Fonémaosztályok felügyelet nélküli tanulása
Absztrakt

Gyarmati Ágnes, Vásárhelyi Dániel

{MTA Nyelvtudományi Intézet, ELTE BTK Elméleti nyelvészet doktori program}
1068 Budapest, Benczúr u. 33.
{aagnes, vad}@nytud.hu

Kivonat Írásunkban különféle természetes fonémaosztályok különféle felügyelet nélküli tanulóalgoritmusok általi tanulását mutatjuk be különböző korpuszokon. Ezek az algoritmusok kizárólag az egyes fonémák korpuszon belüli eloszlása alapján, mindenféle fonológiai vagy bármilyen más előzetes ismeret nélkül alkalmasak bizonyos természetes osztályok elkülönítésére.

1. Bevezetés

A hagyományos fonológiai elemzés alapja a megkülönböztető erővel bíró egységekre, a *fonémákra* való szegmentálás. Az egyes fonémákat többnyire négy alapkritérium: a szembenállás, a kiegészítő eloszlás, a fonetikai hasonlóság és a szabad váltakozás elve alapján szokás azonosítani. Erre a műveletre már a strukturalista irányzatokban, a számítógép megjelenése előtt létezett algoritmikus módszer [1]. A generatív fonológia elterjedésével a szamítógépes módszerek háttérbe szorultak, és csak az utóbbi időben kerültek előtérbe.

A fonológiai általánosítások, szabályok vagy megszorítások megfogalmazásában, reprezentációjában alapvető szerepet játszanak a természetes osztályok. Azokban az esetekben, amikor egy természetes osztály egy másik természetes osztályban vett komplementere (azaz egy őt tartalmazó másik természetes osztályból való kivonás eredménye) szintén természetes osztály, értelmezhetjük a két komplementer osztályt, mint a nagy természetes osztály *partícióját* vagy másképpen (kétértékű) *fonológiai jegyet*. Ilyen fonémaosztályok – a teljes igénye nélkül – a magán- és mássalhangzók, az előbbin belül a magyarban az elöl- és hátulképzettek, az utóbbiban a zöngések és zöngétlenek, és így tovább.

Írásunkban három különböző tanulási módszert ismertetünk, melyek közül a legrégebbi Szuhotyin algoritmusa [7], amely az alapján sorolja egy osztályba az elemeket, hogy bizonyos tulajdonságaik mennyire térnek el egymástól. A 2-means klaszterezés egy meghatározott metrika szerinti közelség alapján osztja fel két osztályra a fonémák halmazát [5]. Harmadik módszerünk az összes partícionálás közül választja ki azt, amelyre egy célfüggvény értéke a lehető legnagyobb lesz.

Az algoritmusokat különféle nyelvekből vett korpuszokra teszteltük: magyar, angol, francia, japán, cseh, maori és hawaii. Az algoritmusok bemeneti adatként fonémasorozatokat várnak, ezért a nyelvtől és az adott nyelv helyesírási konvencióitól függően a szokványos írott szövegek általában nem használhatók. Megoldás lehet speciális korpuszok, fonémikus reprezentációban írt korpusz használata, illetve a szöveges korpusz a mi céljainknak megfelelő előfeldolgozása. Az általunk használt angol, japán és francia korpusz fonémikusan átírt szólisták [3], a cseh, maori és hawaii helyesírás jól közelíti a fonémikus reprezentációt. Az írott magyar nyelvű korpusz azonban gondos előfeldolgozást igényel, ezt automatikus eszközökkel oldottuk meg (pl. az azonos értékű fonémák/fonémasorozatok egy közös szimbólummal/szimbólumsorozattal való helyettesítése ($ly \rightarrow j$, $qu \rightarrow kv$). Az eredmény ugyan nem adott tökéletes fonémikus reprezentációt, de a mi céljainknak már megfelelt.

