Classification of Peterson & Barney's vowels using Weka

Aldebaro Klautau
UC San Diego - EBU - I
La Jolla, CA 92093-0407
Email: a.klautau@ieee.org
February 29, 2002

1. Introduction

Gordon Peterson and Harold Barney describe a detailed investigation of sustained American English vowels in [1]. The article presents acoustic measurements of fundamental frequency (F0) and first three formant frequencies (F1-F3). The authors also conducted experiments where listeners were asked to identify words.

Raymond Watrous [2] re-organized the database collected in [1] and made it temporarily available from University of Pennsylvania. In 1994, Murray Spiegel posted a message on *comp.speech* news group[1] indicating the database had been made available again, but from Bellcore. Later, Tony Robinson made it permanently available from Cambridge University[2]. Now the database, formatted for Weka [3], is also available from [4]. It is called the *pbvowel* database because Robison has made available another database that is usually identified as *vowel* in [5].

The *vowel* database consists of eleven sustained vowels of British English collected by David Deterding and represented by log-area ratio (LAR) parameters. When using the standard partitions into test and train sets[3], the error rates for the *vowel* database are usually higher (around twice) than for *pbvowel*. Besides being based on different English accents, the *vowel* and *pbvowel* databases were not obtained through the same experimental procedures. Also, formants can be seen as a more efficient representation of vowels than LAR parameters. LAR is a well-known, but outdated parameterization of speech [6]. In speech coding the LARs were substituted by the line spectral frequencies [7] and in speech recognition, parameters obtained through cepstrum analysis are more popular [8].

The motivations for writing this report about *pbvowel* were:

- it has been used by several researchers (e.g. [9], [10], [11]) but their results can not be easily compared due to the lack of a standard experimental procedure;
- the open-source Weka machine learning package [3] provides implementations of several classical pattern recognition techniques. The command lines for Weka are provided here, so the reported results can be easily reproduced;
- the conventional nomenclature for formants (F1, F2 and F3) and fundamental frequency (F0) is confusing. Some publications mistakenly mention that *pbvowel* contains four formants (e.g. [11], [2]). It seems important to present their definitions and emphasize the distinction between F0 and formants;
- [1] completes 50 years in 2002 and deserves a celebration!

---

[1] This group was later split in *comp.speech.users* and *comp.speech.research*.
[2] The file is called PetersonBarney.tar.Z and is available at *ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/data/*.
[3] The error rate when using *n*-fold cross-validation is usually considerably smaller.

This report is organized as follows. Section 2 presents a brief description of the concepts of formants and fundamental frequency, the numerical attributes of the *pbvowel* database. Section 3 reviews the work reported in [1]. A partition of the database is presented in section 4. The results of vowel classification using Weka are reported in section 5. Section 6 shows some plots obtained considering only the first two formants. The final considerations are in section 7.

2. Speech formants and fundamental frequency

In the Fifties, Gunnar Fant made significant contributions to the development of an acoustic theory of speech where the speech wave is seen as the response of the vocal tract filter systems to one or more sound sources. This is the basic principle of the so-called *source-filter* model of the speech production process. A detailed description of this model can be found in [12]. Given the scope of *pbvowel*, it suffices to consider here only the production of vowels.

Fundamental frequency (*F0*)*:*

When a vowel is produced, the vocal cords vibrate on a rate called fundamental frequency or F0. In practice, F0 is not exactly the same over time but varies intentionally or unintentionally (machines can produce a monotonic F0 though). The average F0 values of children are higher than of adults, and women have higher F0 than men, as can be inferred from *pbvowel*. F0 is a parameter related to the source (as discussed later, the formants are related to the filter).

The *pitch* frequency is closely related to F0 and in many cases these terms are used interchangeably. In more strict terminology, pitch is a tonal *sensation* and *frequency* a property of the sound *stimulus* [12]. It is not an easy task to estimate F0, but it is harder to quantify the subjective sensation of pitch. The *mel* scale [13], which is popular in speech recognition, is an attempt to relate frequency and pitch.

