A Speech Recognition Linear Systems Lab

Dr. William Joseph Ebel Sr., Saint Louis University

Dr. William J. Ebel received his Ph.D. degree from the University of Missouri - Rolla in 1991 in Electrical Engineering. He joined St. Louis University in the Fall of 2000 as an Associate Professor of Electrical Engineering and has served as the department Chairman for two different periods. He teaches in the areas of signal and image processing, communications, and robotics and he is actively involved in novel instructional methods. His research interests include image processing and robotics.

Dr. Chris Carroll, Saint Louis University

Dr. Carroll is an Assistant Professor and the Civil Engineering Program Coordinator in Parks College of Engineering, Aviation and Technology at Saint Louis University. His experimental research interests focus on reinforced and prestressed concrete, while his engineering education research interests focus on experiential learning at both the university and K-12 levels. Dr. Carroll is the chair of ACI Committee S802 - Teaching Methods and Educational Materials and he has been formally engaged in K-12 engineering education for nearly ten years.

A Speech Recognition Linear Systems Lab

Introduction

This this paper describes a Linear Systems laboratory project that involves designing a simplified speech recognition system to recognize the 5 long vowel sounds for a team of 3 or 4 students. This project is assigned soon after the student has been introduced to the Fourier Transform in the associated Linear Systems lecture course. This paper describes the Laboratory project by illustrating the solution with a specific example drawn from real data for a single student team.

This laboratory project has the primary goals:

1. Understand the importance of the *Fourier Spectrum* for developing useful signal analysis algorithms and systems.
2. Develop a speaker-independent *vowel classification system* to distinguish the 5 long vowel sounds for a team of students.
3. Use a *metric space* and *support vectors* to perform classification.
4. *Statistically evaluate* a speaker-independent vowel classification system.
5. Appreciate the relationship between *commercial viability* and *system performance*.

These goals reinforce concepts in the associated Linear Systems course, expand student learning to classifiers, provide a setting for the student to analyze and interpret data, provide a setting for students to learn how to evaluate a system, and lead students to appreciate the broader issues of system design such as commercial impact.

This laboratory project is introduced to the student by reviewing the basic physiology of the human hearing system [2]. The audio pressure wave enters the auditory canal and impinges on the tympanic membrane which then vibrates the malleus and causes an auditory excitation in the cochlea. The cochlea is an organ that contains filaments that resonate in different frequency bands. These resonant filaments are attached to nerves which carry the signal strength to the brain for processing. It is essential to convey the fact that humans hear in the frequency domain. Since humans are clearly suited for recognizing speech, formulating an auditory signal as a Fourier Spectrum is natural. This establishes the relevance of the Fourier Transform for developing a speech classification system.

Commercial speech recognition systems use sophisticated algorithms that process the Fourier Spectrum of an auditory signal to classify human speech [3]. The algorithms built by students in this laboratory project are much more simplified due to the limited student background and time constraints. However, the algorithms they develop still provide robust classification for such a limited data set and achieve the goal of experiencing the relevance of the Fourier Spectrum to an engineering application.

This is a first semester Junior year laboratory (1 credit hour) with a typical enrollment of 25 students. The associated lecture is a typical 3 credit hour course covering signals and systems concepts such as the Fourier Transform, Laplace Transform, Z-Transform and applications. This laboratory project spans three weeks of class time with work generally required outside of laboratory time. Matlab [4] is the chosen programming environment. Teams of either 3 or 4 students are assigned by the instructor with the goal of achieving as much diversity as possible for each team. Since engineering students typically come from all parts of the world and the female population is increasing, achieving diversity is fairly easy. The specific team requirements for this laboratory project are

1. Record two data sets of the long vowel sounds a, e, i, o, and u, by each team member, one for training and one for independent performance assessment.
2. Write 3 matlab functions: **`spectral_band_energy`**, **`nearest_neighbor`**, and **`classify_vowel`**.
3. Perform a market analysis of the final system that includes system performance issues.
4. Write a report detailing the results of the laboratory project.
5. Fill out a team survey individually by each team member.

Students work in teams to complete these requirements and decide, as a team, how these requirements will be carried out. The difficulty of this laboratory project along with the limited time constraints require that each team self organize in order to agree on how decisions will be made, devise a solution strategy that includes individual work components, assign work components to individual team members, and to otherwise function like an engineering design team.

This paper includes a description of the nature of the classifier, a discussion of the expectations regarding data analysis and interpretation, a discussion of the performance results and performance assessment, and a discussion of the relationship to the ABET Student Outcomes of various aspects of this laboratory project.

The Classifier

The classifier can be visualized as a tree diagram, containing nodes and branches, with possibly one or more team members or one or more vowels associated with each branch. The end nodes are referred to as leaves and are associated with a single vowel. Classifying a single vowel sound involves traversing down the tree until a leaf is reached which is associated with a single vowel.

