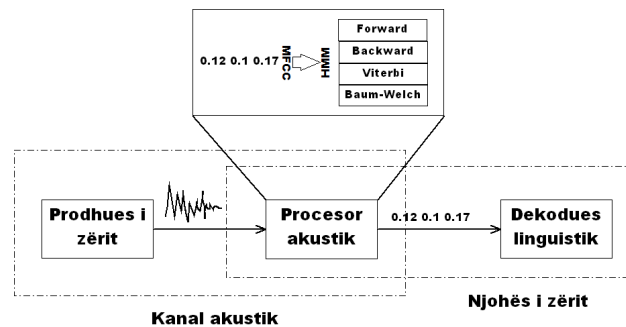
Fjalëkyçe: Modeli i Fshehtë i Markovit, ASR, HMM, algoritmi Viterbi, algoritmi Backward, algoritmi Forward, algoritmi Baum – Welch.

Hyrje

Procesi i njohjes së gjuhës së folur

Procesi i njohjes së gjuhës së folur është një prej proceseve më të vështira për t'u kalkuluar nga ana informatike. Ndryshe nga aparati i dëgjimit të njeriut i cili i njeht mjaft mirë tingujt nëse ata janë shqiptuar qartë dhe nuk ka zhurma në mjedisin përreth, procesi i informatizuar i njohjes së fjalëve të folura mjafton të mbetet një problem i pazgjidhur plotësisht. Edhe gjigandët informatikë industrial apo shkencorë vijnë të provojnë teknika të reja në përmirësim të njohjes së gjuhës së folur. E megjithatë, edhe pse burimet dhe fondet nuk u mungojnë këtyre projekteve rezultatet më të mira janë ende kryesisht të lidhura me identifikimin e fjalëve specifike. Për identifikimin e tog-fjalëshave apo fjalive rezultatet ende janë tepër të dobëta. Shpesh herë mendohet se ende nuk është gjetur metoda optimale për të kryer saktë e

identifikimin e mesazheve. Megjithatë, modeli akustik i zakonshëm i përdorur në njohësit e zërit është modeli i fshehtë i Markovit. Modele të tjera janë të mundura, si ato të bazuara në rrjete artificiale neural apo në ndarjen dinamike të kohës. (Jelinek, 1998). Nga ana tjetër duhet pasur parasysh se modeli akustik është vetëm hallka informatike e trajtimit të zgjidhjes së problemit të njohjes së gjuhës së folur.



Figurë 1: Procesi i njohjes së zërit

Në fakt procesori akustik rrethohet nga njëra anë nga prodhuesi i sinjalit zanor dhe nga ana tjetër nga dekoduesi lingvistik. Detyra konkrete e procesorit akustik është që duke përdorur përshkrimin e modelit akustik të krahasojë një përfaqësi të sinjalit hyrës me një përfaqësi të gjendjeve të regjistruara më parë për të gjetur qasjen e saktë. Në mënyrë që të merret një përfaqësi e saktë e sinjalit hyrës, metoda më e përdorshme do të qe ajo e paraqitjes së cepstrum-it të sinjalit. Pikërisht edhe gjetja e këtij cepstrumi quhet MFCC. Kjo metodë e përdorur fillimisht në aplikime sizmike duket sikur është metoda më e saktë për përpunimin e sinjalit zanor. Pikërisht rezultati i kësaj metode do të analizohet në vijim (rezultati do të qe një vektor me 39 elementë për secilën nga zgjedhjet e njëpasnjëshme që i bëhen sinjalit – semplimi i sinjalit – dhe do të qe paraqitja vektoriale e një subfoni).

Modelet e fshehta të Markovit dhe përdorimi i tyre në modelet akustike

Forma më e thjeshtë e modeleve të Markovit janë modelet e zinxhirëve të Markovit, të cilat mund të përdoren për përshkrimin statistikor të simboleve dhe sekuencave të gjendjeve. Në të ashtuquajturat modelet e fshehta të Markovit (HMM) koncepti i një sekuence gjendjesh, i cili modelohet statistikisht, zgjerohet me daljet specifike të gjendjes së modelit. (Fink, 2003). Nga ky përkufizim i modelit të Markovit nënkuptohet fakti se ky lidhet ngushtë me një sekuencë gjendjesh rrjedhojë të ngjarjeve të rastit (mbi të cilat kemi ngritur të dhëna statistikore). Nga ana tjetër duhet pasur parasysh se informacioni që vjen nga procesuesi i sinjalit zanor është një varg i panjohur që më parë duke pasur parasysh se ai varet në mënyrë të drejtpërdrejtë prej folësit. Sigurisht që folësi kryen jo një proces të thjeshtë rasti por shkronjë pas shkronje ai artikulon një fjalë që gjendet në fjalor ose një variacion mbi fjalën e fjalorit; si dhe fjalë pas fjalë krijon një rrjedhë logjike të lidhur me semantikën e shprehjes. Fjala vjen nën rrjedhën e disa