The method used for estimating F0 in [1] is not discussed by the authors (neither in the companion paper [14]).

Formants:

The main articulators involved in vowel production are tongue and lips. Depending on their configuration the vocal tract imposes different shapes to the resultant speech spectrum. The source-filter model assumes the source spectrum $S(f)$ is modified by the filter (vocal tract) function $T(f)$, leading to a speech spectrum $P(f) = S(f)\,T(f)$. For vowels, $S(f)$ is basically composed by *harmonics* of F0. The speech formants can be defined either as the peaks of $|S(f)|$ or $|T(f)|$, which generates some confusion. From [12] (page 20):

"The spectral peaks of the sound spectrum |P(f)| are called formants. (…) *it may be seen that one such resonance has its counterpart in a frequency region of relatively effective transmission through the vocal tract. This selective property of |T(f)| is independent of the source. The frequency location of a maximum in |T(f)|, i.e., the resonance frequency, is very close to the corresponding maximum in spectrum P(f) of the complete sound. Conceptually these should be held apart but in most instances resonance frequency and formant frequency may be used synonymously. Thus, for technical applications dealing with voiced sounds it is profitable to define formant frequency as a property of T(f)."*

Modern textbooks[4] define formants as the resonance frequencies associated to $T(f)$ [16] (page 18), [8] (page 27). As pointed out by Fant, this definition is sensible because eliminates the influence of the source characteristics, which are speaker-dependent (e.g. different people say the same vowel with potentially different values of F0). On the other hand, it raises the problem that resonances of $T(f)$ are sometimes undetermined given that usually only a measure of $P(f)$ is available. For example, a person (e.g. child) with high F0 would produce $S(f)$ with harmonics of F0 highly separated in frequency, so the peaks of $T(f)$ would be hardly visualized if they were far from any harmonic.

There are many techniques for estimating formants. A survey of formant estimation methods used in the past is given in [17] (page 165). A popular modern approach is to obtain formants from the roots of a filter calculated through linear prediction techniques [6]. Fig. 1 and Fig. 2 show spectrograms [6] with formant tracks (F1-F3) superimposed. The first sentence is composed basically by vowels and the method gives fairly good results. The sentence correspondent to Figure 2 has more phonetic variation (fricatives, nasals, etc.) and the results are not so good as in Figure 1. In fact, under realistic conditions (noise, etc.), formants estimation is a difficult task.

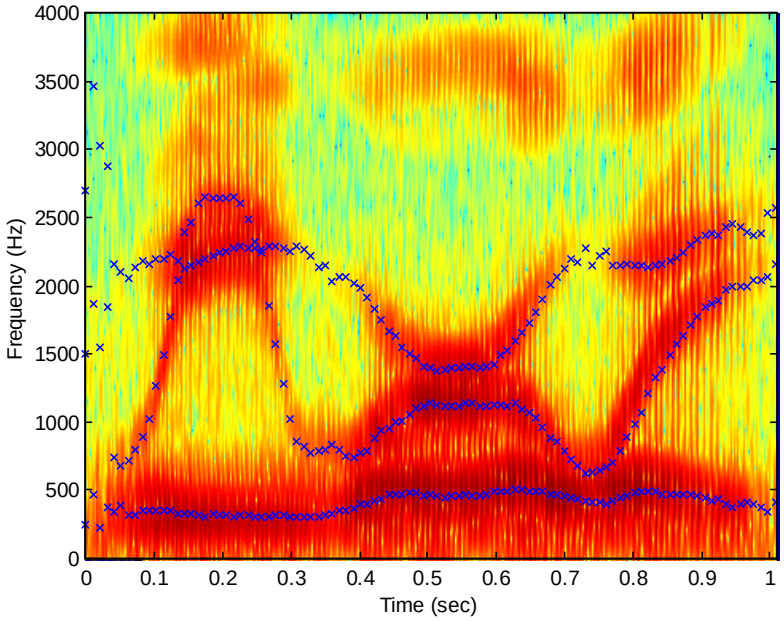

Fig. 1. First three formants F1- F3 estimated from the roots of linear prediction filters of order 8. The sentence is "We were away" spoken by a male speaker with low F0.