In order to develop the classifier tree, the team collects two data sets, one for training, referred to as the *training data*, and one for independent classification, referred to as the *independent data*. Each student records a 1 second audio clip of a long vowel sound, 25 times for each of the long vowel sounds a, e, i, o, and u. For a 3-person team, this results in 75 long a vowel sounds, and a total of 375 vowels sounds for the training data. Once collected, the training data is used to build the classifier tree. In addition, each team member records an additional 15 vowel sounds for each vowel for a total of 75 vowel sounds per team member and 225 vowel sounds across a 3-person team for the independent data. These are used to measure the classifier performance independent of the training data.

The nature of the classifier tree is entirely up to the design team and can take many different forms. For example, it is possible for the team to decide to use a node to separate out a speaker first, and then to build a vowel classifier tree for that speaker only and placed below the appropriate branch. Or a team can decide to directly separate vowels for the entire team. Both these tactics have been applied by past student teams.

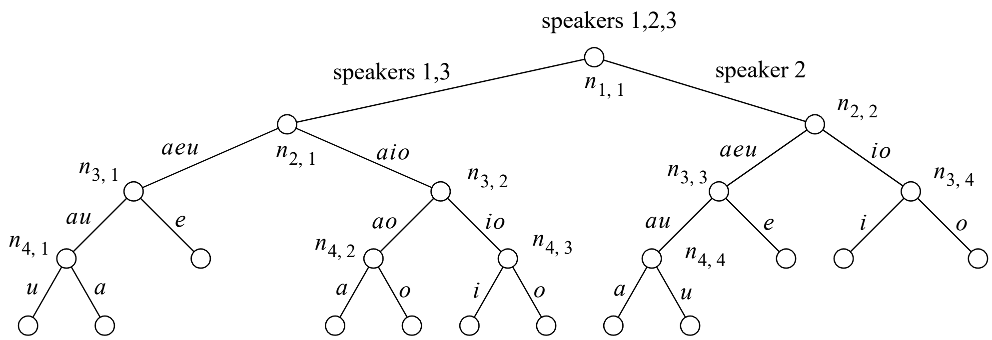

Figure 1. Example classifier tree for a 3-student team.

The classifier for each team is embodied in the matlab function **`classify_vowel`** that implements a decision tree like that shown in Figure 1. The node $n_{i,j}$ refers to a level i node and the j th node from the left. In order to implement a robust classifier tree, the training data may indicate that it is best to separate speaker #2 from speakers #1 and #3 as shown. In practice, the classifier does not know which speaker a vowel came from, however if the classifier tree is well designed, a vowel from speaker #2 will take the speaker #2 branch from node $n_{1,1}$. This example also illustrates the various ways the vowels can be separated. For example, node $n_{2,1}$ separates the vowels aeu with the vowels aio so that vowel a is present in both branches leaving that node.

To make a decision for vowel sound $x(t)$ at node $n_{2,1}$ requires finding the metric vector $m_{2,1}$ and carrying out a nearest-neighbor decision rule. The metric vector is a subset of 2 components of the normalized spectral energy vector E_x/E_s with E_x components given by

$$E_x(k) = \int_{f(k)}^{f(k+1)} |X(f)|^2 df \tag{1}$$

where $k = 1, 2, 3, 4, 5, 6$. The integral limits are defined by the frequency band vector

$$f = [501, 708, 1000, 1413, 1995, 2818, 3981] \tag{2}$$

which defines $f(1)$ through $f(7)$. The energy vector sum $E_s = E_1 + E_2 + E_3 + E_4 + E_5 + E_6$ is a normalizing constant so the vector E_x/E_s has unity sum to eliminate variations due to volume. The integration bounds are defined by a frequency boundary vector, such as

$$f = [501, 708, 1000, 1413, 1995, 2818, 3981] \tag{3}$$

which defines $f(1)$ through $f(7)$ to form 6 frequency band intervals.

The training data for speakers #1 and #3 are plotted in Figure 2 (lower right), referred to as a *metric space*. The metric space contains points using a metric vector drawn from 2-components of the spectral energy vector for each vowel and each team member. This metric space demonstrates that a single decision line can be drawn to separate the aeu vowels from the aio vowels for speakers #1 and #3. The decision line can be implemented with two decision vectors, $d_{2,1,aeu}$ associated with aeu, and $d_{2,1,aio}$ associated with aio. The decision line is the perpendicular bisector of the line joining the two decision vectors. The decision is made with a *nearest neighbor* rule based on the distance of $m_{2,1}$ relative to each decision vector.