nëzërave (subfoneve). Secili nga këto semplime nëzërash duhet të tregojë se cilën prej fonemave bazë të gjuhës, e më pas cilën shkronjë – siç ndodh në rastin e gjuhës shqipe që çdo shkronjë përfaqësohet nga një fonemë duhet përzgjedhur. Procesi është relativisht i vështirë dhe jo kaq i drejtpërdrejtë dhe gjetja e një modeli të saktë lidhet me disa algoritme që duhet të veprojnë në grup për të dhënë një rezultat real.

Ajo që në fund duhet të rezultojë në dalje të këtij Modeli të Fshehtë Markovi do të qe [duke mbështetur rregullat e Bajesit, të kishim si rezultat se sekuenca më e mirë e fjalëve është ajo që maksimizon produktin e dy faktorëve, një model gjuhësor të mëparshëm me një ngjarje rasti akustike. (Jurafsky & Martin. 2009).

Shprehja matematikore e asaj që shprehëm më lart do të qe (ku \mathcal{L} është gjithë fjalori me fjalët dhe O është bashkësia e vrojtimit të rastit):

$$\mathcal{W} = \operatorname{argmax}_{W \in \mathcal{L}} P(O|W) P(W)$$

Ky më saktësisht është modeli i kanalit me zhurmë. Le të fillojmë dalëngadalë të shqyrtojmë elementët që marrin pjesë në këtë ekuacion.

$Q = q_1, q_2, \dots, q_n$ – janë N gjendje të ndryshme (Këto N gjendje të ndryshme jo domosdoshmërisht përfaqësojnë një shkronjë. Ato përfaqësojnë një subfon ose më saktë e thënë një pjesëz e sinjalit të përfuar nga një shkronjë. Kjo lidhet me mënyrën e shprehjes së çdo shkronje si dhe në mënyrë të drejtpërdrejtë me ndarjen e njëjtë që i bëhet zinxhirëve të sinjalit që vijnë për fjalët – në frame – ndarje kjo që nuk mund të shmanget pasi sistemi ende nuk e di a ka shkronja apo thjesht tinguj – klithma – në sinjal.)

$$A = \begin{matrix} a_{11}, a_{12}, \dots, a_{1n} \\ a_{21}, a_{22}, \dots, a_{2n} \\ \dots \end{matrix}$$

$a_{n1}, a_{n2}, \dots, a_{nn}$ – janë të gjitha probabilitetet e kalimeve nga një gjendje i në një gjendje j të tilla që shuma e daljeve nga gjendja i drejt të gjitha gjendjeve j do të qe 1 (pra probabiliteti të qe 100%).

$O = o_1, o_2, \dots, o_T$ – T vrojtimit të nxjerra nga një fjalor F . Këto do të qenë më saktë shkronjat e fjalorit fonetik ose fonemat.

$B = b_i(o_t)$ – janë ato që quhen mundësitë e vrojtimit apo probabilitetet e emetimit të cilat janë probabilitetet e emetimit të një vrojtimit o_t nëse jemi në gjendjen i. Tregon sa është mundësia që në një frame të caktuar (në frame-in e i-të) të kemi kapur shkronjën o_j .

q_0, q_F – janë statuset fillestare dhe finale të cilat kanë edhe këto një varg brinjësh që lidhin këto gjendje me secilën prej n gjendjeve (përkatesisht: $a_{01}, a_{02}, \dots, a_{0n}$ dhe $a_{F1}, a_{F2}, \dots, a_{Fn}$).

Ishte e nevojshme të ndalonim pak më gjatë në këtë pjesë pasi pikërisht këtyre akronimeve do tu referohemi vazhdimisht në vijim të këtij punimi. Paragrafët në vijim do të fillojnë të formësojnë dalë ngadalë si do të bëhej zgjedhja e a_{ij} – ve si dhe si do të zgjidhej shkronjë pas shkronje fjala e saktë deri në fund.

