Voice Recognition:

Training

Metrics

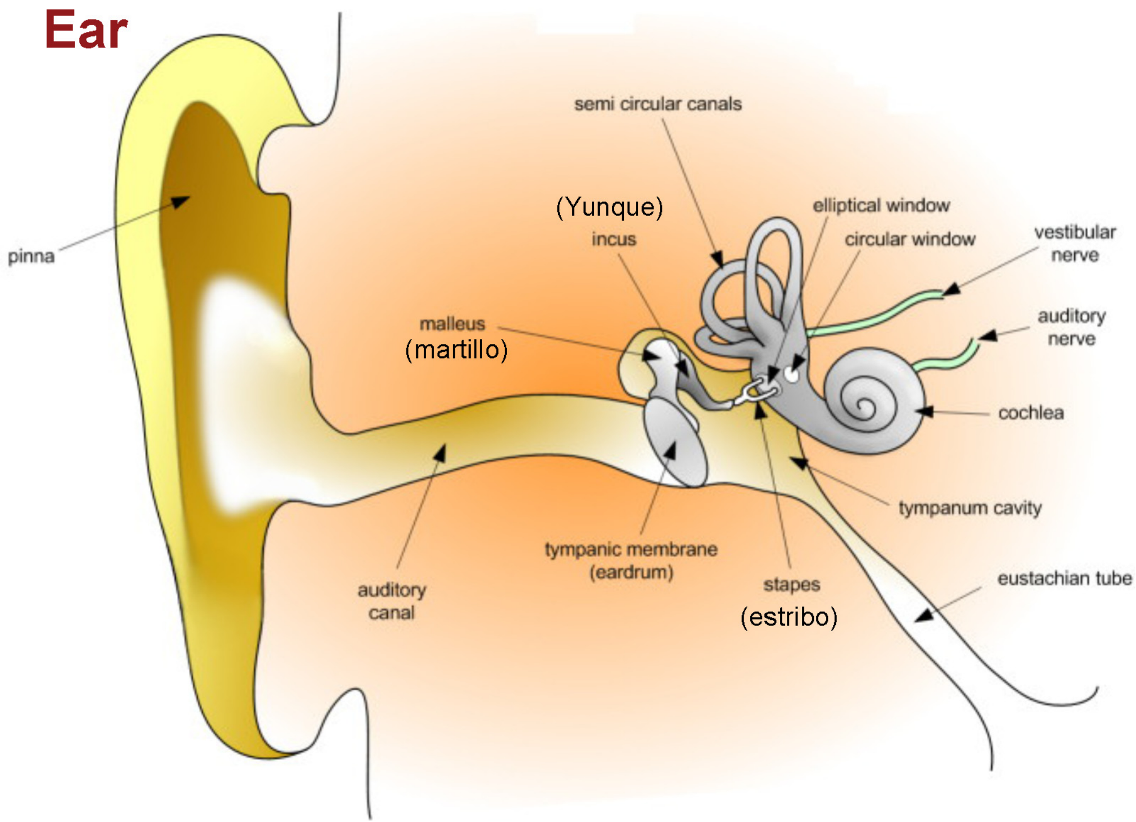
Classification

Performance

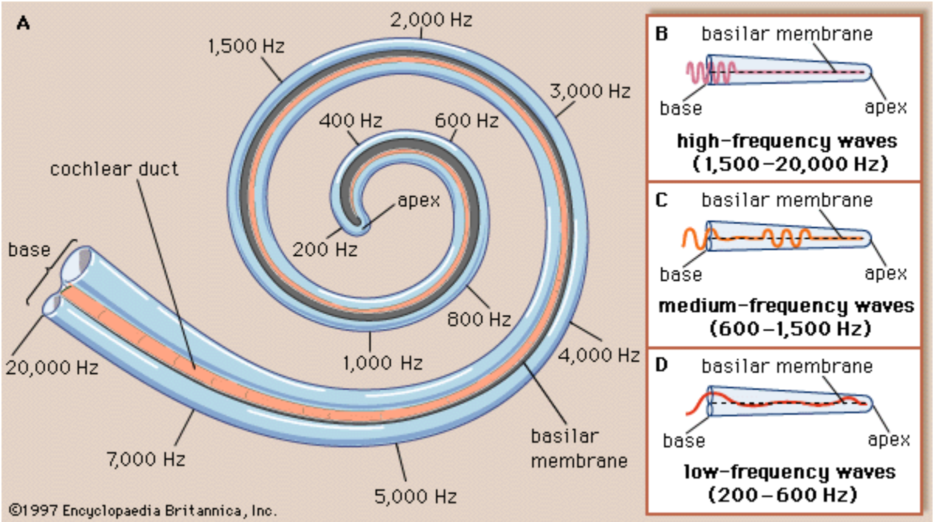
Market Analysis

Goals

- **To develop an understanding of the importance of the *Fourier Spectrum* for developing speech recognition systems.**
- **To develop a speaker-independent *vowel recognition system* to distinguish the 5 long vowel sounds.**
- **To understand the use of a *metric space* for performing classification.**
- **Understand how to *statistically evaluate* a speaker-independent vowel classification system.**
- **To appreciate the relationship between *commercial viability* and *system performance***

