Gender Classification in Speech Processing

Baiju M Nair(2011CRF3637)
Geetanjali Srivastava(2012EEZ8304)
Group No. 22

Introduction

- Main objective is gender classification in speech processing.

- Classify male and female speech using classification models
  SVM and neural network
  Understand the performance

- Extract the features which are more robust and most appropriate

- Data Set : Tested using Harvard-Haskins database

- Implemented in MATLAB 7.10

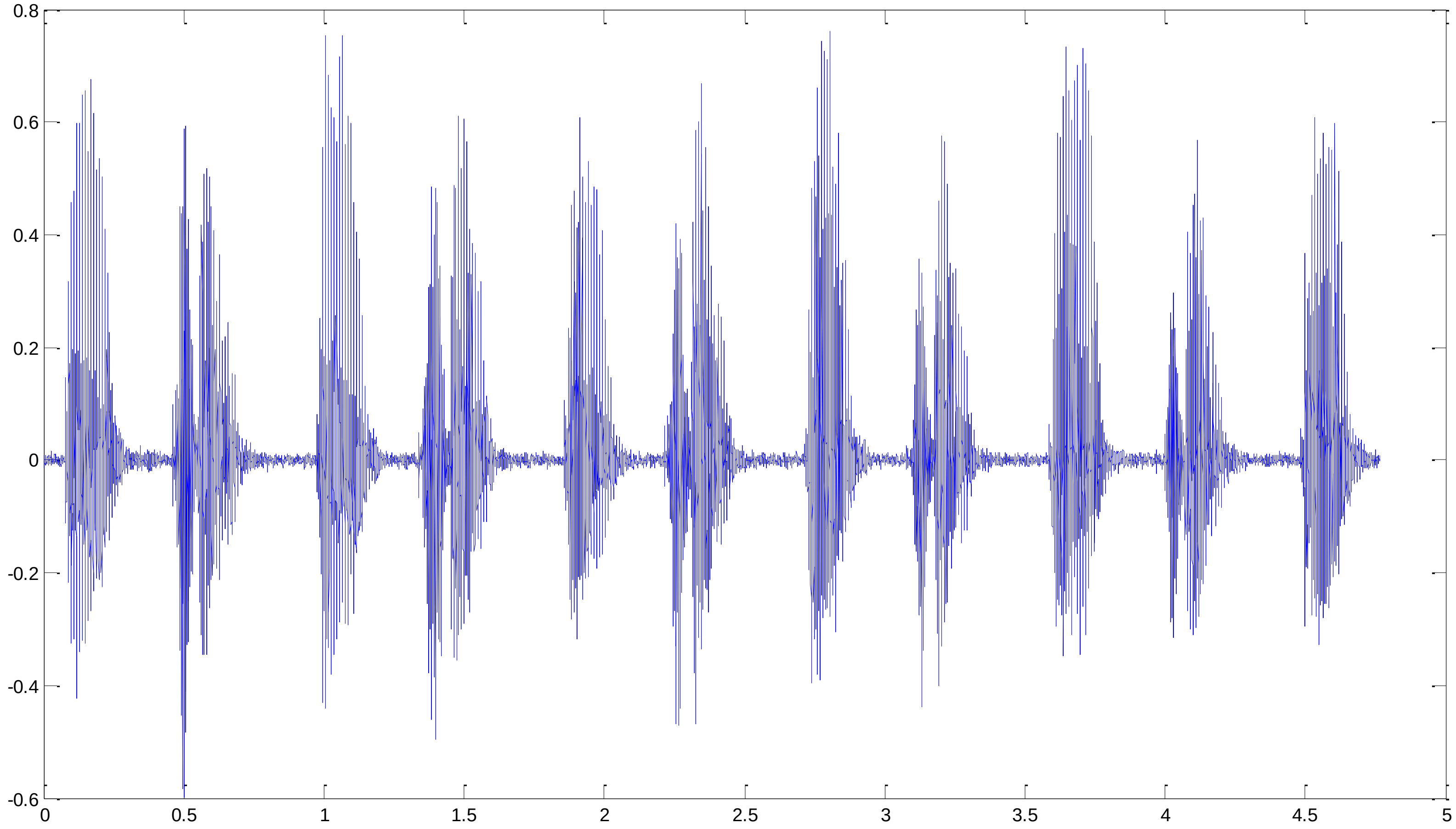
- Preprocessing of acoustic data in order to extract the features

- Application – Speech synthesis
  Speaker recognition
  Acoustic data analysis – heart beat, pulse