Szuhotyin kifejezetten a mgh/msh-elkülönítés automatizálására tervezte algoritmusát, két alapfeltevésre építve: i, egy szöveg leggyakoribb szegmentuma mindig magánhangzó, ii, a magánhangzók és a mássalhangzók gyakrabban váltakoznak, mint nem. Az algoritmus kezdeti lépésként minden szegmentumot mássalhangzónak címkéz fel, majd a korpuszban található bigramok gyakoriságából kiindulva iteratív lépésekben keresi meg azokat a szegmentumokat, melyek a legnagyobb valószínűséggel magánhangzók, minden lépésben egyet, amíg talál megfelelő jelöltet. Goldsmith és Xanthos szerint az eredmény függ a nyelvtől, továbbá az algoritmus más feladatra nem alkalmazható[4]. Ezzel szemben mi azt találtuk, hogy az eredmény elsősorban az átírás minőségétől függ, nem függ a fonémakészlet felépítésétől, ami nem is meglepő, hiszen a fonémák disztribúcióját veszi alapul. Továbbá az algoritmus más problémák esetében is adhat értelmes eredményt, amennyiben a vizsgálat tárgyában váltakozó tendencia rejlik (pl. egyes nyelvekben a hangsúly). Ekkor természetesen az alapfeltevéseket a megfelelő módon kell módosítani. Az i, tulajdonképpen, csak az egyik osztály szerepének kitüntetésére szolgál, nevezhetnénk az első elem címkéjét *1*-esnek is.

A klaszterezés az egymáshoz való hasonlóság alapján osztályozza az elemeket, így nincs szükség semmilyen előzetes tudásra vagy feltevésre a kategóriákat vagy az osztályozandó elemeket illetően.

Mi a *k-means* algoritmust használtuk, amely egy vektortér pontjait k partícióba sorolja be oly módon, hogy az egyes partíció középpontjaitól való távolságok négyzetösszegét minimalizálja. A vizsgált nyelv fonémakészletének számosságából adódó n dimenziós vektortérben a vektorokat a relatív bigramgyakoriságokból képezzük, esetünkben $k = 2$, mivle két kategóriába kívánjuk sorolni a fonémákat. A klaszterezést a Cluster 3.0 szoftver [2] segítségével végeztük.

Mivel a két klaszter közül egyiknek sem volt kitüntetett szerepe, így fordulhatott elő, hogy a tesztelés közben a 0 jelű clusterbe hol a magánhangzók, hol a mássalhangzók kerültek. A magyar, cseh, francia, maori és hawaii nyelvre tökéletes osztályzást kaptunk ezzel a módszerrel, a japán és az angol fonémák osztályzása kisebb hibát tartalmazott. Az angol esetében viszont elmondhatjuk,

hogy gyakorlatilag ugyanazt az osztályozást kaptuk a hangsúlyt jelölő, illetve a hangsúlyt nem jelölő reprezentációra is. Az egyetlen különbség a klaszterek címkéjében volt.

A k-means clusterezés előnye, hogy szélesebb körben alkalmazható kategorizálásra, az egymással alternáló komplementer osztályok mellett képes olyan természetes osztályok megtanulására is, mint a magyar mássalhangzók elölségi osztálya, amelyekre váltakozás helyett az osztály elemeinek harmóniája a jellemző.

Ugyanezeknek az osztályoknak a tanulására még transzparensebben alkalmazható az a módszer, amelyben minden partícióhoz hozzárendeltünk egy számot, amely értéke nőtt abban az esetben is, ha egy *alternáló*bb[1] és abban az esetben is, ha egy *harmonizáló*bb[2] partíciót választottunk. A legalternálóbb vagy legharmonizálóbb partíció által meghatározott osztályokat tanulta meg az algoritmus.

2. Korpuszok

A cikkben ismertetett módszereket több, különböző fonémarendszerel rendelkező nyelvekre teszteltük. Mivel az algoritmusok bemenete *fonémasorozatok*, a korpuszokat ennek a követelménynek megfelelően kellett kiválasztani, illetve alakítani. Ebből a szempontból a korpuszok háromfélék lehetnek.

- fonémikusan adott
- a fonémikus reprezentációt jól közelítő ortografikus
- a fonémikus reprezentációtól jelentősen eltérő ortografikus

Egy átlagos korpusz az utóbbi két kategória egyikébe tartozik, bár léteznek speciális korpuszok is, melyeket korábban már átírtak fonémikussá, és úgy tették hozzáférhetővé[3].

A tesztelés során használt angol, francia és japán korpuszok a John Goldsmith honlapján szabadon hozzáférhető szólisták. Alapjában véve megőriztük az ott alkalmazott átírási rendszert, azonban az angol korpuszt két változatát is figyelembe vettük, az eredeti, a hangsúlyt is jelölő ArpaBet ábécében, illetve a hangsúlyjelölést elhagyva is.

[1] Azaz a partíció által meghatározott komplementer természetes osztályok elemei egymással többet váltakoztak.