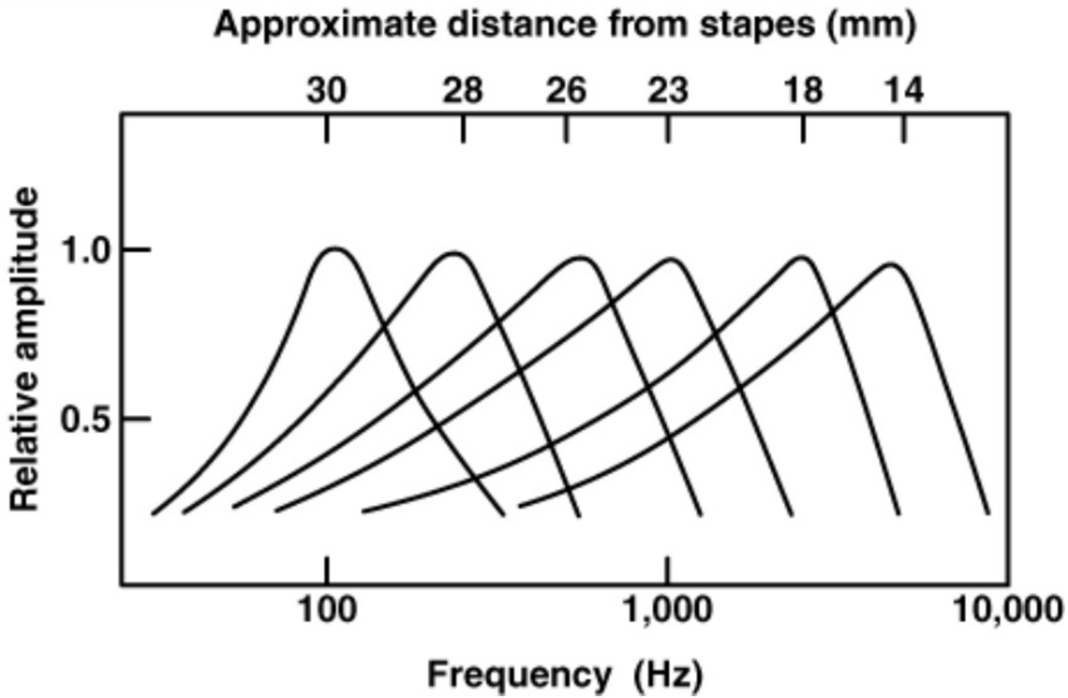

Ear
pinna
semi circular canals
(Yunque)
incus
elliptical window
circular window
vestibular nerve
auditory nerve
malleus
(martillo)
cochlea
tympanum cavity
tympanic membrane (eardrum)
stapes
(estribo)
auditory canal
eustachian tube

The Human Hearing System



- **The human ear perceives sound by measuring energy in different frequency bands. Essentially, we hear in the *frequency domain*!**

- **Speech recognition designers recognize the importance and usefulness of the Fourier Spectrum for constructing speech recognition systems.**



- **The electrical nerve signal from each fiber is filtered using a bandpass filter. The bandwidth of each filter scales according to a logarithmic scale.**

General Approach for Classifying a Vowel Sound

❖ **Classifier Construction and Training**

A. Record 25 vowel sounds for each of the long vowels `'a'`, `'e'`, `'i'`, `'o'`, and `'u'`
   Use the `record_vowels.p` matlab function to create your `vowel_mine.mat` data file.

B. Convert each vowel sound into a metric vector, `E = [E1 , E2 , ... , En]`
   Write the `spectral_band_energy.m` matlab function.

C. Use the metric vector from the training data to build a decision tree
   Requires writing the `classify_vowel.m` matlab function.
   This is the most difficult step and requires analyzing your data
   There is no single solution here. Each group will be different.

UPLOAD TO YOUR ONLINE LIBRARY

❖ **Classification of an unknown vowel sound**

1. Record an unknown vowel sound, represented as an audio signal in matlab `{t,x}`
2. Convert the audio signal into a vector of numbers, `E` (same method as above)
3. Using that vector, `E`, traverse the decision tree to determine which vowel sound was spoken

You will be provided a matlab script to do this called `run_classifier.p`.

For best results, be consistent with your tone of voice for the training data and classification

A. Training Data

❖ Use the `record_vowels.p` function to record long vowel sounds 'a' 'e' 'i' 'o' 'u'

From the matlab command window prompt, type

```
>> record_vowels(25,'vowels_mine');
```

44,100 COL

Ma = 25 ROWS

EACH ROW IS SAMPLES VECTOR FOR 1 VOWEL SOUND

Notes:

- Records 25 long vowel sounds each for `a, e, i, o, u`
- Cleans up some background noise and eliminates transients
- Creates the 5 matrices `Ma, Me, Mi, Mo, Mu` as well as the time vector, `t`
  Each row of the matrix `Ma` is a samples vector (same for the others)
- Automatically saves the data to file `vowels_mine.mat` (you can rename it later if needed)
  This file will only contain the matlab variables: `Ma, Me, Mi, Mo, Mu, t`

  You can reload your vowel data from the command window by using the following command

```
>> load vowels_mine
```

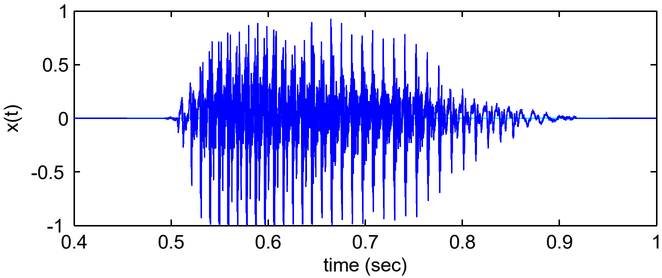
- The vowels can be replayed using the following command:

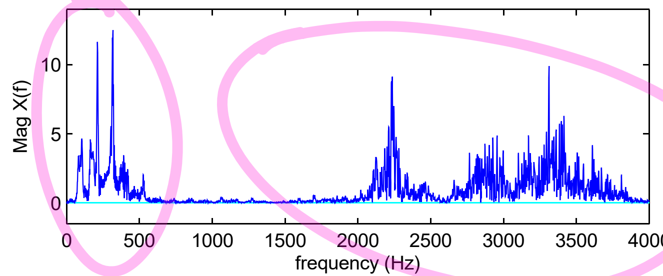
```
>> play_vowels('vowels_mine');
```

❖ Each group member is to record training data and upload their `vowels_mine.mat` file to their online library.