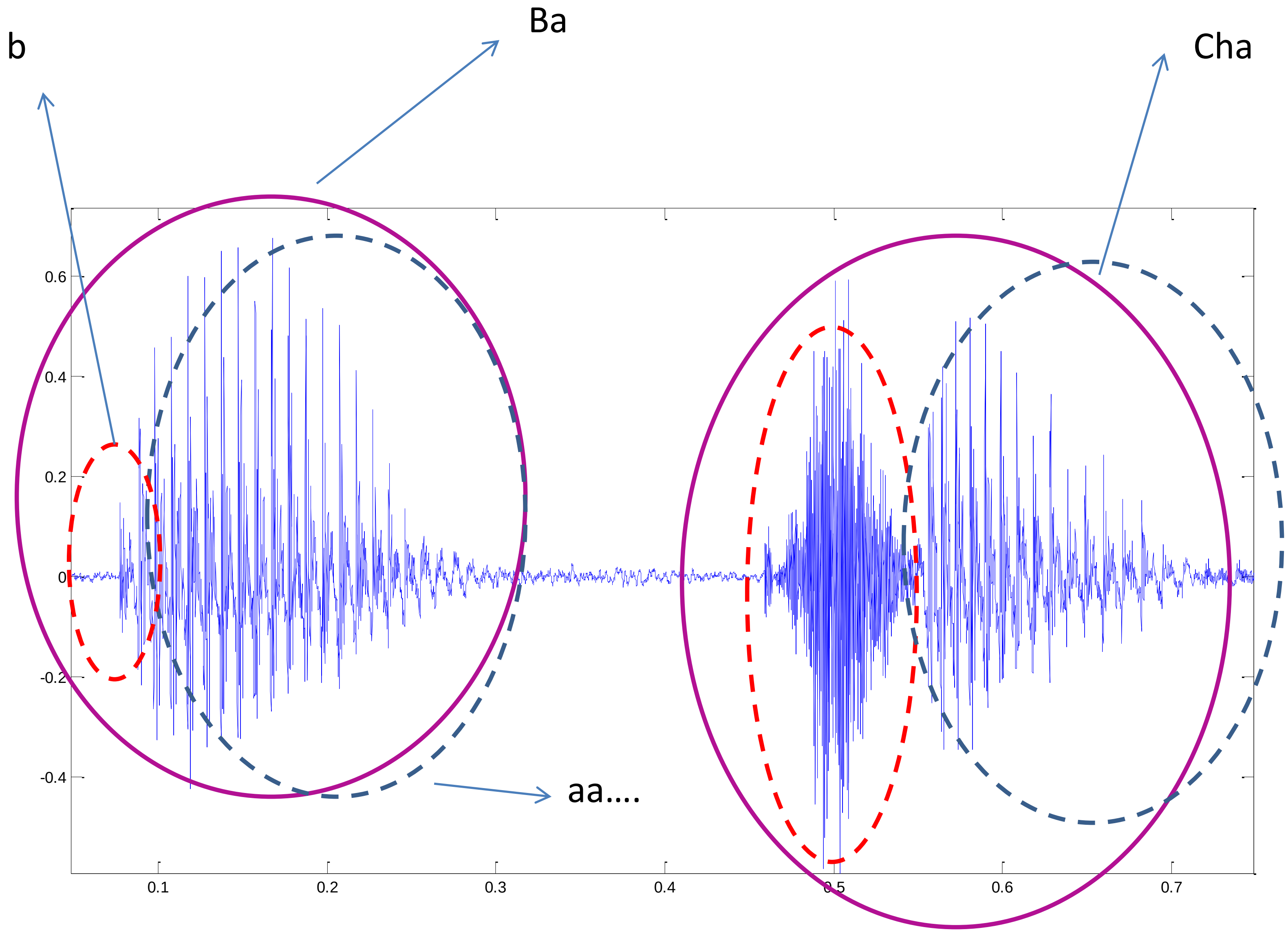
Data and Procedure

- **Extracted 549 acoustic data samples from the database**
- **Six Speakers uttering multiple syllable**
- **Acoustic data were collected in quiet rooms at Harvard and Haskins,**
  16-bit .wav format
  Sample rate = 10 kHz,
- **Read the acoustic data into a vector and load them in matlab**
- **Necessary in MATLAB to convert these particular .wav files back to their original physical units**
- **Analyzed in frequency domain and time domain to extract different feature vectors**

Time series of the signal (ba-cha)



Time series of the signal (ba-cha)



Feature Vectors

Better features

- **Energy Entropy** - Male low and distributed
  Female high and stays for short period of time

$$P(k) = \frac{|X(k)|^2}{\sum_{k=0}^{K} |X(k)|^2} , \qquad H = \sum_{k=0}^{K/2} P(k) \log(P(k)) ; \qquad M = (E - C_E)(H - C_H), \quad EE = \sqrt{(1 + |M|)}$$

- **Short time energy** – Male low , Female High

$$E_{\hat{n}} = \sum_{m=-\infty}^{\infty} (x[m]w[\hat{n} - m])^2 = \sum_{m=-\infty}^{\infty} x^2[m]w^2[\hat{n} - m].$$

- **Zero –crossing rate** – Female ZCR higher than male

$$\text{ZCR}, Z = \frac{1}{N} \sum_{i=1}^{N-1} \frac{\text{sgn}\{x(i)\} - \text{sgn}\{x(i-1)\}}{2} \qquad \text{sgn}\{x(i)\} = \begin{cases} 1; x(i) > 0 \\ 0; x(i) = 0 \\ -1; x(i) < 0 \end{cases}$$

Feature Vectors Contd…

Advanced feature

- **Spectral Centroid**

$$Centroid = \frac{\sum_{n=0}^{N-1} f(n)\, x(n)}{\sum_{n=0}^{N-1} x(n)}$$

- **Frame based teager energy**

$$f_i = w_i^2 X(w_i)\,. \qquad\qquad T_i = \left(\sum_{k=1}^{K} f_k\right)^{1/2}.$$

- **Position of Maximum FFT coefficient**

Position of Maximum FFT coefficient divided by sampling frequency

Classification models

SVM -Support vector machine

Neural network based method

		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)	Positive predictive value = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Test Outcome Positive}}$
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**	Negative predictive value = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Test Outcome Negative}}$
		Sensitivity = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	**Specificity** = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$	

Performance Comparison Parameters

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{(TP + FP)}$$

$$Sensitivity = \frac{TP}{(TP + FN)}$$

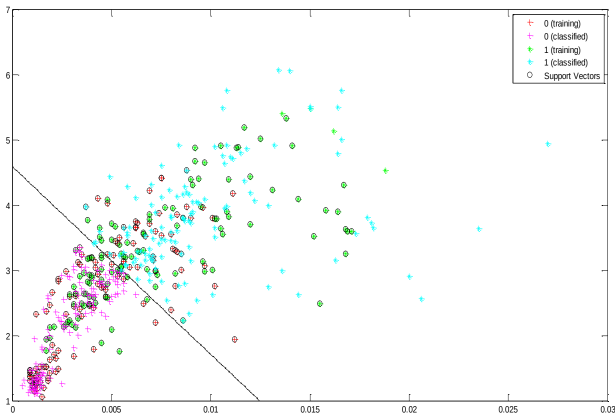
$$Specificity = \frac{TN}{(FP + TN)}$$

$$Likelihood\,ratio\ positive(LRP) = \frac{Sensitivity}{(1 - Specificity)}$$

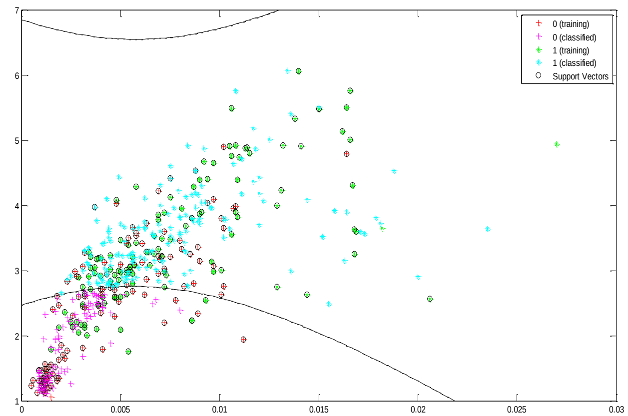
$$Likelihood\,ratio\ negative(LRN) = \frac{(1 - Sensitivity)}{Specificity}$$

TP : True Positive (Positive cases that were Correctly identified)
TN : True Negative(Negatives cases that were Incorrectly classified as Positive)
FP : False Positive(Negatives cases that were classified Correctly)
FN : False Negative(Positives cases that were Incorrectly classified as Negative)