[2] Azaz a partíció által meghatározott komplementer természetes osztályok elemei gyakrabban fordultak elő együtt.

[3] Természetesen fonetikusan átírt szövegeket is használhatunk bemeneti adatként, amennyiben a fonetikai átírás nem sokban különbözik a fonémikustól. Ha ugyanis az átírásban alkalmazott fonetikai rendszer túl finom, akkor elveszhetnek a hangrendszerrel kapcsolatos fontos információk, melyek megragadására a fonológia absztraktabb szintje kínál lehetőséget (pl. a fonémaazonosság kérdése). Erre a későbbiekben még visszatérünk

A maori és a cseh helyesírás gyakorlatilag fonémikus, néhány kisebb átalakítási művelettől eltekintve (pl. a szöveg egységes kisbetűsítése) nem változtattunk rajtuk. A cseh és a maori korpuszt a világhálón szabadon hozzáférhető szövegekből gyűjtve állítottuk össze.

1. táblázat. Extracts from the corpora used

Hungarian
2003-10-11 14:00 Golgota Közösségi Ház.
Bejárat a Hold utca felől.
English (w stress)
BECAME 247 B IH0 K EY1 M
BECAUSE 884 B IH0 K Aó Z
BECKETT 11 B EH1 K IH0 T
BECKONED 8 B EH1 K AH0 N D
BECOME 360 B IH0 K AH1 M
English (w/o stress)
BECAME 247 B IH K EY M
BECAUSE 884 B IH K AO Z
BECKETT 11 B EH K IH T
BECKONED 8 B EH K AX N D
BECOME 360 B IH K AH M
French
abrasif 21 A b r A z i f
abreuver 1 A b r ö v é
abri 39 A b r i
abri-sous-roche 1 A b r i s u r O S
Japanese
bonresuhamu 1 b o n r e s u h a m u
bonryo 1 b o n r y o
bonryuu 1 b o n r y u :
bonsai 1 b o n s a i
Czech
Za další hodinu se vrátil.
Pod paží třímal objemný balík.
Maori
I taku haerenga mai ki te Tari Toko i te
Ora i te marama o Hōngongoi 1993

A legalaposabb előfeldolgozásra a magyar szövegek esetén volt szükség:

– Először eltávolítottuk a szövegből a nem a magyar ábécé betűit jelölő karaktereket, majd kisbetűsítettünk, hogy elkerüljük a kis- és nagybetűk megkülönböztetéséből fakadó fonématöbbszöröződéseket.

– A magyarban, mint tudjuk, néhány fonémát kétjegyű, egy esetben (*dzs*) pedig háromjegyű betűvel jelölünk, ezek a korpusz előfeldolgozása során különös figyelmet igényelnek. A többjegyű betűkkel jelölt hangból képzett geminátákat az első karakter megkettőzésével írjuk le:
  - short: *cs, (dz), dzs, gy, ly, ny, sz, ty, zs*
  - long: *ccs,(ddz),ddzs,ggy,lly,nny,ssz,tty,zzs*

  Ezeket a geminátákat a feldolgozás során felbontottuk, pl. $ssz \rightarrow sz + sz$.
– További átalakítások:
  - $ly = j$
  - $q + u = k + v$
  - $w = v$
  - $x = k + sz$
  - $y = i$
  - $ch = h$

Két különböző magyar korpuszt használtunk az egyik a magyar webkorpusz [6], a másik Jókai Mór *Az arany ember* című regénye volt.

3. Algoritmusok és alkalmazásaik

3.1. Szuhotyin algoritmusa

Szuhotyin algoritmusa [7] egy régi egyszerű algoritmus, mely a magánhangzók és mássalhangzók elkülönítésére szolgál. Egy szöveget beolvasva az abban szereplő szegmentumokat automatikusan két diszjunkt halmazba sorolja be, az algoritmus kimenete ez a két halmaz, a mássalhangzókat, illetve a magánhangzókat tartalmazó halmaz. Ez a felügyelet nélküli gépi tanulási módszer két alapvető feltevésre épül:

– egy átírt szöveg leggyakoribb szegmentuma magánhangzó
– a magánhangzók és mássalhangzók gyakrabban váltakoznak, mint nem

Ebben a fejezetben röviden ismertetjük az algoritmust, majd bemutatjuk tesztelési eredményeinket.