[4] Kenneth Stevens observes that the concept of a formant should be restricted to natural frequencies of the vocal tract when there is no coupling to the nasal cavities [15] (page 131).

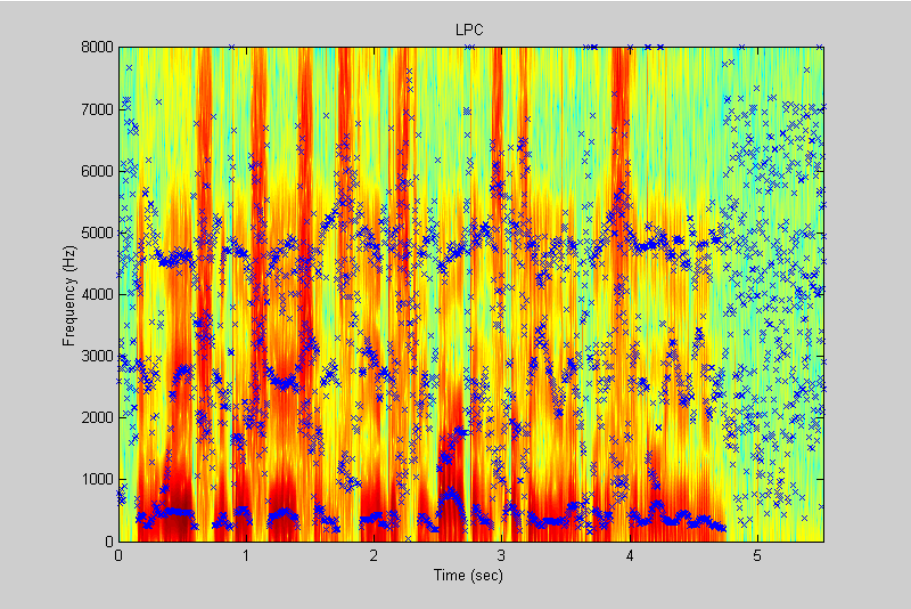

Fig. 2. First three formants F1-F3 estimated from the roots of linear prediction filters of order 8. The sentence is "In wage negotiations, the industry bargains as a unit with a single union".

The method for estimating formants used in [1] is described, though not thoroughly, in [14]. It consists of calculating a weighted average of the spectrum components. This approach is known as "peak-picking" and, as other formant estimation methods, is error prone and usually requires the supervision of an expert for eventual corrections.

3. Peterson & Barney's paper

Data collection

A list of ten words was presented to 76 speakers, each word beginning with [h] and ending with [d], and differing only in the vowel. The words are listed in Table I. Each speaker was asked to pronounce two different lists, each list corresponding to a random permutation of the 10 words. Therefore, the total number of recorded words was 1520.

The first formant F1 can be related to how far the tongue is raised and F2 to which part of the tongue is raised. Therefore, vowels can be organized according to the tongue's position in plots as Fig. 3. Nowadays phoneticians point out that pictorial representations as Fig. 3 should be seen as first-order approximations, given that it is actually possible to produce a vowel with a configuration radically different from the one suggested by Fig. 3 [18].