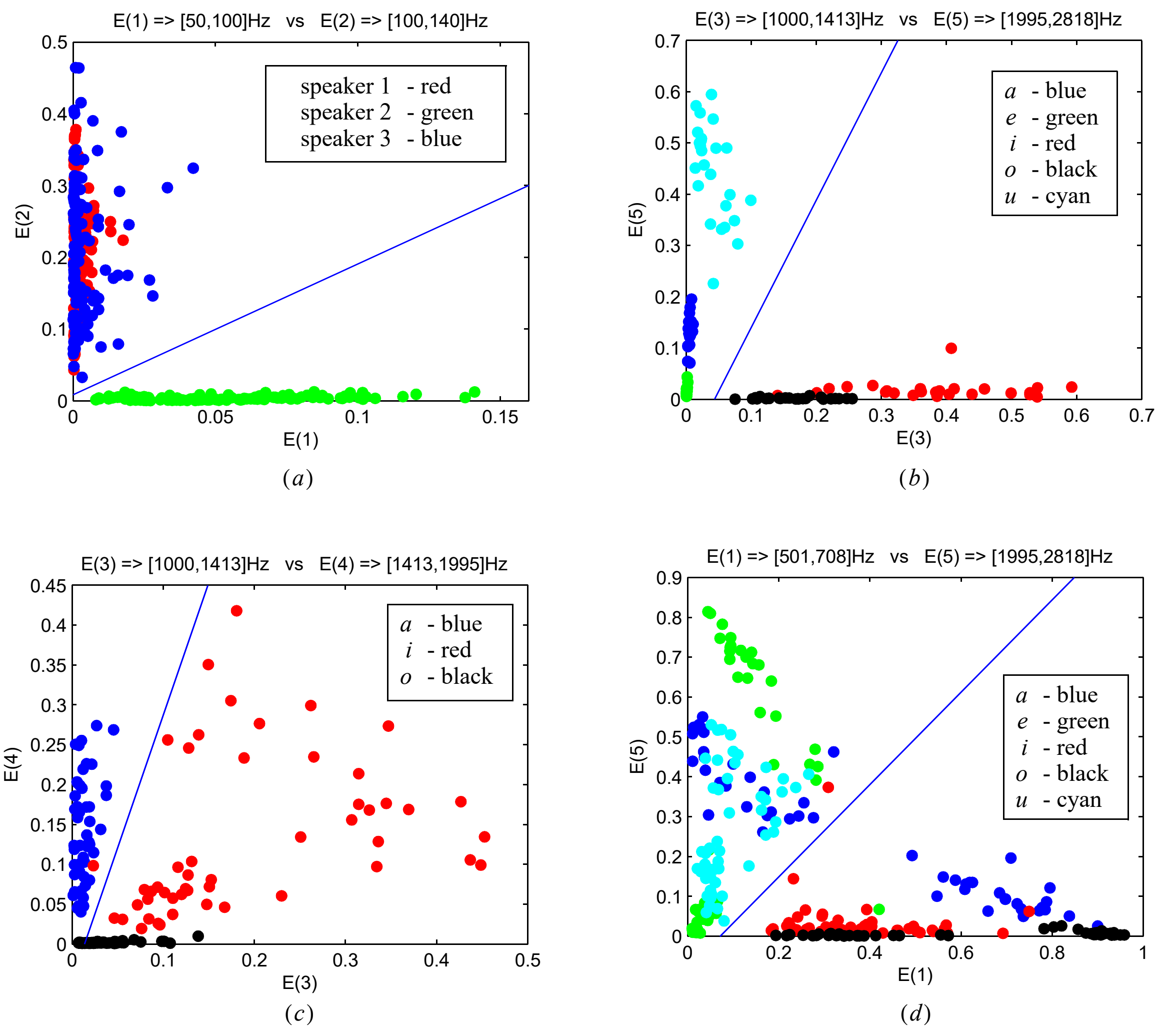

Figure 2. Example metric spaces for (a) node $n_{1,1}$, (b) node $n_{2,2}$, (c) node $n_{3,2}$, and (d) node $n_{2,1}$ (lower right).

Metric spaces for several other nodes in the classifier tree are also shown in Figure 2. Each one of these shows a decision boundary that separates the speakers/vowels into two groups, though not always perfectly. For example, the metric space for node $n_{3,2}$ shows one i vowel on the wrong side of the decision boundary. For the data used to generate the results for this example, all the nodes from $n_{2,1}$ and below that node, as well as node $n_{2,2}$ use the frequency band vector given in Equation (2). The node $n_{1,1}$ uses the frequency boundary vector $f = [50, 100, 140, 220, 260, 300]$ and all other nodes use the frequency boundary vector $f = [1.1, 1.4, 2, 2.5, 2.7, 3, 3.3, 3.9, 4]\text{KHz}$.

This classifier example gives context for the subsequent sections that discuss how students are expected to perform data analysis and interpretation and also how their results are evaluated.

Data Acquisition and Analysis

Each team collects their vowel data as described in the previous section and then analyzes their data to determine both the frequency boundary vector and the specific energy components used to construct the metric vectors. This analysis can be daunting considering the free-form nature of the data and the sheer quantity of data to work with. In order to provide a setting where students can intuitively interpret the data, all metric spaces are required to be 2-dimensional and are only allowed to use a single decision boundary in the form of a line. The 2-dimensional metric spaces are easy to plot and easy to inspect for the purpose of separating vowels.