Leírás. Tekintsünk egy nyelvet, mely hangrendszere n fonémából áll: $P = \{p_1, \ldots, p_n\}$, és egy korpuszt az adott nyelvből. A program ezek ismeretében legelőször egy $(n \times n)$ dimenziós négyzetes R mátrixot hoz létre és tölt felt, melynek elemeire: $r_{ij} = r_{ji} = f_{ij} + f_{ji}$, ha $i \neq j$, ahol f_{kl} a $p_k p_l$ bigram előfordulási gyakorisága a korpuszban. Az elemekre vonatkozó szabályt általánosnak véve a főátlóban az egyforma fonémákból álló bigramok gyakoriságának kétszeres értékeinek kellene megjelennie, de az algoritmus működése szempontjából a főátlóbeli értékek definíció szerint 0-k. R=

$$\begin{pmatrix} 0 & f_{12}+f_{21} & \dots & f_{1n}+f_{n1} \\ f_{21}+f_{12} & 0 & \dots & f_{2n}+f_{n2} \\ \vdots & & \ddots & \vdots \\ f_{n1}+f_{1n} & \dots & & 0 \end{pmatrix} \tag{1}$$

A fonémák fel vannak címkézve, kezdetben minden fonémá címkéje *mássalhangzó*. Az algoritmus ezután egy iteratív fázisba lép, a ciklusmag minden egyes lefutásakor az R mátrix adataiból kiindulva megkeresi és átcímkézi azt a fonémát, mely a legnagyobb valószínűséggel valójában nem is mássalhangzó, hanem magánhangzó. A további lépésekben az „új magánhangzónak"az R mátrixbeli összes adatát figyelmen kívül kell majd hagyni.

A második alapfeltevés alapján az a fonéma lehet jelölt a magánhangzóságra, mely jóval gyakrabban előz meg vagy követ mássalhangzókat, mint magánhangzókat, így mindenképp pozitívnak kell lennie annak azon gyakoriságok különbségének, melyek megmutatják, hogy a korpuszban hányszor állt az adott fonéma mássalhangzó, illetve magánhangzó közvetlen környezetében. Ez a különbség iteratívan hozzá van rendelve minden egyes p_i fonémához a megfelelő $v(p_i)$ értékben:

$$v(p_i) = \sum_{1 \le j \le n, j \neq k_v} r_{ij} - \sum_{1 \le j \le n, j = k_v} r_{ij}$$

ahol k_v azon fonémák indexei, melyek már korábban új, magánhangzós címkét kaptak. Minél magasabb ez a $v(p_i)$ érték, annál valószínűbb, hogy p_i magánhangzó. Az algoritmus mohó, az átcímkézésre mindig a legmagasabb $v(p_i)$ értékkel rendelkező fonémát választja ki (illetve közülük az egyiket, ha több ilyen is van). Amikor a $v(p_i)$ értékek között nincs pozitív, az algoritmus leáll, és visszaadja a felcímkézett fonémák listáját.

Alkalmazások.

3.2. Eredmények

Goldsmith és Xanthos [4], miután Szuhotyin algoritmusát három különböző nyelvre alkalmazták (angol, francia, finn), megállapitják, hogy habár az algoritmus maga nyelvfüggetlen, pontossága erősen függ a bemenő adatoktól. Az algoritmust mi magyar, angol, cseh, maori nyelveken teszteltük.

Az eredmény a magyarra tökéletes volt, azaz minden általunk jól ismert magánhangzó magánhangzó lett, és csak a valódi mássalhangzók maradtak a mássalhangzók között. A jó eredmények arra ösztönöztek minket, hogy megvizsgáljuk Goldsmith és Xanthos azon kijelentését, mely szerint az adatok erősen befolyásolják az algoritmus pontosságát. Angol nyelvű korpuszra is alkalmaztuk az algoritmust, Goldsmithékéhez hasonló eredménnyel.

2. táblázat. Szuhotyin algoritmusa a Magyar webcorpusra

Cluster	Fonémák
Magánhangzók	a,á,e,é,i,í,o,ó,ö,ő,u,ú,ü,ű
Mássalhangzók	b,c,cs,d,dz,dzs,f,g,gy,h,j,k,l,m,n,ny,p,r,s,sz,t,ty,v,z,zs

3. táblázat. Szuhotyin algoritmusa angolra (hangsúlyjelöléssel)