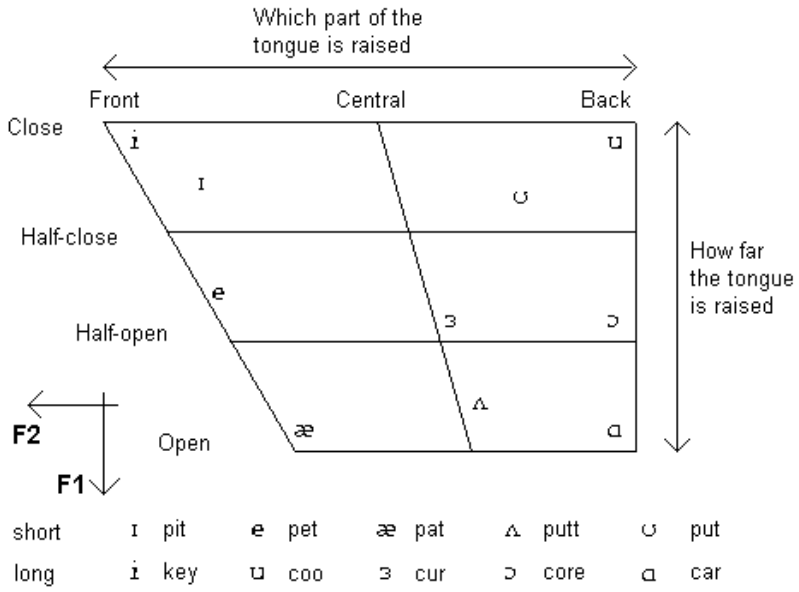

Fig. 3. Relations between formants and tongue's positions. Note that the F1 and F2 axes should have the directions indicated by Figure 2 in order to provide the picture popular among phoneticians.

The vowels are identified in Fig. 3 by symbols defined in the International Phonetic Alphabet (IPA). The IPA makes extensive use of letters not available on computers. The ARPABET is one of the proposed mappings from IPA to ASCII symbols. The vowels in *pbvowel* were labeled according to the two-characters representation of the ARPABET[5]. Table I shows the words and correspondent vowels. More details about these phonetic symbols can be found in [18] (page 29).

Table I - Words used in [1] and the correspondent vowels. The actual sounds can be heard at [20]. The IPA diacritic ː indicates the vowel has longer duration and the "hook" diacritic ˞ indicates the vowel ɜ is influenced ("colored") by [r].

Word used in [1]	IPA symbol for the vowel	More detailed IPA transcription	ARPABET symbol	Example in context /h_/	Example in context /b_d/	Example in context /h_t/	Example in context /k_d/
heed	i	iː	IY	he	bead	heat	keyed
hid	ɪ	ɪ	IH	–	bid	hit	kid
head	ɛ	ɛ	EH	–	bed	–	–
had	æ	æ	AE	–	bad	hat	cad
hod	ɑ	ɑː	AA	–	bod	hot	cod
hawed	ɔ	ɔː	AO	haw	bawd	–	cawed
hood	ʊ	ʊ	UH	–	–	–	could
who'd	u	uː	UW	who	booed	hoot	cooed
hud	ʌ	ʌ	AH	–	bud	hut	cud
heard	ɜ	ɜ˞ː	ER	her	bird	hurt	curd

[5] An on-line version of the ARPABET can be found at [19].

Listening tests

The 1,520 recorded words were presented to a group of 70 adult observers. Thirty-two of the 76 speakers were also among the observers. The experiment was conducted in seven sessions. The general purpose of these tests was to obtain an aural classification of each vowel. Each observer would mark the word he heard. For each vowel, all correspondent 152 words were presented to the observers. The ease with which the observers classified the various vowels varied significantly. Of all IY sounds, for instance, 143 were unanimously classified by all observers as IY. On the other hand, only 9 when the intended vowel was AA. This result is summarized in Fig. 4. From [1]: "*The very low scores of AA and AO result primarily from the fact that some members of the speaking group and many members of the listening group speak one of the forms of American dialects in which AA and AO are not differentiated*."