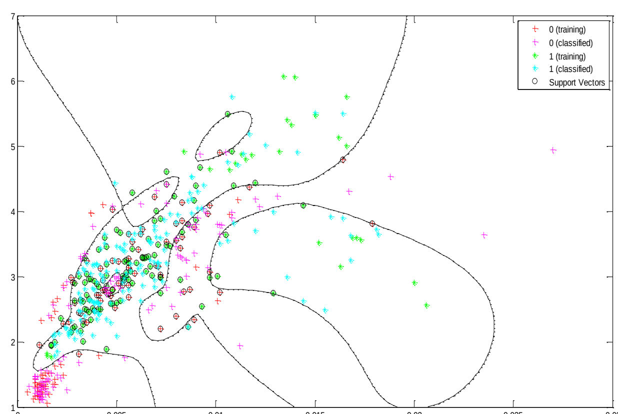
SVM Results



Linear Kernal Function

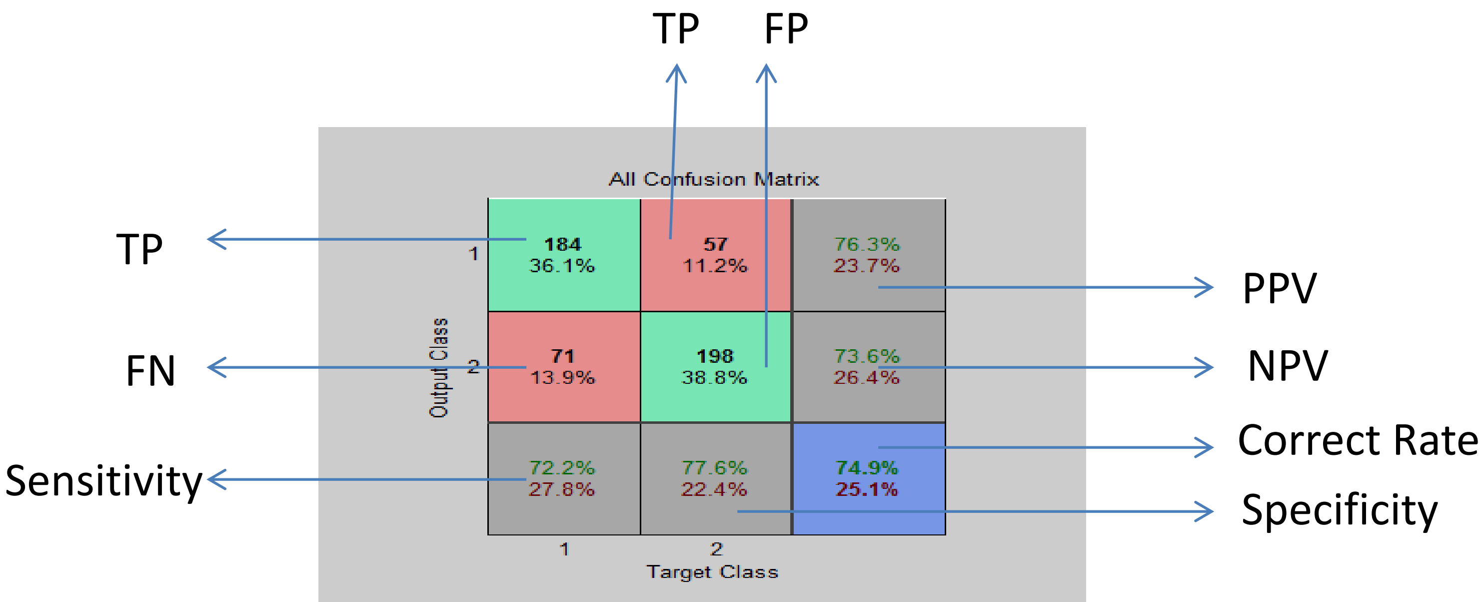


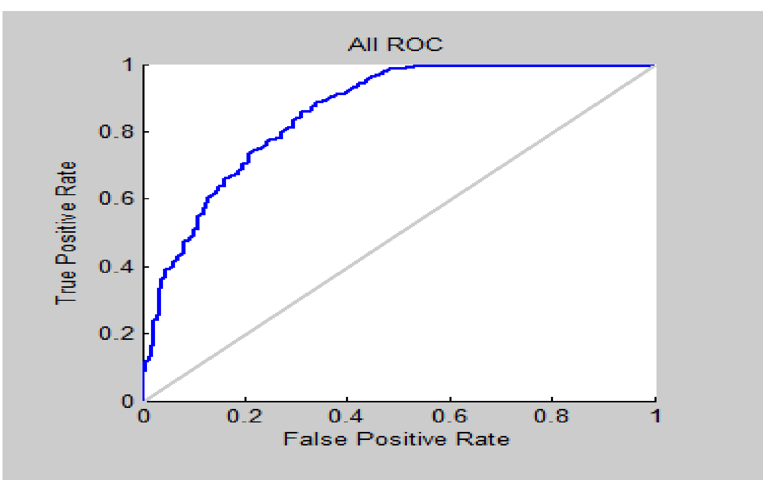
Quadratic Kernal Function



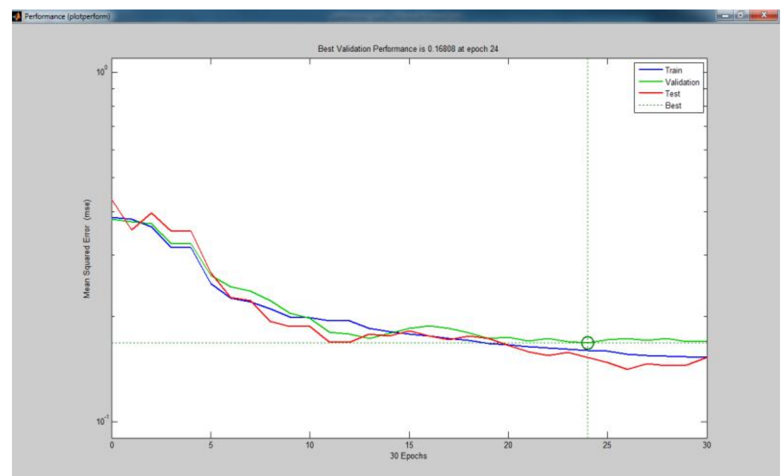
RBF Kernal Function

Neural Network Results





ROC curve



MSE curve

Performance Comparison

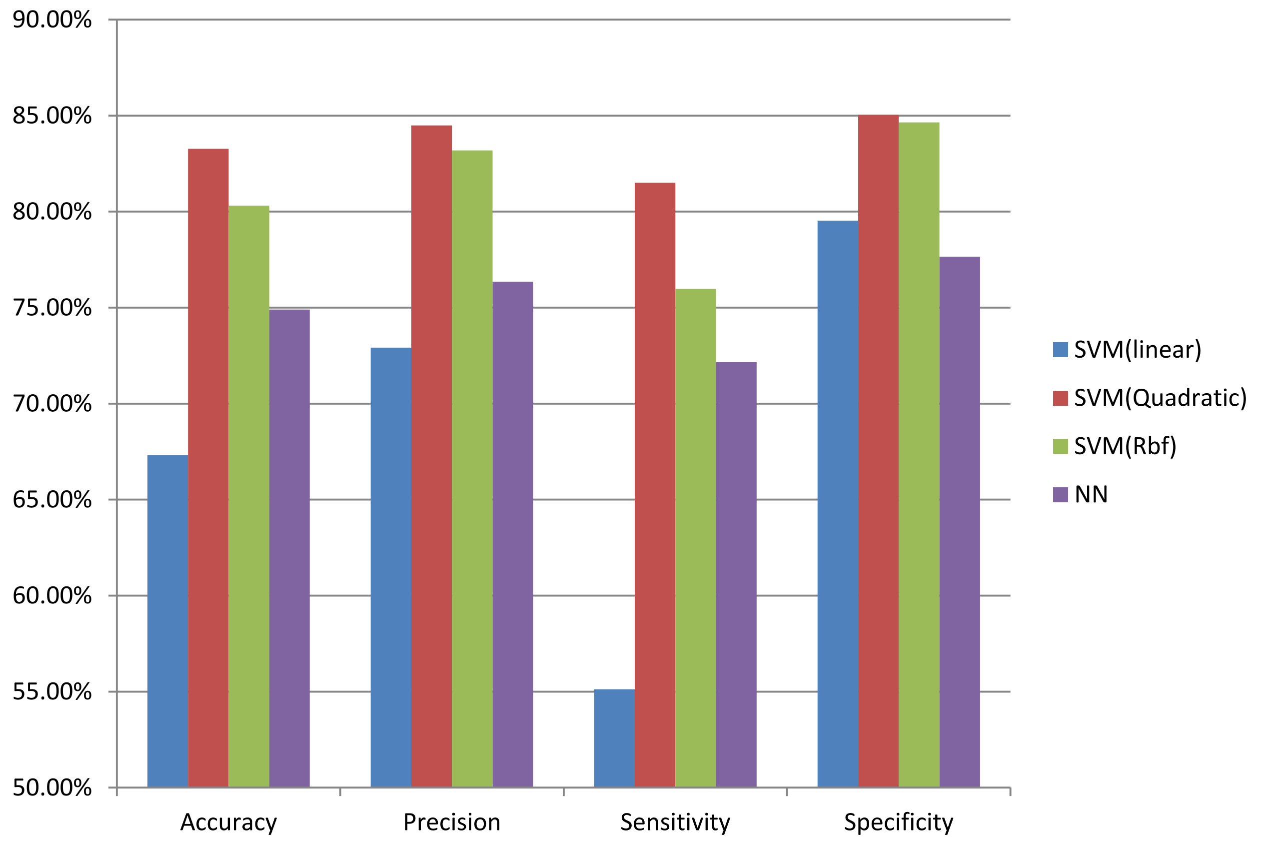
	SVM									NN		
	Linear			Quadratic			RBF					
	Acc	PPV	NPV	Acc	PPV	NPV	Acc	PPV	NPV	Acc	PPV	NPV
1	0.3819	0.3944	0.3661	0.6654	0.6641	0.6667	0.8484	0.9005	0.8084	0.853	0.804	0.921
2,6	0.5394	0.5379	0.541	0.622	0.6281	0.6165	0.687	0.7654	0.6444	0.631	0.614	0.656
4,5	0.6614	0.7733	0.6145	0.7402	0.746	0.7346	0.7736	0.7704	0.7769	0.727	0.72	0.736
1,5	0.6732	0.6719	0.6746	0.6575	0.6754	0.6429	0.811	0.8347	0.7904	0.722	0.698	0.751
1,4	0.4961	0.4947	0.4969	0.5906	0.6018	0.5816	0.7933	0.8066	0.7811	0.663	0.64	0.695
2,3,6	0.5591	0.5573	0.561	0.5906	0.5878	0.5935	0.6516	0.7037	0.6207	0.633	0.626	0.642
All	0.6417	0.6385	0.6452	0.813	0.863	0.7751	0.7736	0.7884	0.7603	0.727	0.734	0.721

Feature Analyzed

1 : Normalized maximum FFT Coefficient
2 : Energy Entropy
3 : Zero Crossing Rate
4 : Frame Based Teager Energy
5 : Spectral Centroid
6 : Short Term Energy

Acc: Accuracy
PPV: Positive Predictive Value
NPV: Negative Predictive Value

Performance Comparison Contd…



Conclusion

- **Comprehensive evaluation of classification methods for Gender determination.**

- **SVM with Quadratic kernel function has a better accuracy rate of 81% when all the feature vector are used in training.**

- **Features like Normalized maximum FFT Coefficient and frame based teager energy which are both frequency domain features has a better accuracy (80%) with RBF kernel function.**

- **Over all score of SVM is better than Neural Network.**

References

1. M. Gomathy, K. Meena and K. R. Subramaniam, 'Gender Clustering and Classification Algorithms in Speech Processing: A Comprehensive Performance Analysis", International Journal of Computer Applications (0975 – 8887) Volume 51– No.20, August 2012

2. Hong,Kook Kim, Hwang Soo Lee. Use of spectral autocorrelation in spectral envelope linear prediction for speech recognition., IEEE Transactions on SAP, Vol.7, 1999, No 5, 533-541.

3. F. Jabloun, A.E. Cetin, and E. Erzin, "Teager Energy Based Feature Parameters for Speech Recognition in Car Noise", *IEEE Signal Processing Letters, Vol. 6, No.10, pp. 259–261, 1999.*