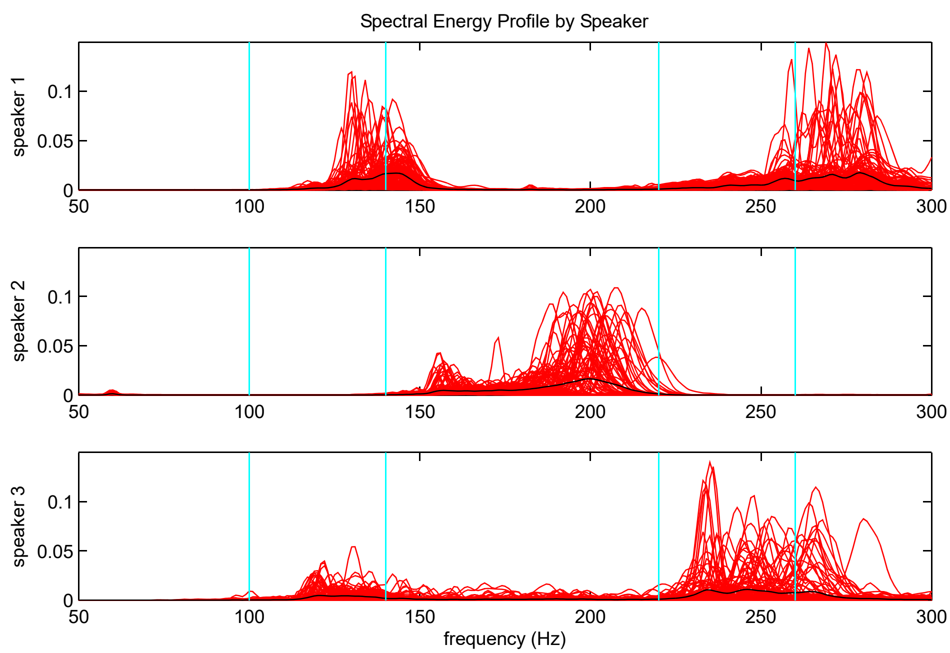

Figure 3. Spectral energy profile by speaker across all vowels. The vertical bars represent the frequency boundaries from the f_b vector.

The laboratory project goal of performing data analysis supersedes the goal of demonstrating an ability to write matlab code, i.e. a tool of engineering practice. Therefore, a set of tools are provided by the instructor (first author) to help the teams "see through" the data. The first is a function that provides a visual representation of the spectral energy profile by speaker across all vowels as shown in Figure 3. This function has the form

```
energy_profile_speaker(fbnd,file1,file2,file3);
```

where **`fbnd`** is the frequency boundary vector, shown as vertical lines on the plots, and where **`file1`**, **`file2`**, and **`file3`** are the data file names for each speaker containing all the vowels. These plots demonstrate that across all vowels, there are clear distinguishing spectral characteristics for the various speakers at low frequencies below 300Hz. The first two frequency bands were chosen to separate speaker #2 from speakers #1 and #3. However, the second two frequency bands also provides good discrimination between speaker #2 and speakers #1 and #3.

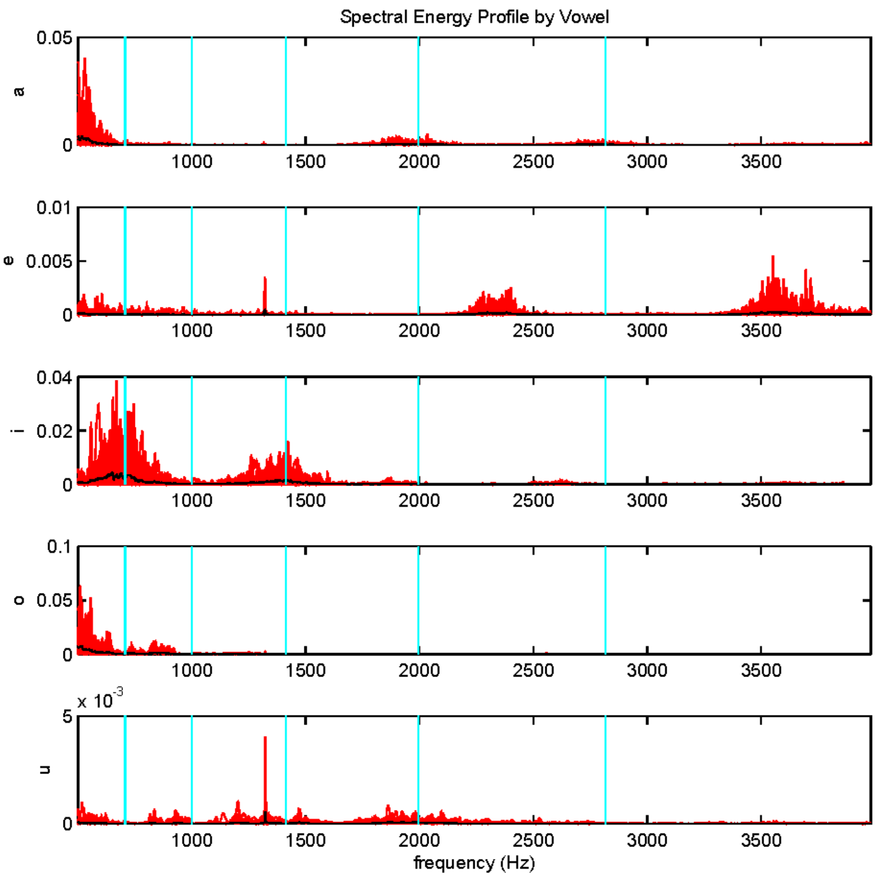

Figure 4. Spectral energy profile by vowel across speakers #1 and #3. The vertical bars represent the frequency boundaries from the f_b vector.