Algoritmet për trajnimin e Modelit të Fshehtë të Markovit

Vetë Modeli i Fshehtë i Markovit (HMM) nga sa përmendëm më sipër përmban një numër të madh parametrash të panjohura dhe të papaërcaktuara. Këto parametra korrelohen me njëri tjetrin dhe krijojnë varësi ku gjetja e disa parametrave do të kërkonte gjetjen e disa parametrave të tjerë ku vetë këta të fundit do të kërkonin gjetjen e parametrave të mëparshëm e kështu me rradhë. Në fund të fundit duket sikur jemi në një cikël të mbyllur ku ndërvarësia e bën të pamundur llogaritjen e elementeve. Zgjidhja në këtë rast do të ishte pak më e veçantë dhe do të lidhej më një model trajnimi të elementeve ku një pjesë e të cilave marrin disa vlera fillestare apriori (sigurisht që kanë një sens logjik të qartë) dhe më pas duke zbatuar disa algoritma të programimit dinamik nis dhe bëhet trajnimi i elementëve ku në çdo hap të iteracionit i afrohem gjithnjë e më shumë vlerave të kërkuara për zgjidhje. Klasikisht tre janë këto algoritme dhe quhen: Algoritmi Forward, Algoritmi Viterbi dhe Algoritmi Forward – Backward ose i njohur ndryshe me emrin Algoritmi Baum – Welsh. Këto tre Algoritme do të zgjidhnin këto tre probleme bazë (Jurafsky & Martin 2009).

Problemi 1 (Probabiliteti i ngjarjes): Jepet një HMM $\lambda = (A, B)$ dhe një sekuençë vëzhgimesh O , gjeni probabilitetin e ngjarjes $P(O|\lambda)$.

Problemi 2 (Dekodimi): Pasi jepet një sekuençë O dhe një HMM $\lambda = (A, B)$, gjeni sekuençën më të mirë të fshehur Q .

Problemi 3 (Mësimi): Pasi jepet një sekuençë O dhe një bashkësi gjendjesh në HMM, të mësohen parametrat A dhe B .

Me pak fjalë këto tre probleme do të shtriheshin si: (i) do të na jepte probabilitetin e të qenit në gjendjen j nëse kemi vëzhguar t vlera të mëparshme, (ii) cila do të qe gjendja pasardhëse $i+1$ nëse është kaluar në t gjendjet më të mundshme (shtegu me probabilitet maksimal) dhe nëse kemi vëzhguar t vlera të mëparshme, (iii) cilat do të qenë probabilitetet e kalimit gjendje- i në gjendje- j dhe cilat do të qenë probabilitetet e vëzhgimit të secilës nga shkronjat në t vëzhgimet e mëparshme.

Tani le të shprehim në C++ paraqitjet grafike të këtyre 3 algoritmeve dhe të vëzhgojmë sesi ato ndërthuren deri në gjetjen e rezultateve përfundimtare.

Algoritmi Forward

Algoritmi Forward është algoritmi tipik i cili merret me gjetjen e probabilitetit të ngjarjes nëse paraprakisht janë dhënë dy të dhëna fillestare: një sekuençë vëzhgimesh si dhe probabiliteti i kalimit nga një gjendje në një gjendje tjetër. Në momentin që ne marrim një fjalë në shqyrtim, të themi

fjalën “fjalë”, kjo do të që njësoj sikur të merrnim një sinjal të vazhduar si rrjedhojë e efektit të një vale zanore. Pra “fjalë” nuk do të që gjë tjetër veçse një shfaqje e ngjashme me këtë në figurë:

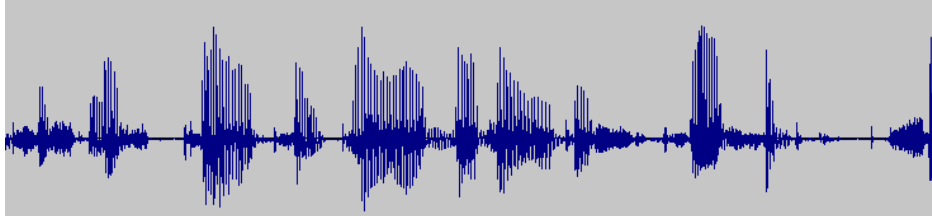
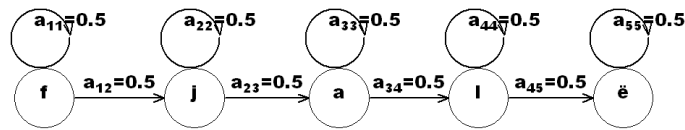


Figure 2: Sinjal zanor (Imazh i marrë nga: <http://www.wirelesscommunication.nl>)

Gjëja e parë që do t’i bëhej këtij sinjali do të që ndarja e tij në nënsegmente të vogla të barabarta mes tyre të cilat eventualisht në rast se do të transformoheshin përmes metodave të cepstrumit do të jepnin një vector me 39 vlera secili. Secili nga këta segmente është i fshehtë për ne sepse ne nuk dimë se ç’fjalë është thënë paraprakisht. Në mënyrë që ne të trajnojmë sistemin parathemi që ky sinjal i përket fjalës “fjalë”. Secili prej këtyre segmenteve i përket një nënzëri (subfon) pra një pjesë e njëres prej shkronjavë që i përket fjalës “fjalë” ose një element i midisshëm midis shkronjave.