B. Vowel Sounds and The Fourier Spectrum

Long E Vowel Sound

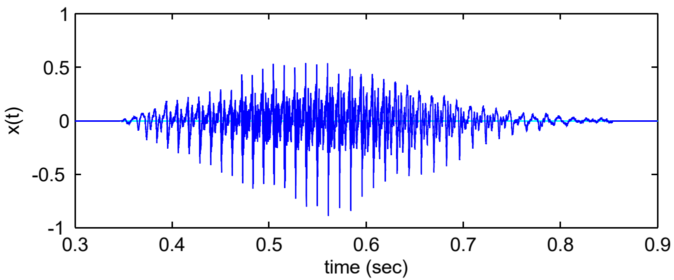


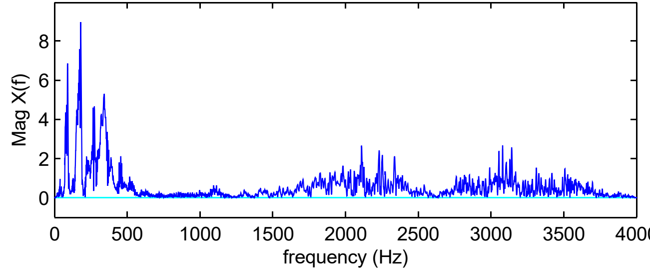


SPEAKER CHARACTERISTICS

VOWEL SOUNDS

Long U Vowel Sound





B. Converting a Vowel Sound to a Metric Vector

❖ Steps

- Calculate the Fourier Transform {`f,Xf`} of the vowel sound defined by vectors {`t,x`}

```
[f,Xf] = myFT(t,x);     % does NOT plot the spectrum
Xmag = abs(Xf);         % Determine the spectral magnitude
[f,Xf] = myFT(t,x,'plot','frange',[200 6000]);  % Plots the spectrum
```

- Calculate the spectral energy over a set of frequency bands

$$E_{f_1,f_2} = \int_{f_1}^{f_2} |X(f)|^2 df$$

```
fbnd = [f1 f2 ... fN];
E(1) = intdef(f,Xmag.*Xmag,fbnd(1),fbnd(2));
E(2) = intdef(f,Xmag.*Xmag,fbnd(2),fbnd(3));
E(N-1) = intdef(f,Xmag.*Xmag,fbnd(N-1),fbnd(N));

E = E/sum(E);    % Normalize the energy vector so the sum is unity
```

B. The `spectral_band_energy` *matlab function*

- **Write a function with the following form:**

```
%--------------------------------------------------
% Put appropriate header here...
%--------------------------------------------------
function E = spectral_band_energy(t,x,fbnd)

% write code to construct all the spectral band energies for the
% appropriate frequency range and with the appropriate width.
%
% Keep code within 80 columns.
%

    E = ...   (this should have a length of one less than fbnd)

end
```

- **Notes**
  - **Make sure to normalize the spectral energy vector `E`**
  - **Make sure `t` and `x` are the same length. If not then default the output to empty.**
  - **Make sure `fbnd` has length of at least `2`, if not then default the output to empty.**
  - **Be sure to GRADE this function after uploading to your online library**

You will want to explore the vowels across your entire group to find the best classifer.

- **Generate a combined set of vowel sounds across the entire group**
  - **Each group member should generate their own vowel sounds into file `vowels_mine.mat`**
  - **Each group member should upload their vowels to their own online library**
  - **The `vowels_mine.mat` files should be renamed to make them unique, perhaps use**
    `vowels_mine1.mat` , `vowels_mine2.mat` , `vowels_mine3.mat`
  - **Combine all the vowels using the following matlab function**

```
>> combine_vowels('vowels_mine1','vowels_mine2','vowels_mine3');
```

CAN PUT IN 4TH INPUT

This will generate the file `vowels_group.mat` which contains the vowels for all the group members

  - **Each group member should upload `vowels_group.mat` to ~~their online library~~.**

- **The vowel classifier should be built using the full collection of vowels across the group and found in file `vowels_group.mat`**

DON'T UPLOAD

C. *The `fbnd` Vector*

- **Start with the `fbnd` vector found in matlab data file `fbnd.mat`**

```
>> load fbnd;
```

You will note that the bands are spaced logarithmically across the frequency axis.

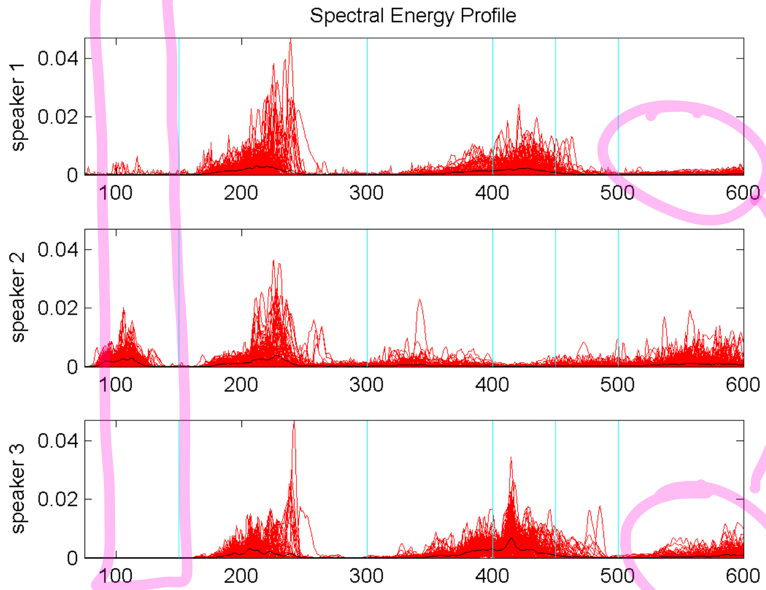
- **Use the following function to show the lines on a spectral magnitude plot**

Make the plot containing the spectral magnitudes the active matlab plot, then type

```
>> fbnd_lines(fbnd);
```