Another tool provided by the instructor creates a plot of the spectral energy profile across, for example, speakers #1 and #3 for each vowel as shown in Figure 4. A review of this plot provides insight into which frequency bands might better distinguish one vowel from another. For example, the vowel e contains more energy in frequencies around 3.5KHz than 2KHz, but vowel u is the opposite. This function has form

```
energy_profile_vowel(fbnd,file1,file2);
```

where **`fbnd`** is the frequency boundary vector, shown as vertical lines on the plots, and where `file1` and `file2` are the data files for the speakers. This function is written so the number of data files can vary from 1 through 4 by simply extending the number of arguments.

These two tools are primarily used to provide a visual setting to set the frequency boundary vector for distinguishing speakers or vowels. The frequency boundary vector is used to construct the spectral energy vector for a vowel.

The next step is to choose the metric vector from the spectral energy vector for distinguishing speakers or vowels. To distinguish speakers, it is helpful to cycle through all possible combinations of spectral energy pairs. A function is provided to help and has the form

```
metric_speaker_cycle(fbnd,file1,file2,file3);
```

where **fbnd**, **file1**, **file2**, and **file3** are the same as those defined for the **energy_profile_speaker** function. If there are N components in the **fbnd** vector, then there are $N-1$ components in the spectral energy vector and $(N-1)(N-2)$ possible 2-component metric spaces. For example, if $N = 6$, then there are 20 pairs of spectral energy components so there are 20 metric space plots to consider. Each plot can be inspected and then with a simple keyboard press the next plot is displayed. The plots looked like that shown in Figure 2(a) but without the decision boundary included.

Visualizing the actual metric space plots gives insight into the degree of clustering of the vowel points for different speakers. Choosing a particular metric space involves selecting clusters that are well separated from each other and also are tightly clustered to some degree. In addition, since each metric space decision boundary is only allowed to be a straight line, the two clusters must be separable by a single line. These restrictions make it easier to select a particular metric space.

The final step is to choose a metric space for each node of the classifier tree. The process of building the classifier tree and choosing the metric space go hand-in-hand. A matlab function is provided to help with these decisions and takes the form

```
metric_vowel_cycle(fbnd,vstr,file1,file2);
```

where **fbnd** is the frequency boundary vector, **file1** and **file2** are the data files for the speakers. The number of speaker data files can vary from 1 through 4 by simply extending the number of arguments provided to the function. The input **vstr** is a vowel string and takes the form of **'aeo'** to only plot the vowels aeo in the metric space. If there are N components in the **fbnd** vector, then there are $N-1$ components in the spectral energy vector and $(N-1)(N-2)$ possible 2-component metric spaces. The metric space plots looked like those shown in Figure 2 but without the decision boundaries included. This function is often used repeatedly to help sort out which metric spaces can best distinguish between various vowels leading to a suitable classifier tree.

The next step is to select the decision boundaries and the decision points. Again, this can be a daunting task considering the number of points in each metric space. To help with this effort, a matlab function is provided by the instructor with the form

```
decV = select_decision_pts()
```

where the output **decV** is a 2×2 matrix with the first row defining the first decision vector and the 2nd row defining the 2nd decision vector. This function operates on the currently active matlab figure. It is interactive and allows students to place decision points with a mouse click right on the matlab plot. When a second point is placed on the plot, the decision boundary automatically appears. The intuitive nature of placing points on an existing matlab plot makes this effort fairly simple. At this point, a team will have established the decision tree, all frequency boundary vectors used throughout the tree, and will have established the decision points that define the decision boundary for each metric space.

Although it appears that much of the matlab programming for this project are provided by the instructor, each team still has to write 3 matlab functions

```
E = spectral_band_energy(t,x,fbnd);
```

where **t** is a time vector, **x** is the samples vector for a single vowel sound, and **fbnd** is the frequency boundary vector. This function needs to perform the calculation given in Equation (1). The second function takes the form

```
index = nearest_neighbor(m,decV);
```

where **m** is a 2-component metric vector from a vowel sound and **decV** is a 2×2 matrix containing two decision vectors corresponding to row 1 and row 2. This function determines which decision vector is closest, in Euclidean distance, to the metric vector and outputs the corresponding row index of **decV**. The third function takes the form

```
vchar = classify_vowel(t,x);
```

where **t** and **x** define an audio signal of a single vowel sound and **vchar** is a single character of one of the 5 vowels. This function implements the full decision tree and can be rather complex. It consists of a series of calculations followed by conditional statements to traverse the decision tree.

The data analysis, interpretation, and decision tree development described in this section demonstrates that the 2nd and 3rd project goals are established.