Le të themi që ne kemi probabilitetin e kalimit nga një subfon në tjetrin (pra nga një gjendje në tjetrën) në rastin e fjalës së dhënë. Nga ana tjetër kemi një pritshmëri të caktuar që një shkronjë që gjendet në fillim të fjalës të ketë më shumë mundësi të jetë në nënzërat e parë sesa në ata të fundit. Pra nënzërat e parë ka më tepër mundësi që të vrojtohen në vrojtimitet e para. Si rrjedhojë ne mund të masim probabilitetin që të vrojtojmë disa ngjarjeve të caktuara nëse kemi probabilitetet e kalimit mes gjendjeve të njëpasnjëshme si dhe probabilitetet e vrojtimit të një shkronje të caktuar në një moment të caktuar. Fillimisht probabiliteti që të kalojmë nga një gjendje në tjetrën (të themi nga shkronja f në atë j) është e njëjtë me kalimin nga një shkronjë në vetë atë shkronjë (nga shkronja f në f). Më pas këto vlera mund të trajnohen përmes algoritmit Baum-Welch.

Nga ana tjetër mundësia e kalimit të një vlere tek një vlerë e mëparshme është thujse 0 (pra në se jemi në shkronjën j nuk do të kishte kuptim të gjenim në nënzërin e mëpasshëm shkronjën f). Po i njëjti arsyetim do të vrente edhe për dy shkronja më pas e kështu me rradhë (pra nga shkronja f do të që thujse e pamundur të kalohej në shkronjën a). Si rrjedhojë mund të thuhet që variant më adapt i HMM-ve të jetë ai Bakis. Në figurën në vijim mund të shihet ky model HMM-je.



Figurë 3: Bakis HMM e patrajnuar e fjalës "fjalës"

Në këtë shembull u morr si e mirëqenë që kemi matricën B të probabiliteteve që një vrojtim i caktuar t'i përkasë një gjendjeje të caktuar. Po ashtu morëm të mirëqenë edhe probabilitetin e kalimit mes gjendjeve. Ai që do të trajnonte këto probabilitete do të qe përdorimi mbi to i algoritmit Baum-Welch i cili çuditërisht ka pikërisht përdorimin e këtij algoritmi (algoritmit Forward) në trup të tij. Me zgjidhjen e këtij problemi do të merremi ca më vonë.

Në vijim mund të shihet ndërtimi në C++ i një funksioni i cili kryen procesin e forward-it.

```

1  /*
2  * Jane bere T vezhgime dhe kemi nje graf me N gjendje (pra N semplime nga fjala e folur).
3  * Supozohet sikur jane te dhena matricat A dhe B. Ne fakt rezultatet e ketyre matricave dalin nga algoritmi Baum-Welch.
4  * Punoi: Olti Qirici
5  */
6  double forward ( float vezhgime[T], int N, int T)
7  {
8      double probForward[N+2][T];
9
10     //inicializimi
11     for (int s = 0; s < N; s++){
12         probForward[s][0] = A[0][s] * B[s][vezhgime[0]];
13     }
14
15     //rekursioni
16     for(int t = 1; t < T; t++){
17         for(int s = 0; s < N; s++){
18             double paraForward = 0;
19             for(int i = 0; i < N; i++){
20                 paraForward = paraForward + (probForward[i][t-1] * A[i,s] * B[s][vezhgime[t]]);
21             }
22             probForward[s][t] = paraForward;
23         }
24     }
25
26     //perfundimi
27     double paraPerfundimit;
28     for (int s = 0; s < N; s++){
29         paraPerfundimit = paraPerfundimit + probForward[s][T-1] * A[s][N+1];
30     }
31     probForward[N+1][T-1] = paraPerfundimit;
32
33     return probForward[N+1][T-1];
34 }
  
```

Figurë 4: Algoritmi Forward në C++

Në rastin e këtij algoritmi hapi i përgjithshëm do të qe:

$$\alpha_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t-1}) \alpha_{t-1}(j), 1 \leq i \leq N, 1 \leq t < T$$