Cluster	Fonémák
Mgh	AH0,R,S,IH0,L,ER0,N,M,EY1,WHH,Y,ER1,AÉ,AY0,EY0,OW0,DH
Msh	AA0,AÁ,AE0,AH1,AO0,AÓ,AW0,AW1,AY1,B,CH,D EH0,EH1,F,G,IH1,IY0,IY1,JH,K,NG,OW1,OY0 OY1,P,SH,T,TH,UH0,UH1,UW0,UW1,V,Z,ZH

Amint azt Goldsmith és Xanthos is megjegyzik, a fonetikai átírás félrevezeti az algoritmust azáltal, hogy olyan szimbólumok is megkülönböztetendők, melyek ugyanazt a fonémikus tartalmat jelölik, de eltérő a hangsúlycímkéjük (0 vagy 1 (hangsúlytalan, illetve hangsúlyos)). A hangsúlyjelölések eltávolítása az „igazi" fonémák kisebb halmazához vezet, egyúttal jelentősen javítja az algoritmus által elérhető eredményeket is. Az egyetlen hibája, hogy az R-t tévesen magánhangzónak osztályozza.

Az algoritmus ugyanazokat a lépéseket végzi, és a korpusz is ugyanaz. Az eredményt tehát nem a korpusz változtatta meg, hanem a szimbólumok halmaza, azaz az ábécé. Mivel az algoritmus célja eredetileg is az volt, hogy tanulja meg a mássalhangzók és magánhangzók kategóriáját, ha adottak a nyelv *fonémái*, az ArpaBet hangsúlyjelölőit egyszerűen figyelmen kívül kell hagyni, hiszen a hangsúly az angolban nem fonémikus.

4. táblázat. Szuhotyin algoritmusa angolra (hangsúlyjelölés nélkül)

Cluster	Fonémák
Mgh	AA,AE,AH,AO,AW,AX,AY,EH,ER,EY,IH,IY,OW,OY,R,UH,UW
Msh	B,CH,D,DH,F,G,HH,JH,K,L,M,N,NG P, S, SH, T, TH, V, W, Y, Z, ZH

Az *r* szokatlan viselkedésére (angolban és franciában is az egyetlen mássalhangzó, mely átkerült a magánhangzók közé) többféle magyarázat lehetséges. Az egyik szerint az *r* a különböző nyelvekben különböző mértékben konszonantális (ld. a fonetikai megvalósítas változatosságát). Ez további vizsgálatokat igényel. Meg kell jegyezni azonban, hogy a cikkben használt korpusz a sztenderd ameri-

kai normát követi, rotikus dialektus átirata. Az *r* gyakran szótagképző, valamit mássalhangzó előtt is megmarad (nemrotikus dialektusokban a prekonszonantális *r*-ek nincsenek[4].

Az algoritmust cseh nyelvű korpuszon is teszteltük. A cseh nyelv egyik jellegzetessége, hogy az *l* és az *r* lehetnek szótagképzők. Várhatnánk, hogy az *l*, de

5. táblázat. Szuhotyin algoritmusa korpuszra

Cluster	Fonémák
Magánhangzók	ý,ó,a,é,y,o,ě,á,e,í,u,ú,i,u
Mássalhangzók	ň,k,v,w,č,ž,l,x,m,b,c,n,z d,ch,p,f,ř,g,r,š,ď,s,h,t,j

főleg az *r* átkerüljön a magánhangzók közé, de ez mégsem történik meg. A szillabikus és a nem-szillabikus *r* fonémikusan azonosak, tehát az algoritmus nem tesz különbséget közöttük. Az algoritmus az osztályozást a fonémák disztribúciójára alapozva végzi, így az a tény a döntő, hogy mely helyzetben, milyen környezetben a legnagyobb a gyakorisága. Az *r* gyakrabban áll tiszta mássalhangzós pozícióban, szótagmagban, így nem kerül a magánhangzók közé.

A maori fonémarendszer az eddig tárgyalt nyelvekétől eltér, mindössze kilenc magánhangzóból és tíz mássalhangzóból áll, így méretében és arányaiban is más. Nyelvfüggetlen tanuló algoritmusunk tökéletes eredményt ad. Szuhotyin

6. táblázat. Szuhotyin algoritmusa maori korpuszra

Cluster	Fonémák
Magánhangzók	a,ē,ō,o,e,ī,ā,i,ū,u
Mássalhangzók	k,w,m,n,ng,p,wh,r,h,t

ezt az algoritmusát kifejezetten a magánhangzók és mássalhangzók automatikus megkülönböztetésére tervezte. Goldsmith és Xanthos ezt egyik hátrányának is tartják, az algoritmus eredeti célja annyira speciális, hogy más feladatra nem alkalmazható. Vizsgáljuk meg azonban a két kezdeti alapfeltevést.