C. Exploring the Spectral Energy by Speaker

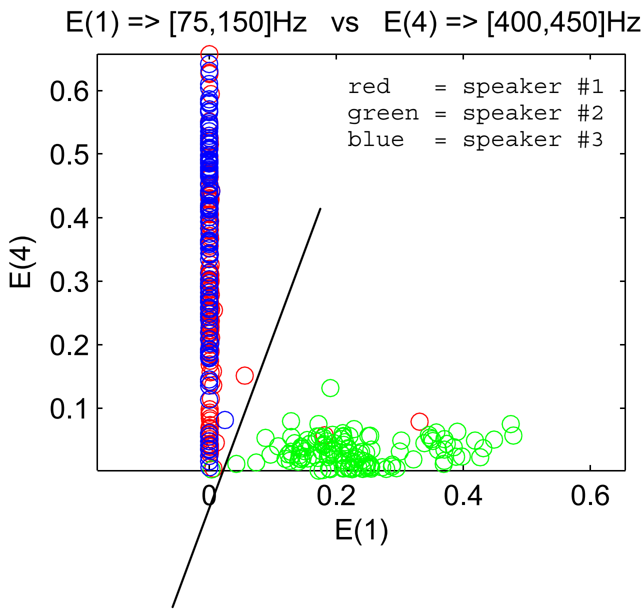

```
energy_profile_speaker(fbnd,tau,file1,file2,file3[,file4]);

fbnd  - vector defining the frequency bands
tau   - smoothening parameter over [0,100], generally use tau = 10
file1 - Vowel data file name for group member #1 (vowel_mine1.mat)
file2 - Vowel data file name for group member #2 (vowel_mine2.mat)
file3 - Vowel data file name for group member #3 (vowel_mine3.mat)
```



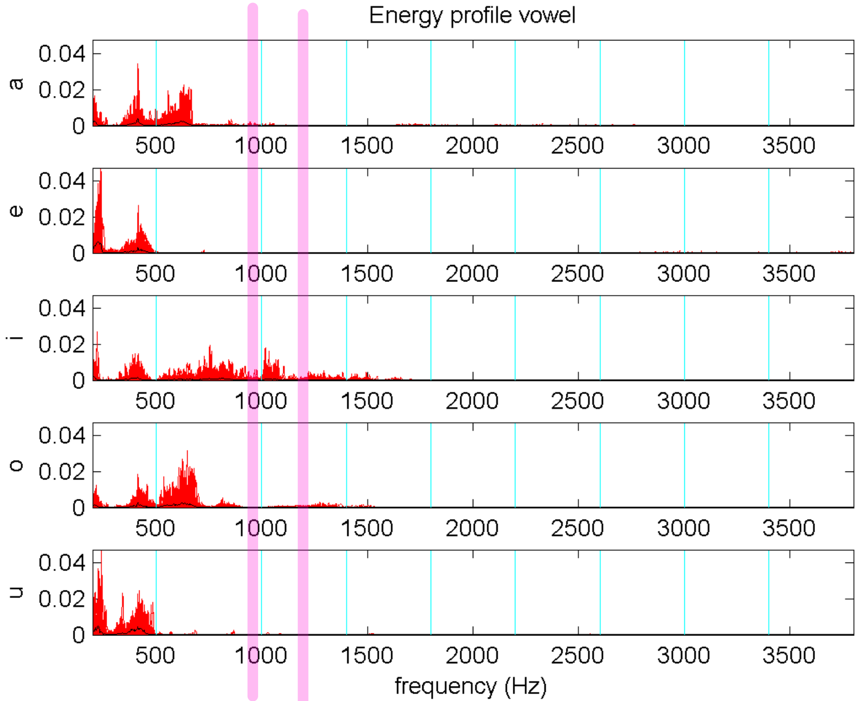
- **Select TWO components of the `E` vector, such as**
  - `E(1)` **which corresponds to the energy in the frequency band** `[fbnd(1),fbnd(2)]Hz`
  - `E(4)` **which corresponds to the energy in the frequency band** `[fbnd(4) fbnd(5)]Hz`

- **Use** `Metric_speaker_plot.p` **to plot all the vowels for each speaker for** `E(1)` **vs** `E(4)`

```
metric_speaker_plot(fbnd,1,4,file1,file2,file3[,file4]);
```



```
energy_profile_vowel(fbnd,tau,'vowels_group');

fbnd  - vector defining the frequency bands
tau   - smoothening parameter over [0,100], generally use tau = 10
```



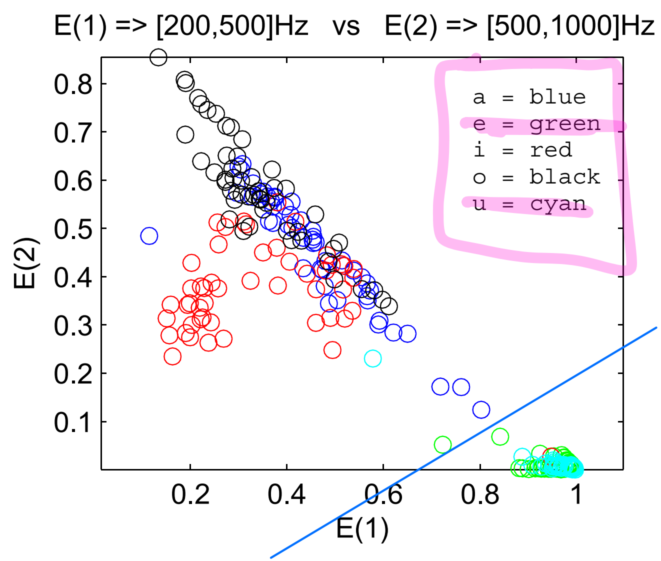
- **Select TWO components of the `E` vector, such as**
  - **`E(2)` which corresponds to the energy in the frequency band `[fbnd(2),fbnd(3)]Hz`**
  - **`E(4)` which corresponds to the energy in the frequency band `[fbnd(4) fbnd(5)]Hz`**

- **Use `Metric_vowel_plot.p` to plot each vowel across all speakers for `E(2)` vs `E(4)`**

```
metric_vowel_plot(fbnd,I0,I1,filegroup,'aeiou');
```

If the `'aeiou'` string is changed to `'aou'`, then only those three vowels are plotted.

If it is omitted entirely, then all 5 vowels are plotted.



Suggestions

❖ **For speaker classification, use frequencies below 1000Hz**

Look carefully at the *spectral energy profile* across each speaker to see where to place frequency band boundaries.