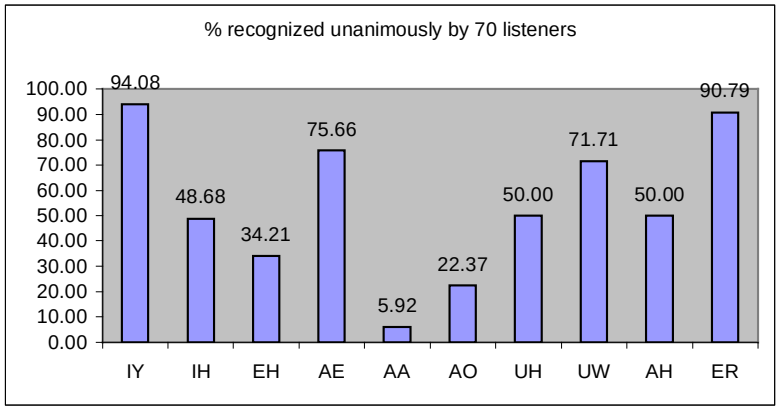

Fig. 4. The percentage unanimously identified by all 70 listeners of 152 repetitions for each vowel.

The complete confusion matrix is shown in Table II. A graphical representation of the confusion matrix is shown in Fig. 5. The total average "error" of the listening test was 5.57 %.

Table II - Confusion matrix for listening experiment in [1]. Lines indicate the intended, and columns the vowel understood by listeners. The last two columns show the total[6] and error per line.

	IY	**IH**	**EH**	**AE**	**AA**	**AO**	**UH**	**UW**	**AH**	**ER**	**Total**	**% error**
IY	10267	4	6	–	–	3	–	–	–	–	10280	0.13
IH	6	9549	694	2	1	1	–	–	–	26	10279	7.10
EH	–	257	9014	949	1	3	–	–	2	51	10277	12.29
AE	–	1	300	9919	2	2	–	–	15	39	10278	3.49
AA	–	1	–	19	8936	1013	69	–	228	7	10273	13.01
AO	–	–	1	2	590	9534	71	5	62	14	10279	7.25
UH	–	–	1	1	16	51	9924	96	171	19	10279	3.45
UW	–	–	1	–	2	–	78	10196	–	2	10279	0.81
AH	–	1	1	8	540	127	103	–	9476	21	10277	7.79
ER	–	–	23	6	2	3	–	–	2	10243	10279	0.35

[6] Each vowel should be voted 10640 (152 × 70) times, but the "Total" column shows the actual average number is around 10279. The total number of votes was 102780, indicating that 3620 from the expected total of 106400 votes were not considered when organizing the table. The reasons for that are not mentioned in [1].

Some instances in *pbvowel* are labeled as being unanimously classified by all observers. It should be noticed that the documentation available with the *pbvowel* version distributed by Cambridge University mentions only 26 observers, while 70 were reported in [1]. In fact, the statistics of the vowels in *pbvowel* that were labeled as unanimously classified does not match the results in [1]. In *pbvowel*, only 321 vowels were labeled as not unanimously identified by observers. The reason for this discrepancy on the number of observers is unknown.

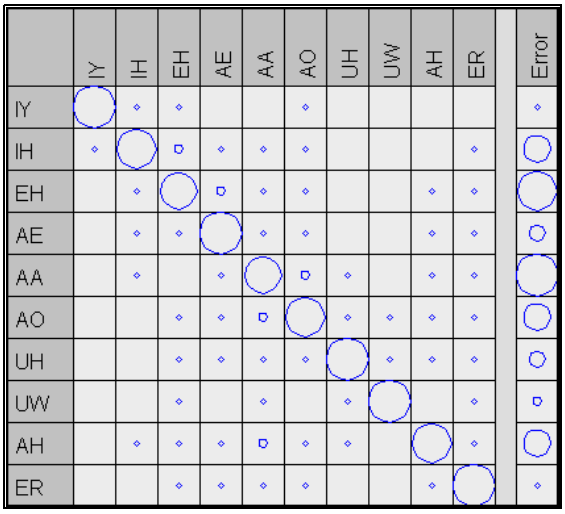

Fig. 5. Graphical representation of the confusion matrix for the listening test in [1]. The columns indicate the recognized vowel. The *radii* are proportional to the entries in Table II. The last column illustrates the difficulty on recognizing each vowel.