– egy átírt szöveg leggyakoribb szegmentuma magánhangzó
– a magánhangzók és mássalhangzók gyakrabban váltakoznak, mint nem

[4] Ez elméletfüggő, hogy a prekonszonantális, illetve szünet előtti helyzetben egyáltalán nincs-e *r*, vagy csak a felszínen nem jelenik meg

Az első tulajdonképpen azt a célt szolgálja, hogy az algoritmus ki tudja választani a megfelelő címkét a két halmazból. Ez azonban csak egy kitüntetett szerep megjelölése, de az algoritmus lényegi működését nem érinti, fel lehet címkézni a *magánhangzók* helyett 1-essel is.

A második feltétel pedig csak annyit vár el, hogy a szegmentumok valamilyen szabály(osság) szerint váltakozzanak (jelen esetben a tendencia a magánhangzók és a mássalhangzók váltakozása). Ha eltekintünk a konkrét szerepektől, és csak a szabályossagra fordítjuk figyelmünket, felfedezhetjük, hogy bármely két osztály elemeinek besorolására használható, amennyiben a két osztály elemei váltakozást mutatnak.

Ilyen váltakozást mutat pl. az angolban a hangsúlykiosztás: nem állhat két hangsúlyos szótag egymás mellett. Ebben az esetben az ArpaBet hangsúlyjelölése már nem irreleváns információ. Valóban, ha korpusznak az eredeti korpusz magánhangzóból álló sorozatot tekintjük, akkor az algoritmus eredményeként az ArpaBet 0-s, illetve 1-es hangsúlyjelű elemeit szetválogatva. Mivel a schwa a legtöbbször előforduló magánhangzó a szövegben, így az 1-es kategóriába a hangsúlytalanok kerülnek. Érdekesebb kérdés, hogy mit ad az algoritmus, ha a

7. táblázat. Szuhotyin algoritmusa az angol magánhangzókra

Cluster	Fonémák
Mgh 1-es kategória (hangsúlytalan)	AA0,AE0,AH0,AO0,AW0 AY0,EH0,ER0,EY0,IH0 IY0,OW0,OY0,UH0,UW0
Mgh 2-es kategória (hangsúlyos)	AÁ,AÉ,AH1,AÓ,AW1 AY1,EH1,ER1,EY1,IH1, IY1,OW1,OY1,UH1,UW1

hangsúlyt nem jelölő ábécét használjuk. Azt már tudjuk, hogy a schwa a leggyakoribb szegmentum, így most is ez fog legelőször sorra kerülni, és az 1-es kategóriát megnyitni. Hipotézisünk az volt, hogy a schwához azok a magánhangzók fognak csatlakozni az 1-es kategóriába, melyek *tipikusan hangsúlytalan* szótagban fordulnak elő, míg a másik osztályba a *tipikusan hangsúlyos* magánhangzók kerülnek. A hipotézishez hasonló eredményt kaptunk.

8. táblázat. Szuhotyin algoritmusa az angol magánhangzókra (2)

Cluster	Fonémák
Mgh 1-es kategória	AX,ER,IH
Mgh 2-es kategória	AA,AE,AH,AO,AW,AY,EH,EY,IY,OW,OY,UH,UW

3.3. 2-means klaszterezés

A klaszterezési algoritmusok valamilyen hasonlóság alapján osztanak részekre egy objektumhalmazt. A számítógépes nyelvészetben elsősorban valamiféle jegyek szerinti csoportosítás felügyelet nélküli megtanulására használják őket, mivel nem szükséges hozzájuk semmiféle előzetes feltevés a csoportokról. Számos klaszterezési módszer ismeretes, ezek közül mi a k-means algoritmust használtuk.

Leírás. A k-means algoritmus a klaszereket a tömegközéppontjuk segítségével definiálja. Egy valós V vektortér elemeit klaszterezzük. Kezdetben véletlenszerűen választunk ki k darab pontot középpontnak ($\mu_j \in V$), majd i, minden pontot besorolunk abba a klaszterbe, amelynek középpontjához legközelebb esik, ii, újraszámítjuk a tömegközéppontok helyét

$$\mu_j = \frac{1}{|C_j|)} \sum_{x \in C_j} x$$

i és ii iterálását addig folytatjuk, míg a középpontok helye már nem változik. Mivel az algoritmus nem feltétlenül a megoldáshoz konvergál, ezért többszöri futtatásra van szükség.