You can use the following function to cycle through pairs of metrics

```
metric_speaker_cycle(fbnd,'vowel1','vowel2','vowel3');
```

```
fbnd     - vector defining the frequency bands
'vowel1' - vowel data filename for group member #1
'vowel2' - vowel data filename for group member #2 (optional)
'vowel3' - vowel data filename for group member #3 (optional)
```

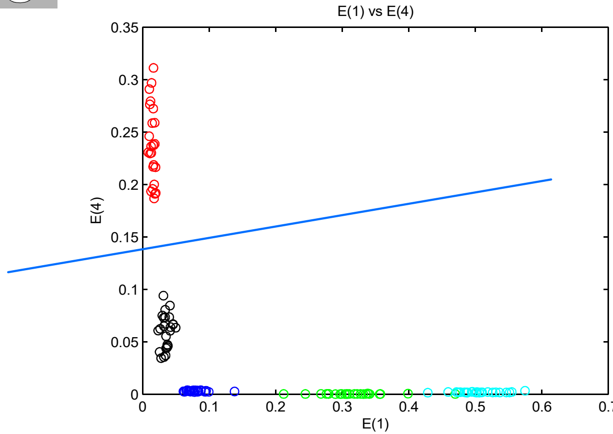
❖ **For vowel classification, use frequencies above 500Hz**

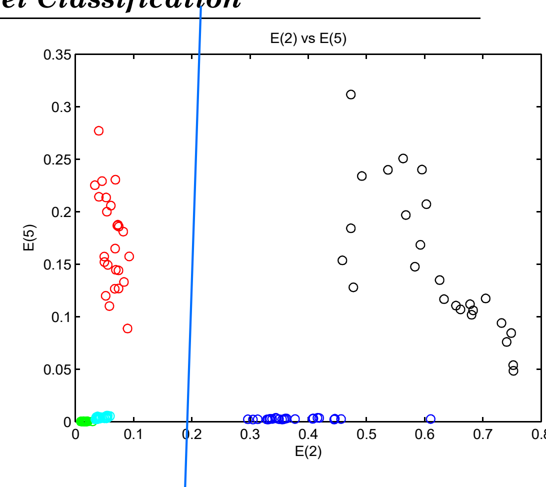
Look carefully at the *spectral energy profile* across each vowel to see where to place frequency band boundaries.

You can use the following function to cycle through pairs of metrics

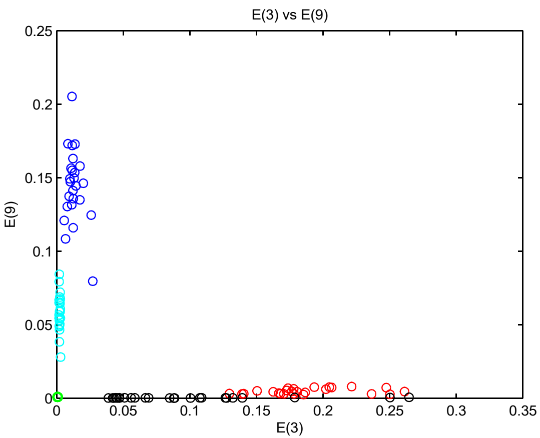
```
metric_vowels_cycle(fbnd,'vowels_group',vstr);
```

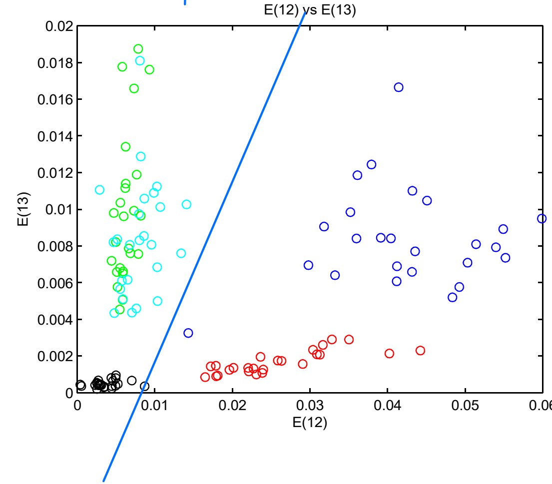
```
fbnd     - vector defining the frequency bands
'vowels_group' - vowel data filename for the group (or subgroup)
vstr - String containing the vowels to plot, vstr = 'aeiou' will plot all the
       vowels, vstr = 'aiu' will only plot those vowels.  The default is to
       plot all the vowels.
```



