Igor Mijić

0036442326

SEMINARSKI RAD -



Sažetak

* Seminar je namijenjen studentima elektrotehnike
* Potrebno je poznavanje programskog paketa MATLAB
* Bit će prikazan rudimentaran program za prepoznavanje govornih naredbi

Ovim seminarskim radom će biti prikazan osnovni sustav za prepoznavanje govornih naredbi. Sustav je izvršen kao *Isolated Word Speaker Independent Speech Recognition* sustav, tj. sustav koji prepoznaje govorne realizacije iz konačnog skupa naredbi, te je neovisan o govorniku čije govorne realizacije koristi. Radom će biti prikazane osnovne cjeline potrebne za rad sustava i teoretska podloga iza svake od cjelina. Kao osnovne probleme sustava za prepoznavanje govora dani su ekstrakcija značajki i predikcija koji se inače izvršavaju MFC analizom i predikcijom korištenjem Skrivenih Markovljevih modela. U ovom radu koristi se MFC analiza i SVM sustav strojnog učenja za predikciju. Prednost korištenja SVM sustava je u manjoj složenosti sustava, lakšem treniranju, te manjoj količini potrebnih računalnih operacija. Uz očitu primjenu u pametnoj kući, sustav se još može koristiti za pomoć osobama sa posebnim potrebama, kod automatske transkripcije govora i sl..

Sadržaj

Ovaj seminarski rad je izrađen u okviru predmeta „“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

[1. Uvod 3](#_Toc357953805)

[2. Praktična izvedba sustava 4](#_Toc357953806)

[2.1. Snimanje govornih naredbi 4](#_Toc357953807)

[2.2. Detekcija i filtracija govora 4](#_Toc357953808)

[2.3. MFC analiza 5](#_Toc357953809)

[2.4. LIBSVM treniranje i predikcija 6](#_Toc357953810)

[2.5. Rezultati 7](#_Toc357953811)

[3. Zaključak 9](#_Toc357953812)

[4. Literatura 10](#_Toc357953813)

[5. Pojmovnik 11](#_Toc357953814)

[6. DODATAK: MATLAB programski kodovi 12](#_Toc357953815)

# Uvod

U računarskoj znanosti prepoznavanje govora ( *eng. speech recognition*) definiramo kao prijevod izgovorenog ljudskog govora u digitalni tekstualni format. Generalno možemo govoriti o ekstrakciji značajki iz snimke govornog podatka, te obrade istih značajki koje računalo dovode do zaključka koja je riječ izgovorena. Bilo da se koristi za automatski zapisnik kod medicinskih operacija ili kao sustav za upravljanje rasvjetom, prepoznavanje govora primjenu može naći u mnogim situacijama u rasponu od svakodnevnog života do vojne industrije, te se iz tog razloga zadnjih godina intenzivno razvija. Neke od primjena navedene su ovdje:

* govorna korisnička sučelja
* unos jednostavnih podataka
* govorno upravljanje uređajima (u rasponu od rasvjete u kućanstvima do upravljanja modernim mlaznim avionima)
* sučelja za pomoć osobama sa posebnim potrebama

Današnji sustavi za prepoznavanje govora uvelike variraju u kompleksnosti, mogućnostima izvedbe, te principima rada, sve ovisno o tome za kakvu primjenu su izvedeni. No kod svih sustava možemo primjetiti neke zajedničke značajke. Prva od njih je ekstrakcija značajki signala (*eng.feature extraction*) koja nam daje određenu „kompresiju“ govornog signala u vektore značajki (*eng. feature vector*). Uz određenu količinu filtracije kojom izvlačimo akustička svojstva govora ona spada u takozvanu predobradbu signala. Daljnja izvedba sustava bazira se ili na računanju neke od matematičkih udaljenosti ili statističkih modela koji su pogodni za strojno učenje koji iz navedenih vektora značajki odrađuju samo prepoznavanje govorne naredbe.

# Praktična izvedba sustava

Blok dijagram cijelog sustava opisanog u narednim potpoglavljima prikazan je na slici 1.