4. Caruntu, A. Nica, and G. Toderean ,"Robust Features for Speech Classification" ,Documents on internet

5. Lawrence Rabiner, Biing-Hwang Juang ,Fundamentals of Speech Recognition , Pearson Education,2003

Thank you

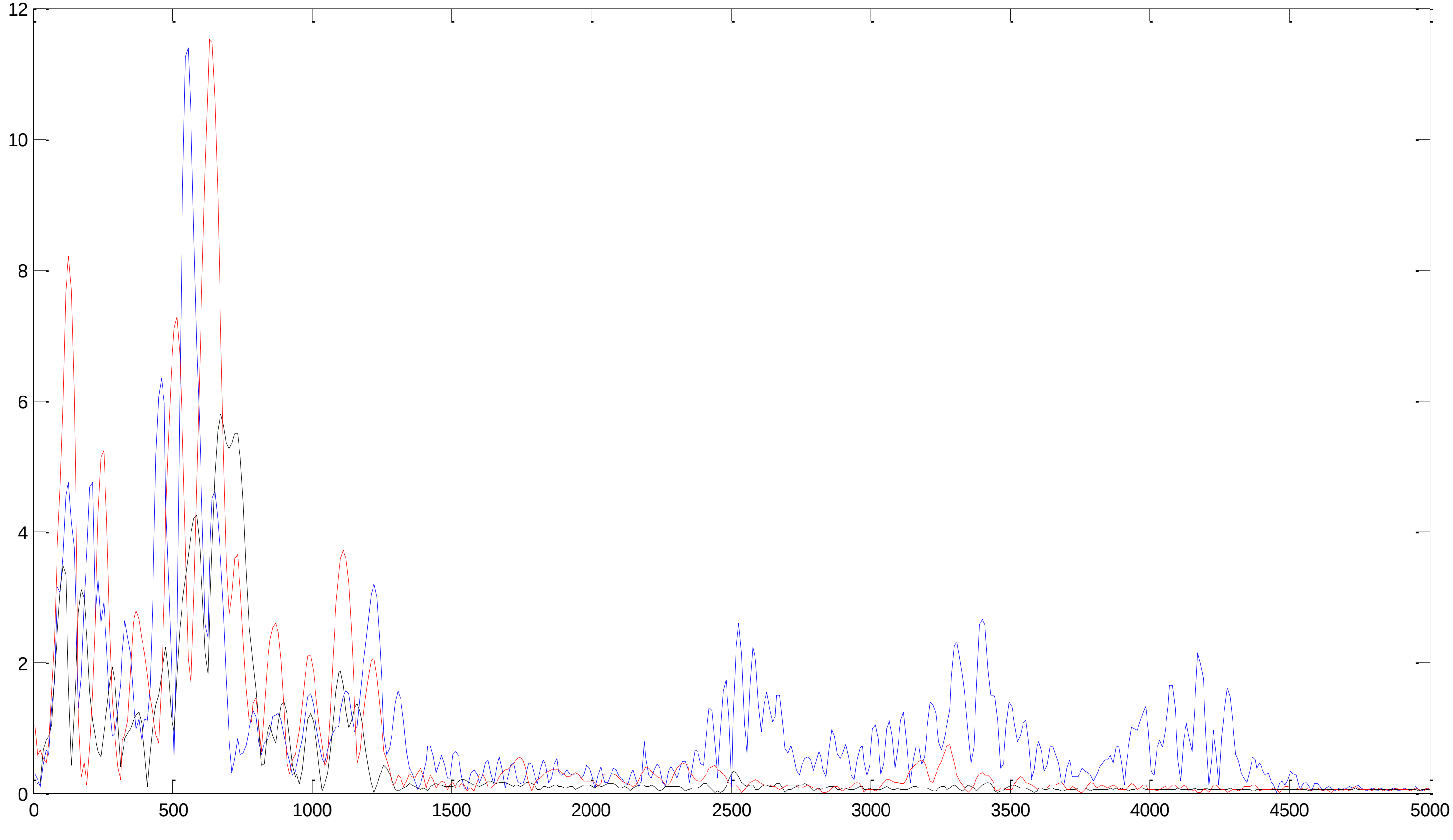
Appendix

```
In classi at 21
                  Label: ''
            Description: ''
            ClassLabels: [2x1 double]
            GroundTruth: [510x1 double]
   NumberOfObservations: 510
         ControlClasses: 2
          TargetClasses: 1
      ValidationCounter: 2
     SampleDistribution: [510x1 double]
      ErrorDistribution: [510x1 double]
SampleDistributionByClass: [2x1 double]
 ErrorDistributionByClass: [2x1 double]
         CountingMatrix: [3x2 double]
            CorrectRate: 0.7913
              ErrorRate: 0.2087
        LastCorrectRate: 0.7992
          LastErrorRate: 0.2008
       InconclusiveRate: 0
         ClassifiedRate: 1
            Sensitivity: 0.8110
            Specificity: 0.7717
 PositivePredictiveValue: 0.7803
 NegativePredictiveValue: 0.8033
     PositiveLikelihood: 3.5517
     NegativeLikelihood: 0.2449
             Prevalence: 0.5000
        DiagnosticTable: [2x2 double]
```

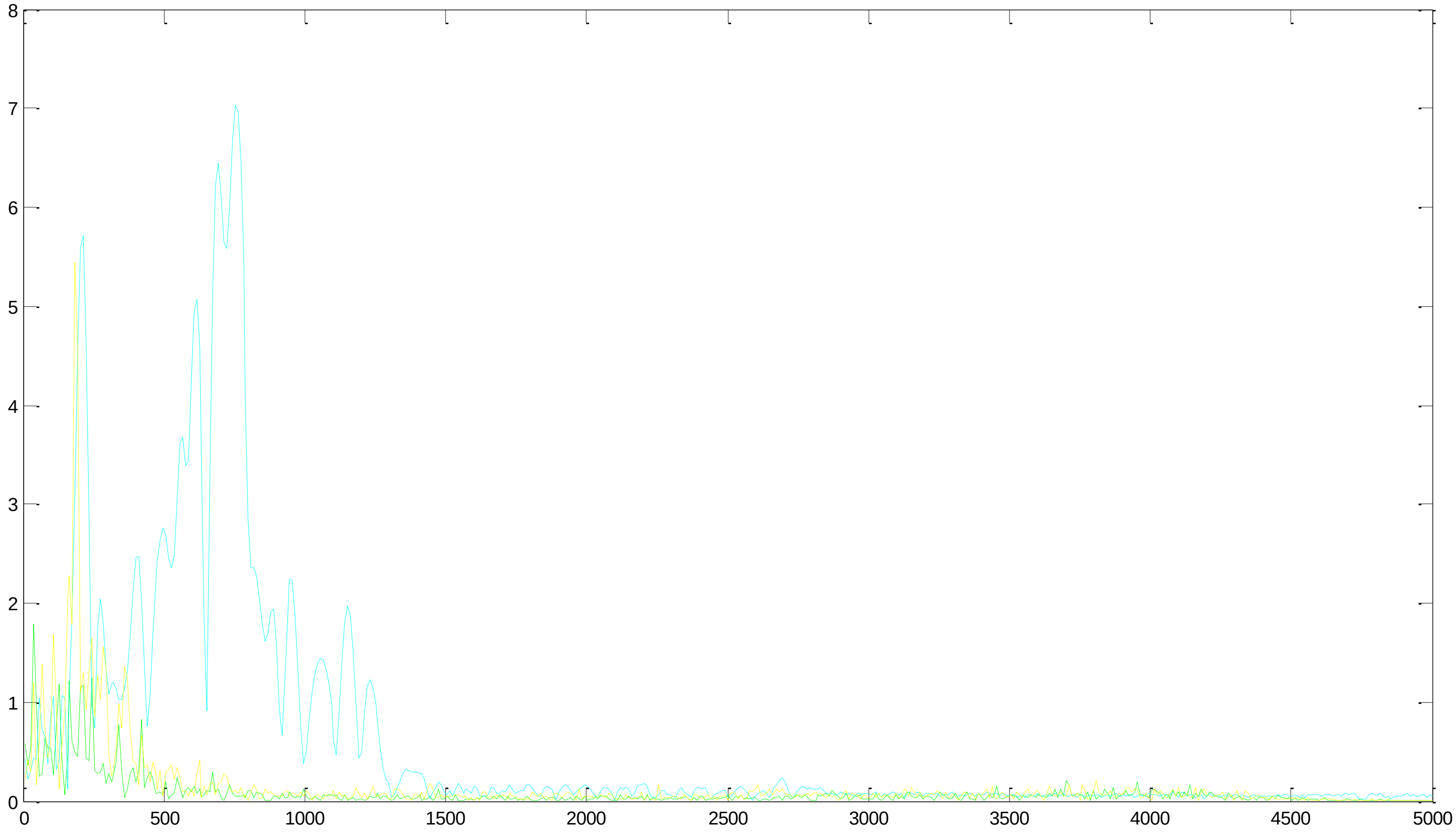
```
svmStruct =

          SupportVectors: [118x6 double]
                   Alpha: [118x1 double]
                    Bias: 0.4089
          KernelFunction: @quadratic_kernel
      KernelFunctionArgs: {}
              GroupNames: [256x1 logical]
    SupportVectorIndices: [118x1 double]
               ScaleData: [1x1 struct]
           FigureHandles: []
```