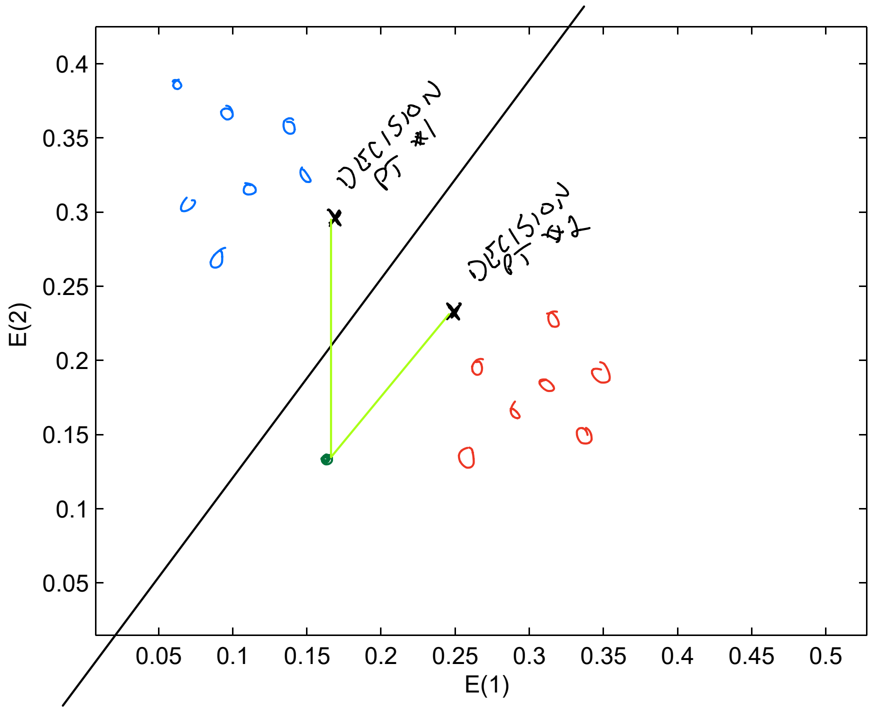


a = blue
e = green
i = red
o = black
u = cyan

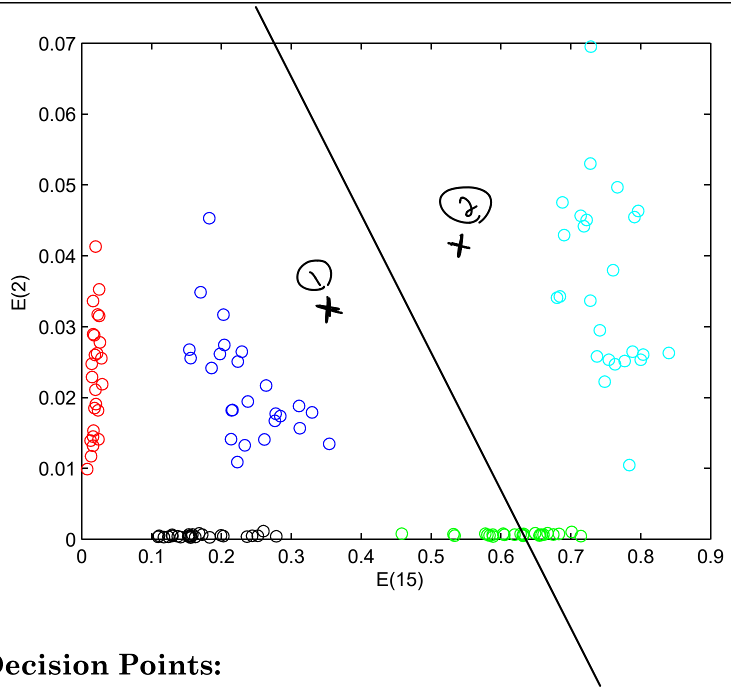




Nearest Neighbor Classification: 2 Points & 1 Boundary

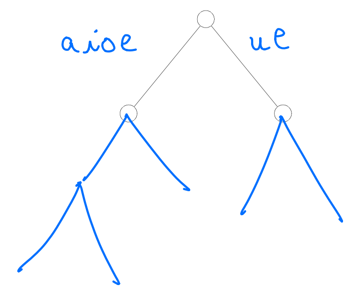


Nearest Neighbor Classification



a = blue
e = green
i = red
o = black
u = cyan

Tree Representation



- **Decision Points:**
  - **Only 2 are allowed per node (it can be different for each node)**
  - **Each point is associated with one vowel sound**
  - **Given unknown vowel with metric vector m, find the decision point closest in distance.**
- **Decision Boundary - Illustrates how points in space are divided using a nearest neighbor rule**

❖ **Write a function with the following form:**

```
%-----------------------------------------------------
% Put appropriate header here...
%-----------------------------------------------------
function [indx,dmin] = nearest_neighbor(m,decV)

% write code to find which decision vector, given by decV, the metric
% vector m is closest to.  The output indx, is the row index of decV that
% corresponds to the closest vector. The value dmin is the distance between
% m and the closest decision vector in decV.
%
% Keep code within 80 columns.
%

    indx = ...
    dmin = ...

end
```

$m = [\ \xi(1)\ \ \xi(4)\ ]$

$decV = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ DECISION PT #1 / DECISION PT #2

❖ **Notes**

- **Make sure that** `decV` **is a 2x2 matrix and that the dimension of** `m` **is** `2`**.**
- **Be sure to GRADE this function after uploading to your online library**

Selecting Decision Points

- **Restrictions:**
  - **Only `2` metrics are allowed per node, i.e. `2` components of `E`**
  - **For each decision node, only `2` decision points are allowed (ONE boundary)**

- **You can use the matlab function `select_decision_pts` to find decision points**
  - **First generate a figure that contains training data like those shown in the previous slides use `metric_vowel_plot.p` or `metric_speaker_plot.p`**
  - **Run the matlab function as follows:**

```
decV = select_decision_pts;
```

This program is interactive and requires entering single character commands.