Acoustic measurements

Besides F0 and three formants, the formant amplitudes in dB were also reported in [1], but are missing in *pbvowel*. The statistics for F0 and formants F1-F3 are listed in Table III.

Table III- Statistics of the numerical attributes in *pbvowel*: minimum, maximum, average and standard deviation for all 1,520 instances.

	Min	***Max***	***Average***	***Std***
F0	91	350	191.29	60.36
F1	190	1300	563.30	201.25
F2	560	3610	1624.38	637.01
F3	1400	4380	2707.81	519.45

4. Partitioning the database

The attributes for the version of *pbvowel* distributed by [5] are listed in Table IV. In classification experiments, the attributes *speaker_number* and *confidence* should not be used.

A partition of *pbvowel* into four subsets based on the speaker identity is presented in this section. All subsets have 19 speakers, corresponding approximately to the same number of males, females and children. An open test set framework is adopted, i.e., speakers in the train do not belong to the test set. The standard partition is a train set corresponding to the union of *A* and *B*

(with *C* and *D* corresponding to the test set). Eventually, if a given algorithm demands more training data, set *C* can be used for training and the situation properly reported. Alternatively, set *C* can be used as a validation set.

Table IV- Attributes of *pbvowel*.

Attribute name	***Type***
gender_age	nominal: {male, female, child}
speaker_number	numerical, integer ∈ [1, 76]
confidence	nominal: {low, high}
F0	numerical, integer
F1	numerical, integer
F2	numerical, integer
F3	numerical, integer
vowel	nominal: {IY,IH,EH,AE,AA,AO,UH,UW,AH,ER}

Table V- Partition of *pbvowel* into four disjoint subsets according to speaker identity. All sets have 380 vowels, corresponding to 19 speakers.

Set	***Males***	***Females***	***Children***
A	1-8	34-40	62-65
B	9-16	41-47	66-69
C	17-24	48-54	70-73
D	25-33	55-61	74-76

It may be interesting to consider only the instances that were labeled as unanimously identified by the listeners, i.e., those with attribute *confidence* equal to *high*. A suffix "*u*", as shown in Table VI, identifies the corresponding subsets. The training set composed by *A_u* and *B_u* contains 599 instances, while the test set (*C_u* and *D_u*) contains 600. The distribution of vowels is not uniform when considering only the ones unanimously identified.

Table VI- Number of vowels unanimously identified by listeners in each set in Table V.

Set	***A_u***	***B_u***	***C_u***	***D_u***
# of vowels	289	310	295	305

Table VII- Identifiers for simulations using different combinations of the attributes in *pbvowel*.

Identifier	***Used attributes (besides the vowel)***
gF0-3	*gender_age, F0, F1, F2, F3*
F0-3	*F0, F1, F2, F3*
F1-3	*F1, F2, F3*
F1-2	*F1, F2*
u (as prefix, e.g., *uF0-F3*)	only instances with *confidence* equal to *high*

This report presents simulations with different combinations of the attributes in *pbvowel*. Table VII summarizes these combinations. For example, a simulation identified by *uF1-3* uses a train set composed by instances for which *confidence* is *high*, and discarding attributes *gender_age*, *speaker_number*, *confidence* and *F0*.