Távolságfüggvénynek használható az euklideszi távolság vagy valamilyen másik távolságfügvény (korreláció, city block). Mi az euklideszi távolságot

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

használtuk, de nem okozott jelentős különbséget más metrika használata sem.

Alkalmazások. Az algoritmus Szuhotyinéhez hasonlóan jól teljesített a magán- és mássalhangzók szétválasztásakor és kiváló eredményt adott a magyar korpuszokon a magánhangzók elölségi osztályozásakor annak ellenére, hogy a két jegy viselkedése gyökeresen különbözik, az előbbi értékei alternálni, az utóbbiak harmonizálni szeretnek.

Magán és mássalhangzók. A különféle korpuszokon többé-kevésbé hibátlanul a magán- és mássalhangzókat kaptuk meg a teljes fonémakészleten való futtatáskor.

Az angol NG és a japán n hibás osztályozására magyarázatot adhat azok speciális eloszlása, az, hogy sohasem fordulnak elő szótagkezdetben és általában magánhangzó előtt.

Elölségi harmónia. A 2-means algoritmust a magyar korpuszok magánhangzóin futtatva a két elölségi osztályt (a,á,o,ó,u,ú ill. e,é,i,í,ö,ő,ü,ű) kaptuk meg, némi bizonytalansággal az i-k besorolásában, ami jól illeszkedik a magyar fonológiai ismereteinkhez.

Corpus	Cluster	Phonemes
Czech	0	ý,ó,a,é,y,o,ě,á,e,í,ů,ú,i,u
	1	ň,k,v,w,č,ž,l,x,m,b,c,n,z,d,c,p,f,ř,g,r,š,ď,s,h,t,j
Hungarian	0	ó,a,é,ő,ö,o,á,e,í,ú,ű,i,ü,u
	1	ny,v,k,l,b,m,zs,c,n,d,z,ty,p,cs,q,f,dzs,sz,r,g,gy,h,s,t,j
English w accents	0	lY0,AE0,lY1,AY0,AO0,EH0,AE1,UW0,AO1,NG,AY1,ER0,EH1,AW0,ER1,UW1,Z,AA0,AW1,OY0,EY0,AA1,IH0,UH0,IH1,OY1,EY1,UH1,OW0,OW1,AH0,AH1
	1	HH,V,K,W,L,B,M,Y,ZH,N,DH,D,CH,P,F,R,G,JH,S,TH,T,SH
English w/o accents	0	HH,V,K,W,L,B,M,Y,ZH,N,DH,D,CH,P,F,R,G,JH,S,TH,T,SH
	1	AW,AA,UW,AY,ER,AO,NG,EH,AE,Z,lY,AH,UH,OW,IH,EY,O
French	0	@,è,A,é,y,õ,Ö,O,o,ö,E,ã,^,i,Ü,u
	1	k,v,!,w,l,b,m,n,z,d,p,f,g,r,s,S,t,J,*,j
Maori	0	k,w,m,n,ng,p,wh,r,h,t
	1	a,ē,ō,o,e,ī,ā,i,ū,u
Japanese	0	a,n,o,:,e,i,u
	1	v,k,w,x,m,b,C,y,z,d,p,f,',G,g,r,h,S,sT,t,J,j

1. ábra. 2-means clustering results for each corpora

Egyéb osztályok. Az algoritmus az angol korpuszon a mássalhangzókra futtatva többé-kevésbé a zöngés-zöngétlen distinkciót adta (HH, K , Y , Z , CH, P , F , JH, S , TH, T , SH és V , L , B , M , ZH, NG, N , DH, D , G , R), ami egy tipikusan harmonizáló jegy, míg a zöngéseken belül az egymással alternáló szonoráns-obstruens osztályok jöttek ki.

3.4. Függvénymaximalizálás

A számítástudományban megszokott módszer, hogy egy feladat optimális megoldását egy célfüggvény maximalizálásával keressük. A következőkben egy olyan egy adott korpusz fonémáinak partícióin értelmezett függvényt mutatunk, amely maximumá egy olyan partíción veszi fel, amelynek elemei maximálisan harmonizálnak vagy alternálnak a korpuszban.