Slika : Blok dijagram sustava

## Snimanje govornih naredbi

Za potrebe treniranja i testiranja modela snima se određen broj realizacija svake od riječi koje se prepoznaju, dok se pri korištenju sustava koristi integrirana MATLAB funkcija *wavrecord*. Za testiranje i treniranje modela snimljeno je 5 govornika (tri muška, dva ženska), po 10 realizacija svake od riječi koje se prepoznaju korištenjem programske podrške Audacity, te *low end* kompjuterskog mikrofona. Signali se snimaju kao *.wav* datoteke uz frekvenciju otpikavanja *fs=8000 kHz*  i kvantizaciju sa 16 bita.

Sustav treba prepoznavati naredbe povezane sa sustavima u pametnoj kući (‘uključi’, ‘isključi’, 'pojačaj', 'smanji', 'svijetlo', 'alarm', 'budilicu')što nas dovodi do ukupnog broja od 350 snimljenih realizacija. Naredbe su provizorno izabrane, te se mogu dodati naredbe vezane za bilo koji ostali sustav u kući. U daljnjem radu sa sustavom taj broj je podložan promjenama ovisno o dodavanju novih naredbi, ili snimanju kvalitetnijih snimki.

## Detekcija i filtracija govora

Pošto snimljene realizacije nisu sačinjene isključivo od govora, nego i od određenog vremena kada govornik ne govori (kada se snima samo šum i vanjski izvori zvuka), govor se mora detektirati, te se mora izdvojiti isključivo uzorke koji sadržavaju govor. Za to koristimo funkciju *crop\_speech.m* koja računa snagu signala na blokovima uzoraka te odbacuje uzorke koji ne sadržavaju dovoljno energije. Nadalje govorne signale prije obrezivanja filtriramo funkcijom *filter* koja izdvaja govorni signal u rasponu od 350 Hz-a do 3.75 kHz-a. Za filtraciju koristimo FIR filtar projektiran Parks-McLellanovom metodom sa valovitošću u pojasu gušenja od minimalno 40 dB-a.

## MFC analiza

Niz MFC koeficijenata (eng. mel frequency cepstral coefficients) zajedno daje reprezentaciju kratkotrajnog spektra snage zvuka, a bazira se na linearnoj kosinusnoj transformaciji logaritma spektra snage izraženog u nelinearnoj mel-frekvencijskoj skali. Mel frekvencijska skala se koristi radi boljeg aproksimiranja ljudskog slušnog sustava. Princip izvedbe MFC koeficijenata prikazan je na slici 1.



Slika : Izvedba MFC koeficijenata

Govorni signal na izlazu iz filtra dovodimo na ulaz funkcije *melcepst.m*  koja provodi kepstralnu analizu na mel frekvencijskoj skali. Funkcija *melcepst.m* dio je VOICEBOX besplatnog *toolbox-a* za rad sa glasovnim podacima u MATLAB programskoj podršci. Vlastita funkcija *mfcc\_analyze.m* je izvedena ali radi brzine rada i modalnosti ulaznih i izlaznih parametara funkcijeizabrana je funkcija *melcepst.* Analizu provodimo nad blokovima uzoraka u trajanju 10 ms, s time da se blokovi preklapaju sa prošlima u 5 ms. Tako se iskorištavaju korelacijska svojstva govora, te za takve blokove možemo reći da su stacionarni u širem smislu, tj. da su podobni za daljnji rad. Odabran je Hammingov prozor,

Nad blokovima se tada provodi pretvorba u MFCC koeficijente po postupku opisanom u pretprošlom poglavlju, te se blokovi spremaju u matricu, te usrednjavaju. Takvi koeficijenti su spremni za LIBSVM treniranje sustava, te na kraju i za predikciju.