Algoritmet Viterbi

Algoritmi është mjaft i ngjashëm me algoritmin Forward në disa aspekte. Nga ana algoritmike, siç do të mund të shihet edhe në fund të këtij paragrafi nga implementimi në C++ i kodit për algoritmin Viterbi, veprimet e kryera janë tepër të ngjashme. Por ndërsa algoritmi Forward tenton të gjej shumatoren e probabiliteteve në secilën nga vëzhgimet për të gjetur së fundmi probabilitetin e vëzhgimit të vargut të dhënë nga parametrat e paracaktuara që pqrshkruajnë HMM-në, algoritmi Viterbi, i quajtur edhe

algoritmi i Dekodimit do të gjente shtegun që maksimizon mundësinë e vëzhgimit mes vlerave të HMM-së shprehur këto përmes matricave A dhe B të vlerave të vëzhguara. Pra në rastin e dytë nuk kemi shumatore por gjetje maksimumesh. Përveç gjetjes së kësaj vlere maksimale gjatë ekzekutimit të algoritmit Viterbi ky shteg edhe shënjohehet. Meqë përmendëm këtu vlerat e A-së dhe të B-së duhet pasur parasysh se në këtë algoritëm këto vlera furnizohen përmes zbatimit të kombinuar të algoritmave Forward, Backward dhe Baum-Welch. Për algoritmin Backward ende nuk është folur por ky si algoritëm nuk ka ndonjë ndryshim të madh me Forward-in përveçse ecën në të kundërtën e këtij të fundit.

Në dy pamjet në vijim mund të shihet implementimi në C++ i algoritmit Viterbi. Në këtë algoritëm mund të vihet re paraqitja e instruksioneve të njëpasnjëshme deri në gjetjen e vlerës përfundimtare.

```

1  /*
2  * Jane here T veshgime dhe kemi nje graf me N gjendje (pra N semplime nga fjala e folur).
3  * Supozohet sikur jane te dhena matricat A dhe B. Ne fakt rezultatet e ketyre matricave dalin nga algoritmi Baum-Welch.
4  * Punoi: Olti Qirici
5  */
6  int * VITERBI ( float veshgime[T], int N, int T)
7  {
8      double viterbi[N+2][T];
9      int shteg[N+2][T];
10     //inicializimi
11     for (int s = 0; s < N; s++){
12         viterbi[s,0] = A[0][s] * B[s][veshgime[0]];
13         shteg[s,0] = 0; //Gjendja e fillimit do te jete gjithnje e para
14     }
15
16     //rekursioni
17     for(int t = 1; t < T; t++){
18         for(int s = 0; s < N; s++){
19             double paraViterbiVlere = -1.0;
20             int paraViterbiIndeks = -1;
21             double temp;
22             for(int i = 0; i < N; i++){
23                 temp = viterbi[i][t-1] * A[i][s] * B[s][veshgime[t]];
24                 if(paraViterbiVlere < temp){
25                     paraViterbiVlere = temp;
26                     paraViterbiIndeks = i;
27                 }
28             }
29             viterbi[s][t] = paraViterbiVlere;
30             shteg[s][t] = paraViterbiIndeks;
31         }
32     }

```

Figurë 5.a: Algoritmi Viterbi ne C++

```

33
34     //perfundimi
35     double paraPerfundimitVlere = -1.0;
36     int paraPerfundimitIndeks = -1;
37     double temp;
38     for(int s = 0; s < N; s++){
39         temp = viterbi[s][T-1] * A[s][N+1];
40         if(paraPerfundimitVlere < temp){
41             paraPerfundimitVlere = temp;
42             paraPerfundimitIndeks = s;
43         }
44     }
45
46     viterbi[N+1][T-1] = paraPerfundimitVlere;
47     shteg[N+1][T-1] = paraPerfundimitIndeks;
48
49     return shteg;
50 }

```

Figurë 5.b: Algoritmi Viterbi ne C++

Algoritmi Backward

Mjaft i ngjashëm me algoritmin Forward algoritmi Backward tenton që të gjejë probabilitetin e ngjarjes së vrojttimeve duke u nisur nga fundi i gjendjeve. Pra:

Probabiliteti Backward β është probabiliteti i vrojtimit të vëzhgimeve prej kohës $t+1$ deri në fund, pasi jepet që ne jemi në gjendjen i në kohën t (si dhe pasi jepet automati λ) (Jurafsky & Martin 2009).