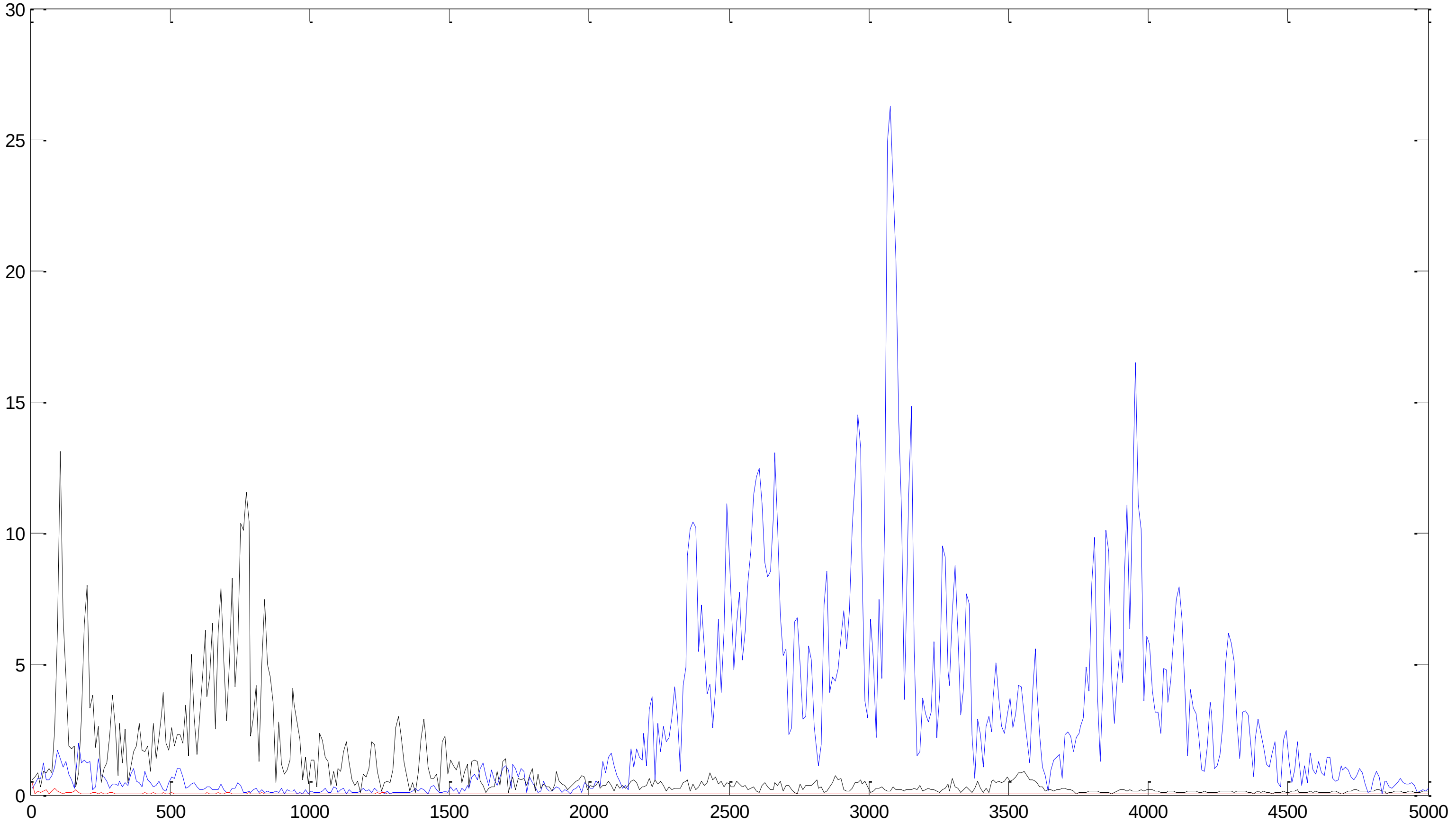
Frequency content of the signal (ba,Male)