Performance Results and Assessment

The decision tree given in Figure 1 resulted from applying the data analysis procedure discussed in this paper to real student data for a team of 3 students. The metric spaces shown in Figure 2 are some of the actual metric spaces used to implement the decision tree. The performance of the classifier developed by each team is determined using a *confusion matrix*. The confusion matrix is a representation of the classified vowels to demonstrate how well the classifier worked. For the example classifier discussed in this paper, the confusion matrix for the training data is shown in

Figure 5. For the training data, each row will sum to 75 corresponding to the 75 vowels collected across the 3-person team. For the a vowel, 73 of those 75 classified correctly to a, however 2 of those vowels classified to u. The correct classification percentage for each vowel is shown to the right of the confusion matrix.

	a	e	i	o	u	
a	73	0	0	0	2	97%
e	0	74	1	0	0	99%
i	1	1	72	1	0	96%
o	0	0	0	75	0	100%
u	3	0	0	0	72	96%

(a)

	a	e	i	o	u	
a	37	3	1	2	2	82%
e	1	44	0	0	0	98%
i	2	2	41	0	0	91%
o	0	0	0	45	0	100%
u	11	0	2	0	32	71%

(b)

Figure 5. Confusion matrix results for the classifier tree given in Figure 1 for (a) the training data and (b) the independent data.

Clearly, the training data performed very well with the u vowel having the worst performance at 96% correct classification. A review of student performance results over several years suggests that the a and u vowels tend to get confused and that the o vowel generally performs the best. The independent data performed worst in general. The greatest change was in the u vowel which classified to the a vowel 11 times. The o vowel still classified the best at 100%. When introducing this laboratory project, the students are not told when to collect each of the data sets, only that they need to be collected. Teams typically collect the training data the day the laboratory project is introduced but they wait until at least a week later before collecting the independent data. Later, they often recognize that by waiting, there can be variations in their voice depending upon a number of conditions such as a difference in background noise or perhaps a difference in their health, such as having a cold when recording one data set and not having a cold when recording the other data set.

Besides assessing the performance of the classifier on the team data as a collective group as well as the independent data as a collective group, each team is also required to assess the performance of the classifier on each team member's data individually. This helps to determine if the classifier performed more poorly for one team member's data relative to the others, which may impact the market analysis requirement of this laboratory project discussed in the next section. For the data used in the example given in this paper, the training data performed well for all vowels and also for each team member's vowels individually. For the independent data, speakers #1 and #2 performed very well but speaker #3 did not perform very well and accounts for most of the decreased performance for the independent data across the group. For example, of the 11 u vowels that were incorrectly classified as vowel a, 10 of those resulted from speaker #3.

An open question remains as to how well the team performed in developing their classifier. This question is important as a factor in assigning a grade and also a factor in assessing ABET Student Outcome (b). Since the data is unique to each group, it is difficult to know if the data simply doesn't lend itself to distinguishing the long vowel sounds across the group very well, especially when considering some of the more prominent accents of international students.

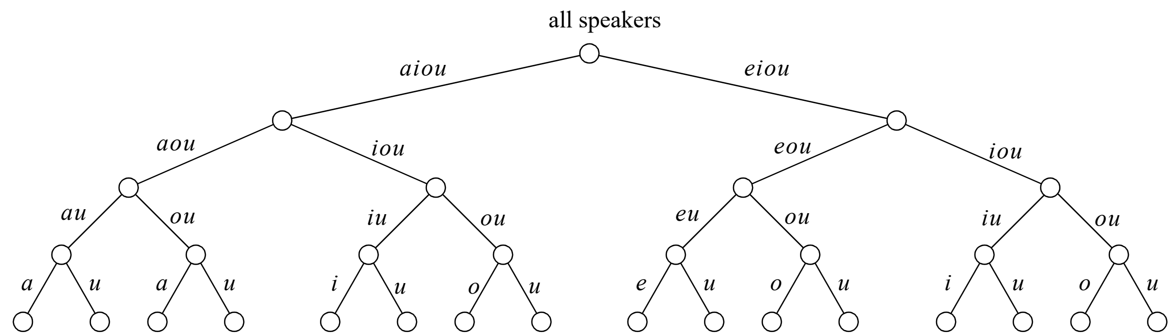

Figure 6. General classifier tree for the automatically generated classifier.

To address this problem, the instructor (first author) developed a program that generates a comparison classifier automatically. The classifier tree always takes the same form as shown in Figure 6. For this classifier, the speakers are not separated. Basically, the classifier tree is designed to separate two vowels at each node with all other vowels carried along both branches. A frequency boundary vector is chosen with logarithmically-spaced frequencies spanning the range from 100Hz to 4000Hz with 12 components. Using this, the classifier generates a single spectral energy vector for each vowel in the training data, generates the metric spaces, and then automatically inspects each metric space for the separation of all possible combinations of vowel pairs. For each metric space and each combination of vowel pairs, a single point is chosen from each vowel cluster to be the decision point for that cluster so that those two decision points lead to the best classification of those two vowels. Since a number of metric spaces can be used to separate a single pair of vowels, a *separation factor* is also calculated for each metric space and pair of vowels to be the square of the distance between the two decision points divided by the product of the standard deviations of each vowel cluster. The metric space selected for separating two vowels is the one with the largest separation factor. The automatically-generated classifier takes about 2 minutes to execute on a typical 3GHz computer for a 3-person team of data.