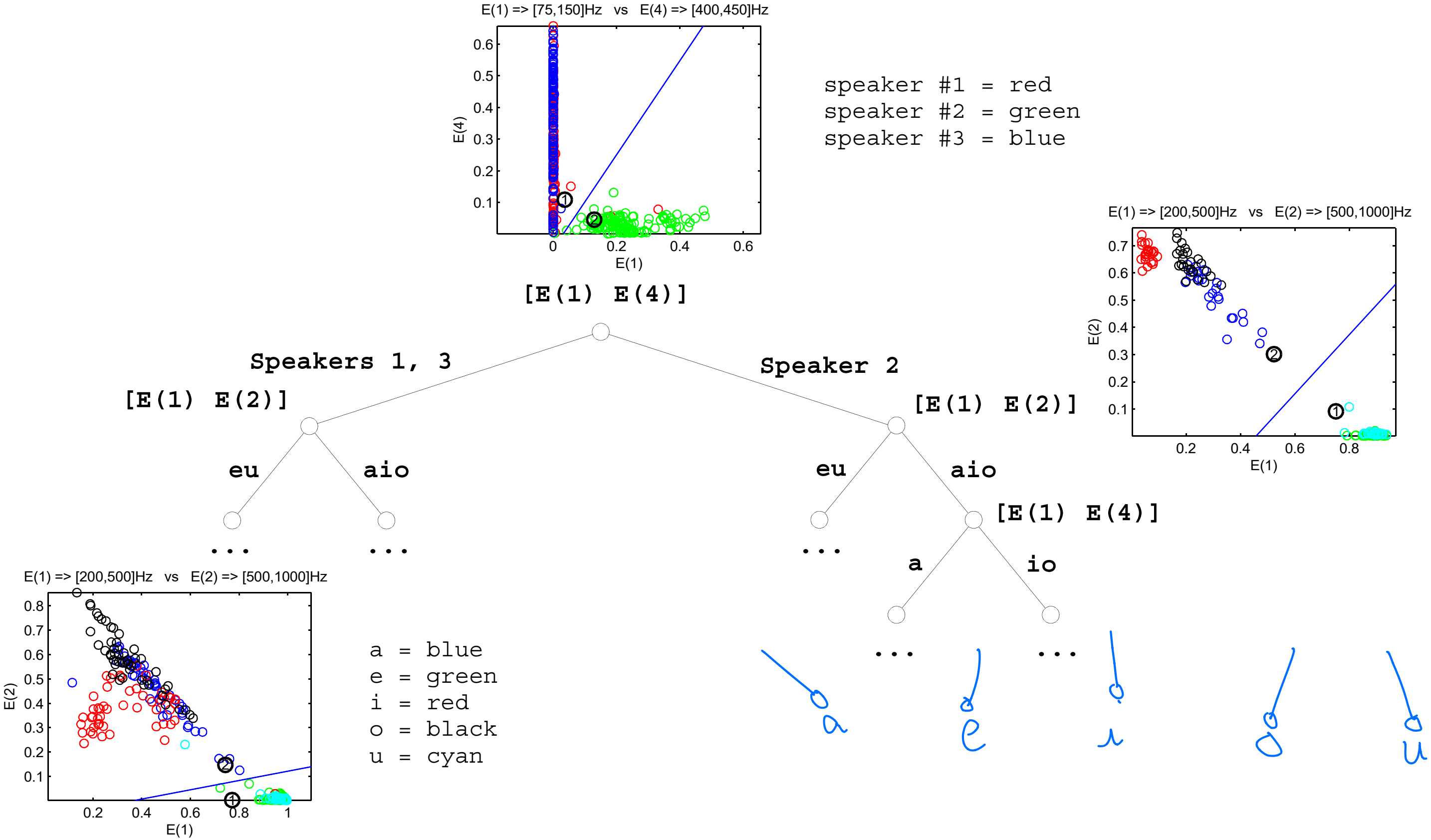
The output `decV` will be a `2x2` matrix. Decision pt #1 is `decV(1,:)` and pt #2 is `decV(2,:)`

The program can also be called with an input vector to show the decision boundary

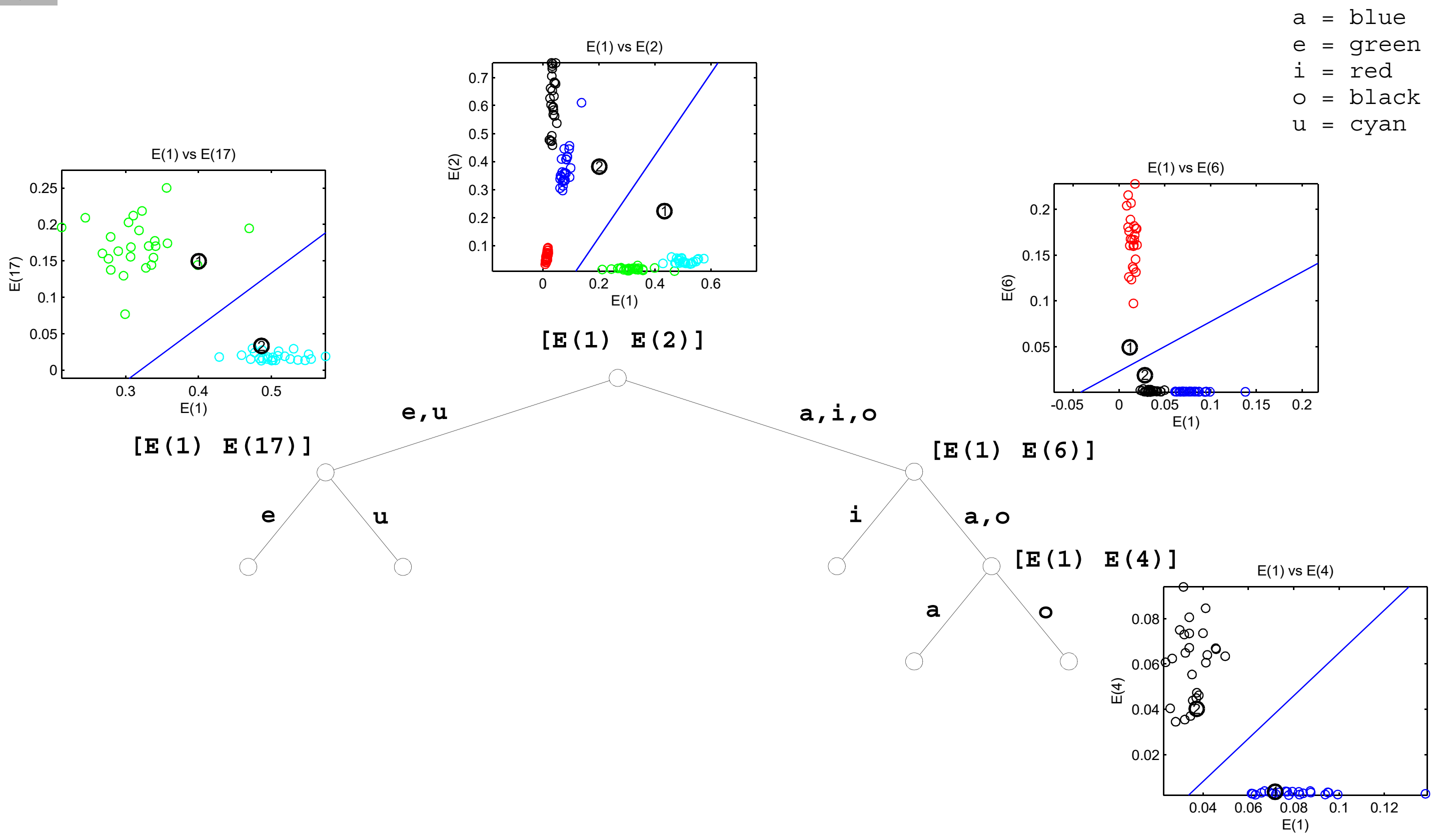
```
decV = select_decision_pts(decV);
```

In either case, this function will set axis to `equal` so the decision lines will look natural.