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | q_t = i, \lambda)$$

```

1  /*
2  * Jane bere T veshgime dhe kemi nje graf me N gjendje (pra N semplime nga fjala e folur).
3  * Supozohet sikur jane te dhena matricat A dhe B. Ne fakt rezultatet e ketyre matricave dalin nga algoritmi Baum-Welch.
4  * Punoi: Olti Qirici
5  */
6  double backward( float veshgime[T], int N, int T)
7  {
8      double probBackward[N+2][T];
9
10     //inicializimi
11     for (int s = 0; s < N; s++){
12         probBackward[s][T] = A[T][s] * B[s][veshgime[T]];
13     }
14
15     //rekursionimi
16     for(int t = T-1; t >= 0; t--){
17         for(int s = 0; s < N; s++){
18             double paraBackward = 0;
19             for(int i = 0; i < N; i++){
20                 paraBackward = paraBackward + (probBackward[i][t+1] * A[i][s] * B[s][veshgime[t+1]]);
21             }
22             probBackward[s][t] = paraBackward;
23         }
24     }
25
26     //perfundimi
27     double paraPerfundimit;
28     for (int s = 0; s < N; s++){
29         paraPerfundimit = paraPerfundimit + probBackward[s][0] * A[s][0];
30     }
31     probBackward[0][0] = paraPerfundimit;
32
33     return probBackward[0][0];
34 }

```

Figurë 6: Algoritmi Backward

Figura më sipër e ilustron këtë algoritëm. Nëse do të viheshin pranë e pranë këto dy algoritme (Forward dhe Backward) do të vihej re se i vetmi ndryshim mes tyre do të qe ndryshimi i indekseve. Formula e hapit të përgjithshëm të këtij algoritmi është:

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad 1 \leq i \leq N, \quad 1 \leq t < T$$

I vetmi interes që do të na lidhte me këtë algoritëm do të qe përdorimi i mëposhëmi tij nga algoritmi Baum-Welch i përdorur për gjetjen e vlerave a_{ij} dhe $b_i(o_t)$.

Algoritmi Baum-Welch për trajnimin e sistemit

Sejcili nga sistemet që mund të ndërtohen në një gjuhë të caktuar mund të jetë i ndryshëm kundrejt të njëjtit sistem të vendosur në kontekstin e një gjuhe tjetër. Nëse do të merrnim tog tingullin *ey sh e n* në anglisht (ation) në të shkruar, ky tog fjalësh është thujse i pamundur që të gjendet në fjalë të gjuhës shqipe (kjo mbaresë për fjalët e ardhura nga gjuha latine në anglisht është i njëjtë me mbaresën *im* të gjuhës shqipe). Megjithatë parimet bazë jo vetëm që mbeten të njëjta, por madje mënyra e implementimit të tyre është tejet e limituar. Si rrjedhojë ajo që do t'i dallonte mes tyre dy sisteme do të

qe pikërisht trajnimi i veçantë që do të merrej në secilën prej gjuhëve. Nëse do të merrnim si objektiv implementimin e një sistemi për gjuhën shqipe atëherë do të kishim avantazhet dhe disavantazhet në ndërtimin e një sistemi specifik. Nëse në rastin e gjuhës shqipe [parimi themelor i drejtshkrimit Të gjuhës shqipe është parimi fonetik: në përgjithësi fjalët dhe pjesët e tyre të kuptimshme shkruhen ashtu siç shqiptohen në ligjërimin letrar](Akademia e Shkencave të RP të Shqipërisë (1973): Drejtshkrimi i gjuhës shqipe) dhe ky do të qe një avantazh sepse nuk do të kishim problemet e mëdha të kërkimit të fjalëve në fjalor ose që të gjenim shkrimin e saktë të fjalës, nga ana tjetër fjalët janë vështirë të kontrollueshme në fjalor sepse gjuha shqipe ka gramatikë mjaft të vështirë e cila e bën të pamundur identifikimin e një fjale në fjalor pa pasur mekanizma transformimi gramatikor (siç janë FST-të). Gjithashtu një problem në përdorimin e aplikimeve të shpeshta do të lidhej me faktin e ekzistencës së fonemave thuajse unike të gjuhës shqipe (si gj dhe q) të cilat duke mos qenë pjesë e korpusit të gjuhëve gjermanike apo atyre latine nuk janë pjesë e alfabetit standard fonetik ARPAbet.

Algoritmi në vijim, është Algoritmi Baum-Welch, një rast i veçantë i algoritmit Pritshmëri – Maksimizim. Ky është algoritmi më i përdorshëm për trajnimin e modeleve e megjithatë nuk është i vetmi. Rrjetat neurale bëjnë të mundur një modelim alternativ të të njëjtit sistem megjithatë rrjetat neurale janë përdorur shumë rrallë në krahasim me modelin e propozuar në këtë artikull.