	a	*e*	*i*	*o*	*u*	
a	75	0	0	0	0	100%
e	0	70	0	1	4	93%
i	0	0	73	1	1	97%
o	0	0	0	75	0	100%
u	1	6	0	0	68	91%

(*a*)

	a	*e*	*i*	*o*	*u*	
a	26	8	0	1	10	58%
e	0	43	1	0	1	96%
i	0	1	44	0	0	98%
o	0	0	0	40	5	89%
u	1	10	0	0	34	76%

(*b*)

Figure 7. Confusion matrix results for automatically generated classifier for (a) the training data and (b) the independent data.

For the data used in the example given in this paper, the automatically generated classifier was applied and gave the results shown in Figure 7. Comparing these results with those shown in Figure 5 indicates that although the percentage correct classification results show different numbers, the overall scale and range of the numbers are comparable except for vowel a for the independent data which shows a significant decrease in performance. However, the goal of the comparison solution is simply to answer the question of whether a team demonstrated a reasonable ability to analyze and interpret data to support ABET Student Outcome (b).

For comparison purposes and to give an indication of how well students performed in general, the comparison solution was generated for student data over three different semesters as shown in Figure 8. The following table gives the percentage correct classification for student training data when applied to their **`classify_vowel`** algorithm as well as the comparison solution. For some vowels, the comparison solution was better and for some vowels the student solution was better. But a few cases, such as group #2 for Fall 2016 and vowel u, suggests that this team could have better analyzed their data. Another example is group #4 for Fall 2016 and vowel u which showed a marked difference in the achievable performance given by the comparison solution.

Fall 2016 - student team solution

	grp 1	grp 2	grp 3	grp 4	grp 5
a	94%	83%	93%	100%	96%
e	94%	91%	99%	99%	100%
i	100%	92%	97%	100%	100%
o	96%	84%	99%	91%	100%
u	96%	48%	96%	25%	100%

Fall 2016 - comparison solution

	grp 1	grp 2	grp 3	grp 4	grp 5
a	94%	93%	97%	96%	100%
e	100%	99%	91%	84%	100%
i	88%	99%	87%	100%	100%
o	96%	99%	99%	100%	99%
u	94%	81%	97%	88%	96%

Fall 2017 - student team solution

	grp 1	grp 2	grp 3	grp 4	grp 5
a	53%	67%	100%	48%	80%
e	75%	65%	100%	92%	98%
i	96%	100%	100%	97%	100%
o	89%	100%	99%	95%	96%
u	75%	97%	63%	94%	98%

Fall 2017 - comparison solution

	grp 1	grp 2	grp 3	grp 4	grp 5
a	84%	100%	99%	89%	100%
e	85%	89%	99%	100%	100%
i	80%	99%	91%	88%	100%
o	92%	99%	99%	75%	100%
u	96%	93%	95%	96%	100%

Fall 2018 - student team solution

	grp 1	grp 2	grp 3	grp 4	grp 5
a	96%	100%	95%	100%	97%
e	95%	95%	85%	91%	90%
i	64%	99%	88%	99%	90%
o	97%	100%	95%	100%	99%
u	93%	95%	83%	100%	88%

Fall 2018 - comparison solution

	grp 1	grp 2	grp 3	grp 4	grp 5
a	89%	100%	88%	97%	93%
e	100%	99%	87%	100%	85%
i	89%	95%	89%	99%	92%
o	96%	93%	88%	100%	94%
u	87%	91%	80%	100%	98%

Figure 8. Percent correct classification for the training data of five teams from three different semesters.

Relationship to ABET Outcomes

The ABET Student Outcomes (SO) that this particular laboratory project addresses are (a), (d), (e), (h), and (k) [1]. The SO (a) refers to the ability to apply knowledge of mathematics, science, and engineering. This project requires that students are applying the Fourier Spectrum to the creation of the spectral energy vector as evidenced by their writing of the **`spectral_band_energy`** matlab function. They also have to apply the concept of a euclidean distance calculation to implement a decision boundary as evidenced by their writing of the **`nearest_neighbor`** matlab function. This outcome is assessed by the numerical performance of the matlab functions on real data.

The SO (b) refers to an ability to design and conduct experiments, as well as to analyze and interpret data. In the Saint Louis University Electrical Engineering program assessment process, this outcome is split into two outcomes (b.1) and (b.2) with the latter defined as an ability to analyze and interpret data. Only (b.2) is assessed for this laboratory project. A team demonstrates this ability by developing a classifier that provides good classifier performance in relation to the comparison classifier. If the confusion matrices are comparable, then the team demonstrated a reasonable ability to analyze and interpret data.