## LIBSVM treniranje i predikcija

U strojnom učenju SVM-ovi (*eng. support vector machines*) su modeli učenja povezani sa potrebnim algoritmima učenja koji analiziraju podatke, te se koriste za klasifikaciju i regresijsku analizu. U najjednostavnijem obliku SVM uzima određeni broj ulaznih podataka, te za svaki podatak vrši predikciju kojoj od dvije moguće klase pripada ulazni podatak. Algoritmi za treniranje SVM-ova uzimaju određeni broj realizacija koje su označene da pripadaju jednoj od dvije moguće klase, te grade model koji kasnije razvrstava ulazne podatke ovisno o izgrađenom modelu. SVM model je reprezentacija ulaznih realizacija kao točaka u višedimenzionalnom prostoru, postavljenih tako da su različite kategorije razdvojene jedna od druge jasnom granicom. Predikcija se vrši mapiranjem testnih podataka u isti prostor te klasifikacijom kategorija ovisno o tome s koje strane granice se podaci nalaze.

Kao što je već navedeno, koristimo LIBSVM library koja kao sučelje između funkcija koje sačinjavaju SVM model koristi dvije funkcije:

*svmtrain.c* : funkcija uzima određeni broj ulaznih podataka te gradi model

*svmpredict.c* : funkcija uzima određeni broj ulaznih podataka, te model, te na račun oba ulaza vrši predikciju klasa ulaznih podataka. Ako među ulazne podatke ubacimo i oznake klasa ulaznih podataka funkcija vraća i postotak točnosti predikcije. Baš taj postotak nam je najzanimljiviji kao mjera uspjeha rada sustava za prepoznavanje govora.

Da bi bili sigurni u rad sustava bitna nam je mjera točnosti istog. Jednostavan postupak za dobivanje mjere točnosti zove se krosvalidacija ili rotaciona estimacija. Krosvalidacija je tehnika za validaciju modela koja procjenjuje kako bi se rezultati neke statističke analize mogli generalizirati na neki drugi neovisni set podataka. Većinom se koristi kod prediktivnih sustava, te kada nam je potrebna procjena točnosti prediktivnog modela u praksi. Varijanta krosvalidacije koju koristimo naziva se desetorostruka krosvalidacija.

Kod desetorostruke krosvalidacije originalni uzorak podataka se nasumice particionira u deset podsetova podataka jednake veličine. Od deset podsetova podataka, jedan se uzima kao skup podataka za testiranje dok se prepostalih devet podsetova koristi za treniranje sustava. Koristeći prije navedene funkcije, izgradimo prediktivni model te vršimo predikciju i računanje točnosti iste. Proces krosvalidacije se tada ponavlja deset puta, te se svaki od deset podsetova podataka koristi kao skup podataka za testiranje. Deset rezultata svake od iteracija se tada usrednjavaju da bi se dobio unificirani podatak. Može se koristiti podjela podataka na bilo koji broj podsetova ali se u praksi najčešće koristi desetorostruka. Prednost ovakve metode naspram ponovljive nasumične podijele je u tome što se svi podaci koriste ujedno i za trenirani i testiranje, dok se svaki podatak koristi za testiranje točno jednom.

## Rezultati

Kao što je već spomenuto snimljena je baza glasovnih realizacija koja će se koristiti za treniranje i testiranje sustava, 10 realizacija po govorniku svake od riječi koje sustav mora prepoznavati, te 5 govornika. Koristeći metodu krosvalidacije opisanu u potpoglavlju o LIBSVM predikciji, te ponavljanjem navedenog postupka 1000 puta i usrednjavanjem rezultata došli smo do točnosti od 89.54%. Izdvajanjem isključivo jednog govornika čije realizacije nisu korištene za treniranje modela nego isključivo za testiranje istog pokazana je neovisnost o govorniku.

## Preporučena izvedba u pametnoj kući

Čak i sa ovako testiranim sustavom daleko smo od cijelog praktično izvedenog sustava. MATLAB programski kod nije pogodan za rad u stvarnom vremenu, te se generalno koristi za simulacije matematičkih sustava. Iz tog razloga biti će predstavljena teoretska osnova za praktičnu izvedbu.