Algoritmi Baum-Welch përbëhet nga dy pjesë, pritshmëria dhe maksimizimi të cilat janë paraqitur edhe në figurat në vijim. Megjithatë duke pasur parasysh që algoritmi Baum-Welch bën trajnimin e modelit duke gjeneruar të dhëna gjithmonë e më të sakta të matricave A dhe B të përmendura më parë (siç u përmend edhe në problemin e 3-të të parashtruar më sipër), vetëkuptohet që vlerat e matricave A dhe B duhet të inciliazohen me disa vlera fillestare të cilat do të mund të jepnin një ndihmesë në rezultatet sa më të sakta të këtyre matricave. Në figurat në vijim paraqitet një model i implementimit të algoritmit Baum-Welch. Figura e parë paraqet pjesën e pritshmërisë dhe figura tjetër paraqet pjesën e maksimizimit. Në këtë implementim, pjesa e pritshmërisë lidhet me ekzekutimin e kodit një numër herësh të paracaktuar nga përdoruesi. Shpesh herë vendoset një vlerë gabimi ϵ e cila është diferenca maksimale e lejueshme midis vlerave të njëpasnjëshme të elementeve të matricave i cili përcakton edhe numrin e herëve që ky kod do të ekzekutohet. Matricat A dhe B janë konvergojnë në vlerat përfundimtare dhe nuk ka rrezik që distance ϵ të rritet.

```

1  /*
2  * Jane bere T veshgime dhe kemi nje graf me N gjendje (pra N semplime nga fjala e folur).
3  * Supozohet sikur jane te dhena matricat A dhe B. Ne fakt rezultatet e ketyre matricave dalin nga algoritmi Baum-Welch.
4  * Funcoi: Olti Qirici
5  */
6  void baum-welch ( float veshgime[T], int N, int T, int H, double A[], double B[] ) //A dhe B jane per output
7  {
8      double gamma[T][N];
9      double ksi[T][N][N];
10     double fwd[N+2][T];
11     double bwd[N+2][T];
12     int hapa = 0;
13
14     fwd = forward(float veshgime[T], int N, int T);
15     bwd = forward(float veshgime[T], int N, int T);
16     //inicializimi
17
18     while (hapa < H){
19         //hapi E - Pritshmeria
20         for(int t = 1; t < T; t++){
21             for(int s = 0; s < N; s++){
22                 gamma[t][s] = (fwd[s][t] * bwd[s][t])/fwd[s][t];
23                 for(int i = 0; i < N; i++){
24                     ksi[t][s][i] = (fwd[i][t] * A[s][i] * B[s][t+1] * bwd[s][t+1])/fwd[N][T];
25                 }
26             }
27         }
28     }

```

Figurë 7.a: Algoritmi Baum-Welch (Pritshmëria)

Nga ana tjetër në dy figurat e fundit paraqitet modeli i inicializimit të matricave A dhe B. Në rastin e matricës A inicializimi është tepër i thjeshtë. Duke u nisur nga modeli Bakis i paraqitur më sipër, që është edhe rasti më efektiv sipas të cilit një shkronjë në një varg të paracaktuar karakteresh (duke qenë se modeli përdoret për trajnim – pra njihet nga njëra anë vargu hyrës i vëzhgimeve dhe nga ana tjetër është caktuar fjala apo grupi i fjalëve që po lexohen) ka vetëm dy mundësi me probabilitet të njëjtë vëzhgimi për të vijuar në vëzhgimin e ardhshëm që është ose rivotimi i të njëjtit karakter ose i karakterit pasardhës të fjalës (pra secila me probabilitet 50%). Pikërisht ky shembull shihet edhe në figurën 8. Nga ana tjetër bëhet inicializimi i matricës B duke u nisur nga Modeli i Përzier Gaussian (GMM).

```

29     //hapi M - Maksimizimi
30     for(int i = 0; i < N; i++){
31         S_Ksi_M = 0;
32         for(int j = 0; j < N; j++){
33             S_Ksi_V = 0;
34             for(int t = 0; t < T; t++){
35                 S_Ksi_V += ksi[t][i][j];
36                 S_Ksi_M += ksi[t][i][j];
37             }
38             A[i][j] = S_Ksi_V/S_Ksi_M;
39         }
40     }
41
42     for(int j = 0; j < N; j++){
43         for(int k = 0; k < Simbole; k++){
44             S_Gamma_M = 0;
45             for(int t = 0; t < T; t++){
46                 S_Gamma_M += gamma[j][t];
47                 if(Gamma[j][t] == k){
48                     S_Gamma_Sim += gamma[j][t];
49                 }
50             }
51         }
52     }
53     B[j][k] = S_Gamma_Sim/S_Gamma_M;
54 }
55 }
56 }
57 }
58 }
59 }
60 }