The SO (d) refers to an ability to function on multidisciplinary teams. Although the student teams are not multidisciplinary in an engineering discipline sense, they do involve bringing together students with different skills to complete this project and also involves integrating individual student work into a single solution. First, this project requires analyzing data, writing matlab code, and writing a report each of which are sometimes best suited to an individual student. Second, many teams break down the classifier into pieces, essentially parts of the decision tree, that the team assigns to individual students. At some point, the team comes together and has to write the **`classify_vowel`** matlab function which requires they integrate the individual work each team member completed. The skill of *design integration* is fundamental to being a productive member of a multidisciplinary team. The assessment of this outcome is partly evidenced by a team survey and partly evidenced by instructor observation. First, the instructor observes the team's work in a laboratory setting, often listening in on conversations to see how a team is functioning and taking notes during class. Seconds, teams are required to individually fill out a survey to describe the work of their team, how it was broken down, who had what responsibility, and how decisions were made. All of this is meant to assess how well the team functioned as an engineering team.

The SO (e) refers to an ability to identify, formulate, and solve engineering problems. Designing the classifier tree is an engineering problem to be solved with the form of the solution up to the individual teams. Designing the classifier tree involves identifying smaller problems to be worked, formulating a solution, and completing the solution. For example, one team might recognize that it is possible to separate out speaker #2 from the others which would then require a subtree to be developed for the vowel data for speaker #2. This activity involves problem identification, formulating a solution and carrying out that solution. The assessment of this outcome results from a report requirement that the team explain the specific process that the group went through to build the classifier tree. The description of this process should include statements that indicate the team observed something in the data that provided the rationale for the broad structure of the classifier tree.

The SO (h) refers to the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context. In addition, this project requires each team to perform a market analysis of their classifier assuming that the performance of a full speech recognition system in their *native language* is the same as the performance of their long vowel sound classifier. The team must choose a reasonable price for their speech recognition system, consistent with current market, and then determine how many people may be able to purchase their product for the price world-wide. Also to be considered is the performance of their long vowel sound classifier for individual team members. If the classifier did not perform as well for one particular team member who speaks Spanish, for example, then their estimates should take that into account. For example, suppose this group consisted of one international student from spain whose native language is Spanish and two students from the US whose native language is English and suppose that the performance of their system worked extremely well for the English speaking student but not as well for the student whose native language is Spanish. The market analysis would need to consider: (1) the number of people world wide who speak Spanish and English, (2) the number of those people who potentially have the economic means to purchase their system, and (3) the number of people who might consider buying their system given the overall system performance. Finding appropriate data for this market analysis is left to the group to find. A bibliography must be provided to document their acquired information.

The assessment of this outcome is rather subjective and is based upon several factors including (1) the degree to which their established price matches currently available systems, (2) the degree to which the population they cite seems reasonable, (3) whether the group considered economic means, and (4) whether the group factored in the performance of their classifier. The Dragon [5] speech recognition software costs around $150 for a computer running Windows 10 and is a reasonable comparison. The population estimates as well as the number of people with need varies widely across the group depending upon their perspectives. For example, there are speech recognition system built into smart phones so that application can be used as a basis for determining population need by finding the numbers of people who have a smart phone. The details of a team's market analysis are less important than the degree to which they form a set of estimates and have a reasonable approach for backing up those estimates. The actual sources teams cite vary widely and almost all are typically found online.

The SO (k) refers to the ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. The modern engineering tools they use for this laboratory project is the matlab software development environment. Teams are required to write 3 different matlab functions as described previously. The assessment of this outcome is based upon a rubric that includes three primary components: (1) the degree to which their functions are well documented, (2) the degree to which their functions are logically correct and computationally efficient, and (3) the degree to which their functions yield correct results.

Conclusions

This paper describes a laboratory project that requires student teams to build a classifier to classify the 5 long vowel sounds a, e, i, o, and u. The project goals are for the student to (1) understand the importance of the Fourier Spectrum for developing useful signal analysis algorithms and systems, (2) develop a speaker-independent vowel classification system to distinguish the 5 long vowel sounds for a team of students, (3) use a metric space and support vectors to perform classification, (4) statistically evaluate a speaker-independent vowel classification system, and (5) appreciate the relationship between commercial viability and system performance.

The free-form nature of the data provides an excellent setting for students to demonstrate, in part, an ability to analyze and interpret data, formulate and solve engineering problems, and demonstrate an understanding of the link between system performance and the marketability of their system. All of these and others directly relate to a number of ABET Student Outcomes.

REFERENCES

[1] Criteria for Accrediting Engineering Programs, 2016-2017, https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2016-2017, Website URL [Accessed: 18-Jan-2019].

[2] Encyclopaedia Britannica, https://www.britannica.com/science/ear/The-physiology-of-hearing, Website URL [Accessed: 20-Jan-2019].

[3] L. R. Rabiner and R. W. Schafer, *Theory and Applications of Digital Speech Processing*. Pearson, 1st ed., 2011.

[4] The Mathworks, https://www.mathworks.com, Website URL [Accessed: 20-Jan-2019].

[5] Nuance, https://www.nuance.com/dragon.html, Website URL [Accessed: 21-Jan-2019].