Kako bi sustav radio u cijeloj pametnoj kući potreban je rad u stvarnom vremenu i rad sa više ulaza. Za rad sa više ulaza koristio bi se sustav sa više omnidirekcionalnih dinamičkih mikrofona ugrađenih u stropove prostorija u pametnoj kući i programabilnom višeulaznom zvučnom karticom spojenom na računalo koje bi obavljalo program. Takav odabir mikrofona bi bio radi cijene (dinamički mikrofoni s kvalitetnim karakteristikama u frekvencijskom području ljudskog govora su jeftiniji od generalno kvalitetnijih kondenzatorskih ), te radi pokrivanja područja (omnidirekcionalni mikrofon postavljen u središte stropa prostorije može pokrivati svaku prostoriju pravilnog oblika ). Pošto većina današnjih kvalitetnih zvučnih kartica omogućava određenu količinu programabilnosti, najčešće korištenjem C++ ili Java programskih jezika, pretpostavljamo izvedbu u C++ programskom jeziku koja bi nam ujedno omogućila integraciju LIBSVM biblioteke koja nam je potreban.

Program bi tako u petlji obavljao akviziciju podataka sa zvučne kartice, te u više dretvi obrađivao podatke sa svakog od ulaza zvučne kartice, obavljajući predikciju korištenjem unaprijed treniranih SVM modela na svakoj od dretvi.

Računalo koje obavlja prepoznavanje govora tada može sa ostalim uređajima u kući komunicirati na nekoliko različitih načina, ovisno o tome kako su ostali sustavi izvedeni. U slučaju izvedbe sustava koji su upravljani mikrokontrolerima, sustav bi bio opremljen vlastitim mikrokontrolerom koji sa računalom komunicira jednim od uvriježenih komunikacijskih protokola, te tako od njega zaprima naredbe, dok preko svojih izlaza spojenih na prekidne ulaze mikrokontrolera ostalih ustava kontrolira rad ostalih sustava.

Nadalje, da bi se ocijenilo uspješnost sustava u stvarni radnim uvjetima potrebno je izraditi prototipnu verziju sustava, no skoro odmah možemo pretpostaviti da će utjecaj šuma smanjivati točnost rada sustava. Za to predlažemo ubacivanje dodatne funkcije u sami sustav koja bi na ulazu u sustav nakon obrezivanja govorne informacije, računala srednju vrijednost govornih signala i oduzimala je od samog signala. Tim postupkom iz signala oduzimamo veliku količinu šuma, te određenu količinu informacije ali MFC koeficijenti ostaju dovoljno jedinstveni da bi se moglo vršiti treniranje i predikcija.

# Zaključak

Sustavi za prepoznavanje govora su već par desetljeća fokus velike količine istraživanja. S povećanjem računalne moći dovedeni su do visoke razine iskoristivosti, te visoke kvalitete rada u praksi, no još uvijek postoji mnogo prostora za napredak.

Ovim radom dan je uvod u osnovna razmatranja sustava za prepoznavanje govora, te su opisane specifičnosti raznih vrsta istih. Nadalje, opisana je osnovna mehanika rada sustava za prepoznavanje govora koja se može vidjeti u gotovo svim sustavima, neovisno o kompleksnosti istih. Opisane su glavna metoda za ekstrakciju značajki iz govora u vidu mel frekvencijskih kepstralnih keoficijenata, te jednostavna metoda za treniranje i predikciju korištenjem LIBSVM modela strojnog učenja.

Ovakvim sustavom najviše bi se moglo pomoći osobama sa raznim invaliditetima, ili starijim osobama koji bi u konačnici puno lakše živjeli u pametnim kućama sa ovakvom funkcionalnošću.