Leírás. Tegyük fel, hogy adott egy korpusz bigram gyakorisági $n \times n$ mátrixa M. Ekkor a fonémák egy p $0,1$-értékű n hosszú vektorral adott partíciójára összegezhetjük azokat a bigram gyakoriságokat, amelyek egy osztály elemein belüli bigramokhoz tartoznak (f_h) illetve azokat, amelyek két különböző osztály elemei közötti bigramokéi ($f_a = C - f_h$, ahol C egy konstans, az összes bigram

gyakoriság öszege, ezzel normálva $f'_a = 1 - f'_h$). A két szám különbsége így annak négyzete $f = (f'_a - f'_h)^2 = (2f'_a - 1)^2$ akkor maximális, ha *valamelyik* gyakorisági összeg maximális.

Az $f(p)$ például az alábbi módon számolható ki:

- Legyen e egy n hosszú 1-vektor $((1, 1, ..., 1))$
- Legyen $M_p = \begin{pmatrix} tr((e-p)^T M(e-p)) & tr((e-p)^T Mp) \\ tr(p^T M(e-p)) & tr(p^T Mp) \end{pmatrix}$
- Végül legyen $f(p) = \left(1 - 2tr\left(\frac{M_p}{\|M_p\|_{tc}}\right)\right)^2$ az optimalizálandó függvény, ahol a $\|\|_{tc}$ a mátrix taxi normája, azaz $\sum_{i,j} m_{ij}$

Keressük tehát a adott M-re $ArgMax\, f(p)$-t. Ezt egy mohó rekurzióval végezzük, egy véletlen partícióból indulunk ki és egy-egy elemet osztályozunk át, valahányszor az $f(p)$ értékét ez növeli.

A mohó rekurzió

- Egy véletlen p választása, $f(p)$ kiszámítása
- Amennyiben létezik p', amely egyetlen elem osztályának megváltoztatásával áll elő és $f(p') > f(p)$, akkor a rekurzió meghívása p'-re

Alkalmazások. Ez az algoritmus a klaszterezéssel szinte teljesen megegyező eredményeket adott a teljes fonémakészleten (magán- és mássalhangzók), a magyar magánhangzókon (elölségi osztályok) és az angol mássalhangzókon (zöngésség és szonoritás).

4. Összefoglalás

A fentiekben bemutattunk három gépi tanulási módszert, amelyek segítségével különböző korpuszokon, különféle fonémaosztályok megtanítását végeztük el. Az eredmények azt mutatják, hogy – legalábbis bizonyos– természetes osztályok megtanulhatók mindenféle előzetes fonológiai, akusztikai vagy fonetikai feltételezések nélkül is.

4.1. További feladatok

Az ismertetett algoritmusok lehetőségeit még korántsem merítettük ki. Számos nyelvet és korpuszt kell még megvizsgálni ahhoz, hogy kiderüljön, a fenti módszerek valamelyike, vagy esetleg többük is alkalmas-e egy korpusszal adott nyelv (legalább részleges) fonológiai reprezentációjára.

Szükséges lenne továbbá az algoritmusok teljesítményének optimalizálására is.

Az ismertetett algoritmusokon kívül további feladat más módszerek keresése és alkalmazása is fonémaosztályozási feladatokban, mint például a rejtett Markov-modell (HMM) használata.

Hivatkozások

1. J. Durand, Siptár P.: Bevezetés a fonológiába. Osiris, Budapest (1997)
2. M. Eisen, M. de Hoon: Cluster 3.0 Manual for Windows, Mac OS X, Linux, Unix. Standford University (1999), University of Tokyo (2002)
3. J. Goldsmith: Phonological Complexity. Software and corpora at http://hum.uchicago.edu/%7Ejagoldsm/PhonologicalComplexity/
4. J. Goldsmith, A. Xanthos: Learning phonological categories Draft at http://hum.uchicago.edu/%7Ejagoldsm/Papers/phonolcat.pdf
5. A. K. Jain, R. C. Dubes: Algorithms for clustering data. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1988)
6. Kornai, A, Halácsy, P, Nagy, V, Oravecz, Cs, Trón, V, and Varga, D (2006). Web-based frequency dictionaries for medium density languages. *In: Proceedings of the 2nd International Workshop on Web as Corpus, edited by Adam Kilgarriff, Marco Baroni ACL-06, pages 1–9*
7. Sukhotin, B. V.: Eksperimental'noje vydelenie klassov bukv s pomoščju evm. Problemy strukturnoj lingvistiki, 234:189–206 (1962)