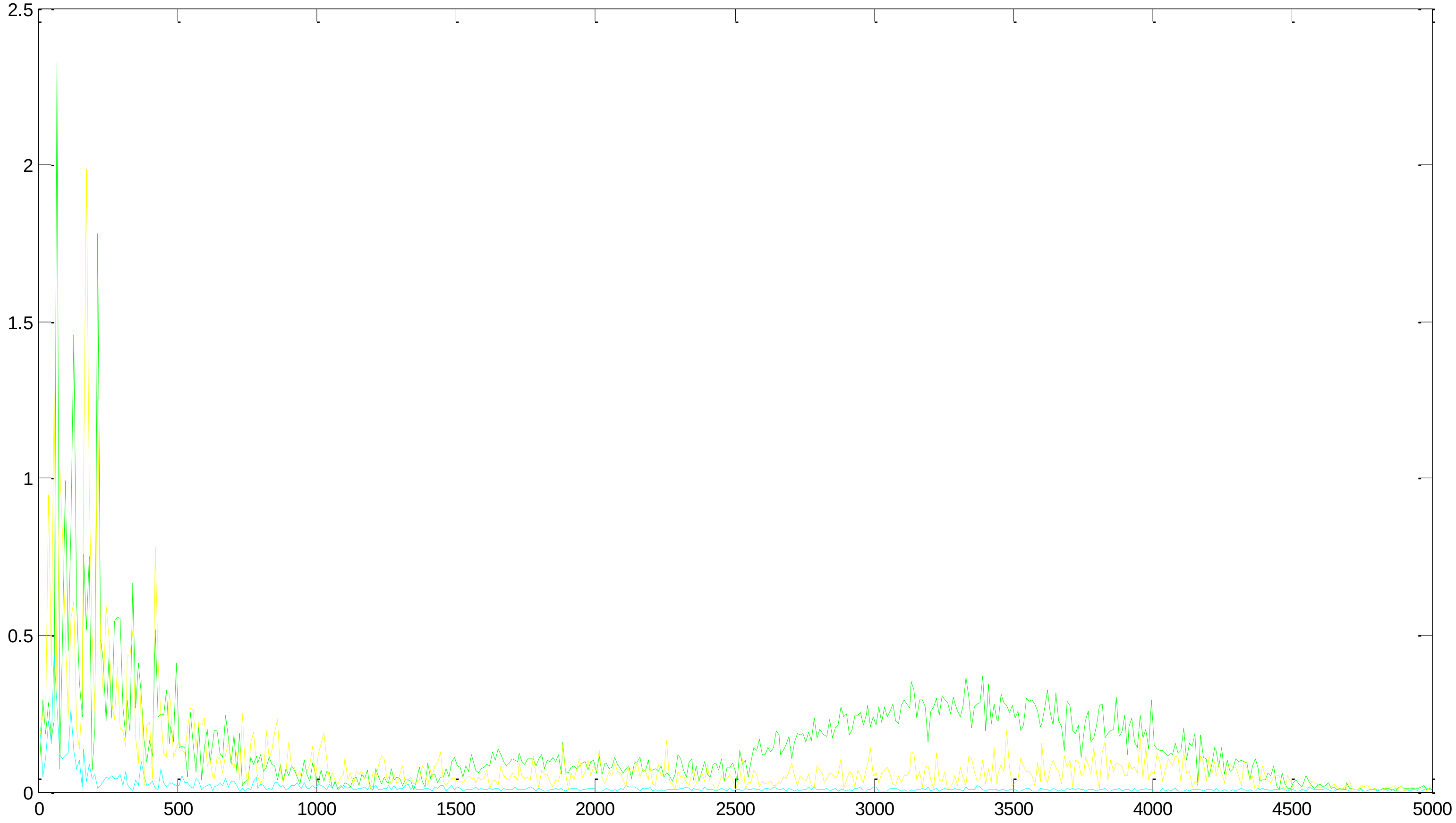
Frequency content of the signal (ba,Female)



Frequency content of the signal (Cha,Male)