Prvi slijedeći korak bi bio izraditi prototipni sustav po ranije navedenim smjernicama, dok se za daljnji rad sa sustavima za prepoznavanje govora preporučuje korištenje ustoličenih metoda poput HMM-ova za koje već postoje kvalitetno izgrađeni modeli poput HTK, ATK, MIT HMM Toolbox-a. Druga alternativa bi bila eksperimentiranje sa SVM sustavima za prepoznavanje govora koji klasificiraju foneme, a ne cijele riječi čime bi se moglo postići sustave koji prepoznaju velike količine riječi čime bi se moglo izvesti sustave koji nisu samo za upravljanje.

# Literatura

1. Speech Recognition Wikipedia Page, zadnje izmijenjeno 11. svibnja 2013. u 14:31, http adresa : <http://en.wikipedia.org/wiki/Speech_recognition>
2. Support Vector Machine Wikipedia Page, zadnje izmijenjeno 3. svibnja 2013. u 08:07, http adresa : <http://en.wikipedia.org/wiki/Support_vector_machine>
3. Padrell S.J., Iglesias-Martin D., Diaz de Maria F., Support Vector Machines for Countinuos Speech Recognition, 14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, 4.-8. rujna, 200.
4. Mel-frequency cepstrum Wikipedia Page, zadnej izmijenjeno 16. Travnja 2013., u 11:04, http adresa : <http://en.wikipedia.org/wiki/Mel-frequency_cepstrum>.
5. Molau S., Pitz M., Ney H., Computing Mel-frequency Cepstral Coefficients on the power spectrum, ICASSP,2001
6. Picone J., Fundamentals of speech recognition: A short course,1996

# Pojmovnik

|  |  |  |
| --- | --- | --- |
| **Pojam** | **Kratko objašnjenje** | **Više informacija potražite na** |
|  |  |  |
| ASR | Automatic Speech Recognition – Automatsko prepoznavanje govora | <http://en.wikipedia.org/wiki/Speech_recognition> |
| MFCC | Mel Frequency Cepstral Coefficients | <http://en.wikipedia.org/wiki/Mel-frequency_cepstrum> |
| SVM | Support Vector Machine – model za podržano vektorsko strojno učenje | <http://en.wikipedia.org/wiki/Support_vector_machine> |
| Krosvalidacija | Tehnika za validaciju modela strojnog učenja | [http://en.wikipedia.org/wiki/Cross-validation\_(statistics)](http://en.wikipedia.org/wiki/Cross-validation_%28statistics%29) |
| LIBSVM | SVM library implementirana u C programskom jeziku | <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> |
| HMM | Skriveni Markovljevi modeli | <http://en.wikipedia.org/wiki/Hidden_Markov_model> |
| Voicebox | MATLAB toolbox za rad s glasom | <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html> |

# DODATAK: MATLAB programski kodovi

%% crossval\_test.m - ulazna funkcija kod testiranja rada sustava - računa

%% krosvalidaciju da dobijemo točnost sustava

clear all

close all

clc

rmpath([matlabroot '\toolbox\bioinfo\biolearning']) ;

data=data\_prep(50,7);

labels=label\_prep(50,7);

N\_blok = 10;

x=size(data);

N\_sample\_blok=x/N\_blok;

for j=1:1000

indx=randperm(size(data));

permData=data(indx,:);

permLabels=labels(indx);

for i=1:10

 if i==1

 testData=permData(1:N\_sample\_blok,:);

 trainData=permData((N\_sample\_blok+1):end,:);

 testLabels=permLabels(1:N\_sample\_blok);

 trainLabels=permLabels((N\_sample\_blok+1):end);

 elseif i==10

 testData=permData(9\*N\_sample\_blok+1:end,:);

 trainData=permData(1:9\*N\_sample\_blok,:);

 testLabels=permLabels(9\*N\_sample\_blok+1:end);

 trainLabels=permLabels(1:9\*N\_sample\_blok);

 else

 testData=permData(i\*N\_sample\_blok+1:(i+1)\*N\_sample\_blok,:);

 trainData=[permData(1:i\*N\_sample\_blok,:); permData((i+1)\*N\_sample\_blok+1:end,:)];

 testLabels=permLabels(i\*N\_sample\_blok+1:(i+1)\*N\_sample\_blok);

 trainLabels=[permLabels(1:i\*N\_sample\_blok), permLabels((i+1)\*N\_sample\_blok+1:end)];

 end

 model=svmtrain(trainLabels',trainData);