The Multi-Level Classifier Tree: Separating Speakers First



The Multi-Level Classifier Tree: Separating Vowels First



- **Write a function with the following form:**

```
%-----------------------------------------------
% Put appropriate header here...
%-----------------------------------------------
function Vchar = classify_vowel(t,x)

% write code to implement the full classifier.  That is, it takes in a single
% vowel sounddefined by t and x, computes the metrics for that vowel,
% then traverses the classifer tree.  At each node, it finds the metrics
% using the appropriate indexes into E and then uses nearest_neighbor.m to
% make a decision. The vowel associations with each branch are defined in
% this function. The output Vchar is a single character string such as 'a' or
% 'e' that identifies the spoken vowel. Hardcode your decision vectors.
%
% Note:  The training data should NOT be used in this function, only your
%        frequency bands and decision points need to be defined and used.
%
   Vchar = ...
end
```

IF, ELSE, END

"a"
"e"

HARDCODE DECISION PTS

- **Notes**
  - **Make sure the dimensions of t and x are consistent**
  - **This function is NOT graded directly, but you will evaluate it later by how well it performs on your vowel data.**

- Record an additional 15 set of vowels, `a e i o u`, for each group member for evaluation
- Collect these vowel sounds into a single `vowels_data.mat` matlab data file.
- Upload the `vowels_data.mat` to your online library (same file for each group member)
- Write a script that classifies all these new vowels and stores the results in a confusion matrix

  `Lab11_test.m`

  - Use your `classify_vowel.m` function to classify each vowel sound
  - The *Confusion Matrix* takes the following form

INDIVIDUAL GROUP MEMBER

ESTIMATE

TRUTH	a	e	i	o	u
a	68	2	3	1	1
e					
i					
o					
u					

SUM OF VALUES SHOULD BE 75 FOR 3-PERSON GROUP

FOR TRAINING DATA

Runtime Classifier Script

- **Use the matlab script**

```
run_classifier.m
```

to see a real-time long vowel sound speech recognition system at work!

This requires that all of your required functions have been written and are working.

Market Analysis

- **Who is your target market?**
  - **Identify the people in your group and where they're from and define an appropriate market**
  - **Your market should include countries and/or parts of countries that speak similarly to the group**

- **What is the possible market size?**
  - **How many people are in your target market world wide? Gather data from the internet and make a best guess.**
  - **How many of those people have the financial means to buy your system? Gather data from the internet and make a best guess.** CHOOSE A PRODUCT COST

- **Suppose a full speech recognition system performed according to the performance on your long vowel sounds.**
  - **How well would your system be received?**
  - **For each part of your target market, discuss how well your system would be received?**

You must look up at least 3 references to support your data and conclusions!

- **The vowel data uploaded to your website.**

  `vowels_mine.mat` (individual training data, 25 sounds/vowel for YOU only)
  ~~`vowels_group.mat` (training data for the ENTIRE group~~)
  `vowels_data.mat` (additional data, ~~ENTIRE group~~, 15 sounds/vowel/~~person~~)

  ONE GROUP MEMBER

- **The following functions uploaded to your online library**

  `spectral_band_energy.m` (function, requires grading)
  `nearest_neighbor.m` (function, requires grading)
  `classify_vowel.m` (function, graded ???)

- **The project report uploaded to your online library with name**

  `Lab11_report.---`

- **Fill out the `eval` link on the website after the report is uploaded.**

- **Write a project report described by:**
  - **No more than 3 pages in length (not including the title page and figures)**
  - **Has the format described on the website**
  - **Uses proper english and grammer (have someone outside the group read it if necessary)**
  - **Has the following parts**

Title page with the name of each group member, lab #, class, etc. (separate page)

I. INTRODUCTION (introduce speech recognition in general)

II. RESULTS (answer the following questions)

A. Draw a fully descriptive diagram of your classifier tree. This needs to include the branch structure, the metric vector indexes used at each node, the vowels that are associated with each decision, as well as a figure that shows the metric space and decision line for each node. Explain the process that the group went through to end up with this classifier tree design. Explain any difficulties the group had in separating vowels.

B. Show the classifier results by showing the confusion matrix calculated on the training data across all students (25 sounds/vowel/student) and show a 2nd confusion matrix for the classifier data across all students (15 sounds/vowel/student). Explain in words how well the classifier worked. Were there any consistent errors? Were certain vowels more difficult to distinguish. Clearly explain. Did the training data confusion matrix look like the confusion matrix for the additional vowel sounds? If not, conjecture as to why.

C. In this section, answer the questions given in the Market Analysis slide from above.

III. CONCLUSIONS (summarize your final results and discuss them)

IV. BIBLIOGRAPHY (list all references used including books and websites)

The functions will be graded according to the grade rubrics given on the website. The percentage breakdown for the grade metrics for this lab are given below.

Each matlab function will be graded according to:

(D) Documentation	**30%**
(L) Logic and Efficiency	**30%**
(R) Results	**40%**

The project report will be graded according to:

Grammar, spelling, punctuation	**30%**
Completeness	**30%**
Solution approach	**40%**

Groups

- **This lab will be worked in groups of 3 (generally).**
- **Your group members have been chosen for you. Please look on the website for your group members.**
- **All project deliverables must be uploaded to all student online libraries.**
  The SAME deliverables should be uploaded for each group member EXCEPT for the data in the file `vowels_mine.mat`. 15 V/PERSON

  VOWELS_DATA.MAT 15 V/PERSON

- **Each group should strive to:**
  - **Make every effort to listen to one another so the best ideas are brought up and discussed**
  - **Make sure both group members are actively participating in the project**
  - **Make sure that the group uses their time wisely**
  - **Make sure that the group has a plan to get the work done in the time allowed .**