Frequency content of the signal (Cha,Female)



```
clear;clc
close all
load data_new.txt
data=data_new;
training_dat=data(:,1:6) ;
training_op=data(:,7);
groups = ismember(training_op,1);

[train, test] = crossvalind('holdOut',groups);

cp = classperf(groups);


svmStruct = svmtrain(training_dat(train,:),groups(train),'showplot',true,'Kernel_Function','quadratic');
title(sprintf('Kernel Function: %s',func2str(svmStruct.KernelFunction)),'interpreter','none');
classes = svmclassify(svmStruct,training_dat(test,:),'showplot',true);
classperf(cp,classes,test)
cp.CorrectRate
figure
svmStruct = svmtrain(training_dat(train,:),groups(train),'showplot',true,'boxconstraint',1e6,'Kernel_Function','quadratic');
classes = svmclassify(svmStruct,training_dat(test,:),'showplot',true);
classperf(cp,classes,test)
cp.CorrectRate
```

```
[aco_data1,sam_rate1,nbits1]=wavread('E:\MTECH2011\SEM4\pattern\database\ap\apya1.wav');
[aco_data2,sam_rate2,nbits2]=wavread('E:\MTECH2011\SEM4\pattern\database\dy\dyya1.wav');
[aco_data3,sam_rate3,nbits3]=wavread('E:\MTECH2011\SEM4\pattern\database\sc\scya1.wav');
[aco_data4,sam_rate4,nbits4]=wavread('E:\MTECH2011\SEM4\pattern\database\jb\jbya1.wav');
[aco_data5,sam_rate5,nbits5]=wavread('E:\MTECH2011\SEM4\pattern\database\lc\lcya1.wav');
[aco_data6,sam_rate6,nbits6]=wavread('E:\MTECH2011\SEM4\pattern\database\lk\lkya1.wav');
puls11=aco_data1(1:4500);
puls12=aco_data2(1:4500);
puls13=aco_data3(1:4500);
puls14=aco_data4(1:4500);
puls15=aco_data5(1:4500);
puls16=aco_data6(1:4500);

puls21=aco_data1(4500:9000);
puls22=aco_data2(4500:9000);
puls23=aco_data3(4500:9000);
puls24=aco_data4(4500:9000);
puls25=aco_data5(4500:9000);
puls26=aco_data6(4500:9000);


% Energy
ST1=sum(puls11.^2)/length(puls11);
ST2=sum(puls12.^2)/length(puls12);
ST3=sum(puls13.^2)/length(puls13);
ST4=sum(puls14.^2)/length(puls14);
ST5=sum(puls15.^2)/length(puls15);
ST6=sum(puls16.^2)/length(puls16);
ST=[ST1 ST2 ST3 ST4 ST5 ST6];

S2T1=sum(puls21.^2)/length(puls21);
S2T2=sum(puls22.^2)/length(puls22);
S2T3=sum(puls23.^2)/length(puls23);
S2T4=sum(puls24.^2)/length(puls24);
S2T5=sum(puls25.^2)/length(puls25);
S2T6=sum(puls26.^2)/length(puls26);
S2T=[S2T1 S2T2 S2T3 S2T4 S2T5 S2T6];

x=puls26;
```

```
%zero crossing
for n=1:length(x)-10
    pu= [ puls11(n) puls12(n) puls13(n) puls14(n) puls15(n) puls16(n)];
    fd=zeros(1,6);
    gh=find(pu >0);
    fd(gh)=1;
    gh1=find(pu<0);
    fd(gh1)=-1;
    gh2=find(pu==0);
    fd(gh2)=0;

    pu= [ puls11(n+1) puls12(n+1) puls13(n+1) puls14(n+1) puls15(n+1) puls16(n+1)];
    ma=zeros(1,6);
    hj=find(pu >0);
    ma(hj)=1;
    hj1=find(pu<0);
    ma(hj1)=-1;
    hj2=find(pu==0);
    ma(hj2)=0;
    z1(n,:)=(fd-ma)/2;

        cha= [ puls21(n) puls22(n) puls23(n) puls24(n) puls25(n) puls26(n)];
    cfd=zeros(1,6);
    cgh=find(cha >0);
    cfd(cgh)=1;
    cgh1=find(cha<0);
    cfd(cgh1)=-1;
    cgh2=find(cha==0);
    cfd(cgh2)=0;

    cha= [ puls21(n+1) puls22(n+1) puls23(n+1) puls24(n+1) puls25(n+1) puls26(n+1)];
    cma=zeros(1,6);
    chj=find(cha >0);
    cma(chj)=1;
    chj1=find(cha<0);
    cma(chj1)=-1;
    chj2=find(cha==0);
    cma(chj2)=0;
    z2(n,:)=(cfd-cma)/2;
end
zcr1=sum(z1)/length(z1)
zcr2=sum(z2)/length(z2)
```

```
% energy entropy
fra_dur=40e-3;
shift=10e-3;
Ng=length(puls11);
L40=round(fra_dur*sam_rate1);
L10=round(shift*sam_rate1);
nf=0;
for ui=1:100:4100
x1=[puls11(ui:ui+L40) puls12(ui:ui+L40) puls13(ui:ui+L40) puls14(ui:ui+L40) puls15(ui:ui+L40) puls16(ui:ui+L40)];
x2=[puls21(ui:ui+L40) puls22(ui:ui+L40) puls23(ui:ui+L40) puls24(ui:ui+L40) puls25(ui:ui+L40) puls26(ui:ui+L40)];
nf=nf+1;
%plot(x1)
f_x1=abs(fft(x1));
f_x2=abs(fft(x2));
sum_ft1=sum(f_x1.^2,1);
sum_ft2=sum(f_x2.^2,1);
p1=f_x1./repmat(sum_ft1,length(f_x1),1);
p2=f_x2./repmat(sum_ft2,length(f_x2),1);
%plot(p)
avg_e1(nf,:)=sum(x1.^2,1);
avg_e2(nf,:)=sum(x2.^2,1);
h1(nf,:)=sum(p1.*log10(p1),1);
h2(nf,:)=sum(p2.*log10(p2),1);
end
Ce1=sum(avg_e1,1)./nf;
Ch1=sum(h1,1)./nf;
Ce2=sum(avg_e2,1)./nf;
Ch2=sum(h2,1)./nf;


M1=(avg_e1-repmat(Ce1,length(avg_e1),1)).*(h1-repmat(Ch1,length(avg_e1),1));
M_av1=mean(M1,1);
EE1=sqrt(1+abs(M_av1))

M2=(avg_e2-repmat(Ce2,length(avg_e2),1)).*(h2-repmat(Ch2,length(avg_e2),1));
M_av2=mean(M2,1);
EE2=sqrt(1+abs(M_av2))

tru_op=repmat([1 1 1 0 0 0],1,2);
% FB=[ ST S2T; EE1 EE2; tru_op].'
```

```
% frequency bias
L=length(puls11);
Fs=sam_rate1;
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
f = Fs/2*linspace(0,1,NFFT/2+1);
sigfft1=fft(puls11,NFFT)/L;
sigfft2=fft(puls12,NFFT)/L;
sigfft3=fft(puls13,NFFT)/L;
sigfft4=fft(puls14,NFFT)/L;
sigfft5=fft(puls15,NFFT)/L;
sigfft6=fft(puls16,NFFT)/L;

[fg msig1]=max(abs(sigfft1(1:NFFT/2+1)));
[fg msig2]=max(abs(sigfft2(1:NFFT/2+1)));
[fg msig3]=max(abs(sigfft3(1:NFFT/2+1)));
[fg msig4]=max(abs(sigfft4(1:NFFT/2+1)));
[fg msig5]=max(abs(sigfft5(1:NFFT/2+1)));
[fg msig6]=max(abs(sigfft6(1:NFFT/2+1)));
msig=[ msig1 msig2 msig3 msig4 msig5 msig6]./sam_rate1;


ftsig1=sum((f'.^2).*abs(sigfft1(1:NFFT/2+1))).^0.5;
ftsig2=sum((f'.^2).*abs(sigfft2(1:NFFT/2+1))).^0.5;
ftsig3=sum((f'.^2).*abs(sigfft3(1:NFFT/2+1))).^0.5;
ftsig4=sum((f'.^2).*abs(sigfft4(1:NFFT/2+1))).^0.5;
ftsig5=sum((f'.^2).*abs(sigfft5(1:NFFT/2+1))).^0.5;
ftsig6=sum((f'.^2).*abs(sigfft6(1:NFFT/2+1))).^0.5;

ftsig=[ftsig1 ftsig2 ftsig3 ftsig4 ftsig5 ftsig6]./1000;

chafft1=fft(puls21,NFFT)/L;
chafft2=fft(puls22,NFFT)/L;
chafft3=fft(puls23,NFFT)/L;
chafft4=fft(puls24,NFFT)/L;
chafft5=fft(puls25,NFFT)/L;
chafft6=fft(puls26,NFFT)/L;


ftcha=[ftcha1 ftcha2 ftcha3 ftcha4 ftcha5 ftcha6]./1000;
```

```
cenftsig1=sum((f').*abs(sigfft1(1:NFFT/2+1)))/sum(abs(sigfft1(1:NFFT/2+1)));
cenftsig2=sum((f').*abs(sigfft2(1:NFFT/2+1)))/sum(abs(sigfft2(1:NFFT/2+1)));
cenftsig3=sum((f').*abs(sigfft3(1:NFFT/2+1)))/sum(abs(sigfft3(1:NFFT/2+1)));
cenftsig4=sum((f').*abs(sigfft4(1:NFFT/2+1)))/sum(abs(sigfft4(1:NFFT/2+1)));
cenftsig5=sum((f').*abs(sigfft5(1:NFFT/2+1)))/sum(abs(sigfft5(1:NFFT/2+1)));
cenftsig6=sum((f').*abs(sigfft6(1:NFFT/2+1)))/sum(abs(sigfft6(1:NFFT/2+1)));

cenftsig=[cenftsig1 cenftsig2 cenftsig3 cenftsig4 cenftsig5 cenftsig6]./1000;

cenftcha1=sum((f').*abs(chafft1(1:NFFT/2+1)))/sum(abs(sigfft1(1:NFFT/2+1)));
cenftcha2=sum((f').*abs(chafft2(1:NFFT/2+1)))/sum(abs(chafft2(1:NFFT/2+1)));
cenftcha3=sum((f').*abs(chafft3(1:NFFT/2+1)))/sum(abs(chafft3(1:NFFT/2+1)));
cenftcha4=sum((f').*abs(chafft4(1:NFFT/2+1)))/sum(abs(chafft4(1:NFFT/2+1)));
cenftcha5=sum((f').*abs(chafft5(1:NFFT/2+1)))/sum(abs(chafft5(1:NFFT/2+1)));
cenftcha6=sum((f').*abs(chafft6(1:NFFT/2+1)))/sum(abs(chafft6(1:NFFT/2+1)));

cenftcha=[cenftcha1 cenftcha2 cenftcha3 cenftcha4 cenftcha5 cenftcha6]./1000;

FB=[ ST S2T; EE1 EE2;zcr1 zcr2; ftsig ftcha; cenftsig cenftcha;msig mcha;  tru_op].'
```