```

Figurë 7.b: Algoritmi Baum-Welch (Maksimizimi)

Modeli i përzier Gaussian paraqitet përmes kësaj formule: $b_j(o_t) = \prod_{d=1}^D \exp(-\frac{(o_{td}-\mu_{jd})^2}{2\sigma_{jd}^2})/\sqrt{(2\pi\sigma_{jd}^2)}$. Me μ është shënuar mesatarja dhe me σ dispersioni i elementeve të vrojtuar, pra të ardhura nga përpunimi i mëparshëm MFCC. Të dyja këto veprime kryhen mbi grupin prej 39 elementesh të ardhura nga

faza e procesorit të sinjalit dhe paraqesin tiparet e ekstraktit të sinjalit zanor të një frame-i të caktuar.

```

47 //inicializimi i vektoreve te tranzitimit mes dy gjendjeve
48 for (int i = 0; i < N; i++)
49     for (int j = 0; j < N; j++)
50         A[i][j] = 0;
51
52 for (int i = 0; i < N; i++){
53     A[i][i] = 0.5;
54     A[i][(i+1)%N] = 0.5;
55 }

```

Figurë 8: Algoritmi Baum-Welch, inicializimi i A ne nje HMM te llojit Bakis

```

19 //inicializimi i vlerave te B, pra i observimet e mundshme
20
21 double my[39];
22 for (int i = 0; i < 39; i++){
23     double S=0;
24     for (int t = 0; t < T; t++){
25         S += cepstrum[t][i];
26     }
27     my[i] = S/39.0;
28 }
29
30 double sq_sigma[39];
31 for (int i = 0; i < 39; i++){
32     double S=0;
33     for (int t = 0; t < T; t++){
34         S += pow(cepstrum[t][i]-my[i],2);
35     }
36     sq_sigma[i] = S/39.0;
37 }
38
39 for (int s = 0; s < N; s++){
40     for (int t = 0; t < T; t++){
41         B[s,t] = 1;
42         for (int i = 0; i < 39; i++){
43             B[s][t] = B[svt] * exp(-pow(cepstrum[t][i]-my[i],2)/(2*sq_sigma[i]))/sqrt(2*PI*sq_sigma[i]));
44         }
45     }

```

Figurë 9: Algoritmi Baum - Welch, inicializimi i B duke perdorur Multivariantin Gaussian

Konkluzione

Në artikullin e mësipërm u paraqit përzgjedhja dhe ndërtimi i njërit prej modeleve akustikë. Si rrjedhojë u konkludua në faktin se modeli i fshehtë i Markovit, me implementimin e paraqitur në artikull është një metodë efektive e trajnimit dhe e identifikimit të gjuhës së folur. Modeli akustik qëndron si një ndërmjetës i procesorit të sinjalit dhe i dekoduesit linguistik. Në sistemin trajtohen grupe 39-she të dhënash të ardhura nga MFCC fillimisht me të dhëna të njohura dhe më pas ku fjalët janë të panjohura. Gjithsesi, shpesh herë në praktikë, përdoren edhe grupe 12-she po të rrjedhura nga procesi i ekstraktimit të MFCC. Panvarësisht nga koha e gjatë e përdorimit të kësaj metode apo edhe të metodave të tjera, në sistemet e njohjes së gjuhës së folur mbetet ende vend për përmirësime të vazhdueshme. Shpesh herë shprehjet e komplikuar matematikore që përdoren në zgjidhjen e këtyre problemeve japin zgjidhje jo të plota të problemit dhe më e mira e mundshme është të jepet një fjala apo shkronja më probabël që të jetë shprehur e cila shpesh herë është e pasaktë.

Literatura

Jelinek F. (1998): Statistical Methods for Speech Recognition

Fink G. (2003): Markov Models for Pattern Recognition

Jurafsky D., Martin J. (2009): Speech and Language Processing, 2nd Edition)

Akademia e Shkencave të RP të Shqipërisë (1973): Drejtshkrimi i gjuhës shqipe

Walker W., Lamere P., Kwok P., Raj B., Singh R., Gouvea A., Wolf P., Woelfel J. (2004): Sphinx-4: A Flexible Open Source Framework for Speech Recognition

Huang X., Acero A., Hon H. W. (2001): Spoken Language Processing: A Guide to Theory, Algorithm and System Development