 [predict\_label,acc(j,i,:),xpx] = svmpredict(testLabels',testData, model);

end

end;mean(mean(acc(:,:,1)))

%% data\_prep.m - funkcija učitava i priprema ulazne podatke

function [ mat ] = data\_prep( N\_word\_real,N\_word )

 mat=zeros((N\_word\_real\*N\_word),12);

 for j=1:N\_word

 switch j

 case 1

 word='uključi';

 case 2

 word='isključi';

 case 3

 word='pojačaj';

 case 4

 word='smanji';

 case 5

 word='svijetlo';

 case 6

 word='alarm';

 case 7

 word='budilicu';

 end

 for i=1:N\_word\_real

 ime=sprintf('%s (%d).wav',word,i);

 [speech,fs,nbits]=wavread(ime);

 crop=crop\_speech(speech);

 n=round(100\*(length(crop)/fs));

 c=melcepst(crop,fs,'M',12,n);

 mfcc=mean(c);

 mat((j-1)\*N\_word\_real+i,:)=mfcc;

 end

 end

end

%% label\_prep.m - priprema labele za testiranje i treniranje

function [ mat ] = label\_prep( N\_word\_real,N\_word )

 for j=1:N\_word

 for i=1:N\_word\_real

 mat((j-1)\*N\_word\_real+i)=j;

 end

 end

end

%% ------------------------FUNKCIJA - CROP\_SPEECH-----------------------

% Funkcija crop\_speech.m računa snagu blokova snimljenih realizacija

% signala odstranujući blokove koji ne nose dovoljnu količinu signala te

% tako spremajući islkjučivo signal koji sadrži govor

% Ulaz je snimka zvuka (snd\_rec) a izlaz je ekstaktirani govor

function [ speech ] = crop\_speech( snd\_rec )

 open b.mat;

 B = ans.b;

 snd\_rec = filter (B,1,snd\_rec);

 l = length(snd\_rec);

 Pwr = 0;

 for i=1:l

 Pwr = Pwr + snd\_rec(i)^2;

 end

 Pwr\_mean = (Pwr)/l;

 Pwr\_mean = 0.5\*Pwr\_mean;

 tmp = 0;

 N\_blok = round(l/50)-2;

 for j=1:N\_blok

 Pwr\_tmp = 0;

 for i=1:50

 Pwr\_tmp = Pwr\_tmp + snd\_rec(50\*j+i)^2;

 end

 Pwr\_blok = Pwr\_tmp/50;

 if Pwr\_blok > Pwr\_mean

 for k=1:50

 speech(tmp\*50 + k) = snd\_rec(j\*50+k);

 end

 tmp=tmp+1;

 end

 end

end

%% ulazna funkcija sustava kod rada

clear all

close all

clc

rmpath([matlabroot '\toolbox\bioinfo\biolearning']) ;

disp('Molimo pričekajte, pripremaju se podaci te se trenira SVM model.')

n\_w\_r=50;

n\_w=7;

data=data\_prep(n\_w\_r,n\_w);

label=label\_prep(n\_w\_r,n\_w);

model=svmtrain(label',data);

disp('Molimo, izrecite govornu naredbu');

fs=8000;

nbits=16;

x=wavrecord(3\*fs,fs,1);

disp('Naredba se obrađuje')

x=crop\_speech(x);

n=round(100\*(length(x)/fs));

c=melcepst(x,fs,'M',12,n);

test\_data=mean(c);

[predict\_label]= svmpredict(0,test\_data, model);

 switch predict\_label

 case 1

 disp('uključi');

 case 2

 disp('isključi');

 case 3

 disp('pojačaj');

 case 4

 disp('smanji');

 case 5

 disp('svijetlo');

 case 6

 disp('alarm');

 case 7

 disp('budilicu');

 end