5. Classification using Weka

This section presents results obtained with Weka, version 3.2. Weka can be obtained at [3] and the source code in Java is also available. Table VIII shows results obtained with 20 different classifiers, for 5 different combinations of attributes as described in Table VII. The command lines for reproducing these results are given in the Appendix. The best results in Table VIII are also the best in the literature that the author is aware of. However, some classifiers were not tuned, as Table VIII was designed to serve simply as basis for comparisons. For example, the default in Weka's multilayer perceptron is a number of units in the hidden layer given by the average between the number of attributes and classes (e.g., (4+10)/2 = 7 units for the first column *F0-3*). Tuning the network topology leads to improvements. However, it is questionable to use the test set to validate parameters, i.e., choosing the parameters that lead to the best results for the test set, implies the performance on this set does not necessarily indicate the generalization capability of the classifier. A better approach in this case would be to use set *C* as a validation set and report results on set *D*. Here, the default values were used for most classifiers and a validation set was not adopted.

It can be seen from Table VIII that the extra attribute *gender_age* in *ugF0-3* does not bring better results when compared to *uF0-3*. In fact, as discussed in section 2, F0 alone does not bring information about vowel identity. However, F0 is related to attribute *gender_age*. Given that vowels (and consequently F1-F3) vary depending if the speaker is a man, woman or child, *F0* or *gender_age* can improve classification accuracy if the classifier effectively uses this information. For classifiers as Naïve Bayes, which assumes independence among attributes given the class, *F0* and *gender_age* do not help.

For an easier visualization, the results of Table VIII are also shown in Fig. 6. The reader is refereed to the Weka documentation for an explanation of acronyms as ECOC (error-correcting output code), etc.

Table VIII - Misclassification rate for 20 classifiers designed with Weka. The columns are labeled according to Table VII. The best result for each data set is indicated in bold.

#	*Algorithm and configuration*	*F0-3*	*ugF0-3*	*uF0-3*	*uF1-3*	*uF1-2*
1	Naïve Bayes - simpler implementation	24.74	21.33	21.00	21.33	26.83
2	Naïve Bayes	24.60	21.17	21.50	22.00	27.17
3	Naïve Bayes with kernel estimation	22.10	19.00	17.83	19.50	25.00
4	Kernel density estimator	17.37	13.50	13.17	13.33	18.67
5	K-nearest neighbor with K = 5	16.58	13.50	13.17	14.33	21.50
6	K-nearest neighbor with entropic distance (KStar)	16.84	11.83	12.33	12.67	**18.33**
7	Multilayer perceptron	13.42	12.83	**9.83**	15.50	19.67
8	C4.5 decision tree with reduced error pruning	23.29	20.5	21.5	19.83	27.83
9	Bagging 10 iterations using C4.5 with reduced error pruning	19.08	14.67	14.33	16.00	20.33
10	Bagging 50 iterations using C4.5 with reduced error pruning	17.89	14.33	14.67	17.83	22.00
11	AdaBoost M1 using C4.5 without reduced error pruning	16.05	14.33	14.00	16.17	29.50
12	AdaBoost M1 using C4.5 with reduced error pruning	18.29	17.50	16.83	19.17	26.83
13	Boosted stumps with LogitBoost	20.00	18.33	18.50	19.50	24.17
14	10 binary classifiers (one-against-all): SVM with polynomial kernel of order 5	15.92	11.17	11.50	17.17	28.83
15	10 binary classifiers (one-against-all): 50 iterations of AdaBoost using stumps	19.34	16.50	17.83	19.50	27.83
16	10 binary classifiers (one-against-all): 50 iterations of LogitBoost using stumps	19.60	15.83	15.67	18.67	23.50
17	10 binary classifiers (one-against-all): multilayer perceptron	12.89	10.50	10.33	**11.00**	21.17
18	20 binary classifiers (random code): multilayer perceptron	14.60	11.33	**9.83**	22.83	30.83
19	20 binary classifiers (random code): 100 iterations of AdaBoost using stumps	31.58	22.50	22.33	27.83	30.00
20	20 binary classifiers (random code): 10 iterations of AdaBoost using multilayer perceptrons	**12.63**	**10.00**	10.00	11.67	22.33

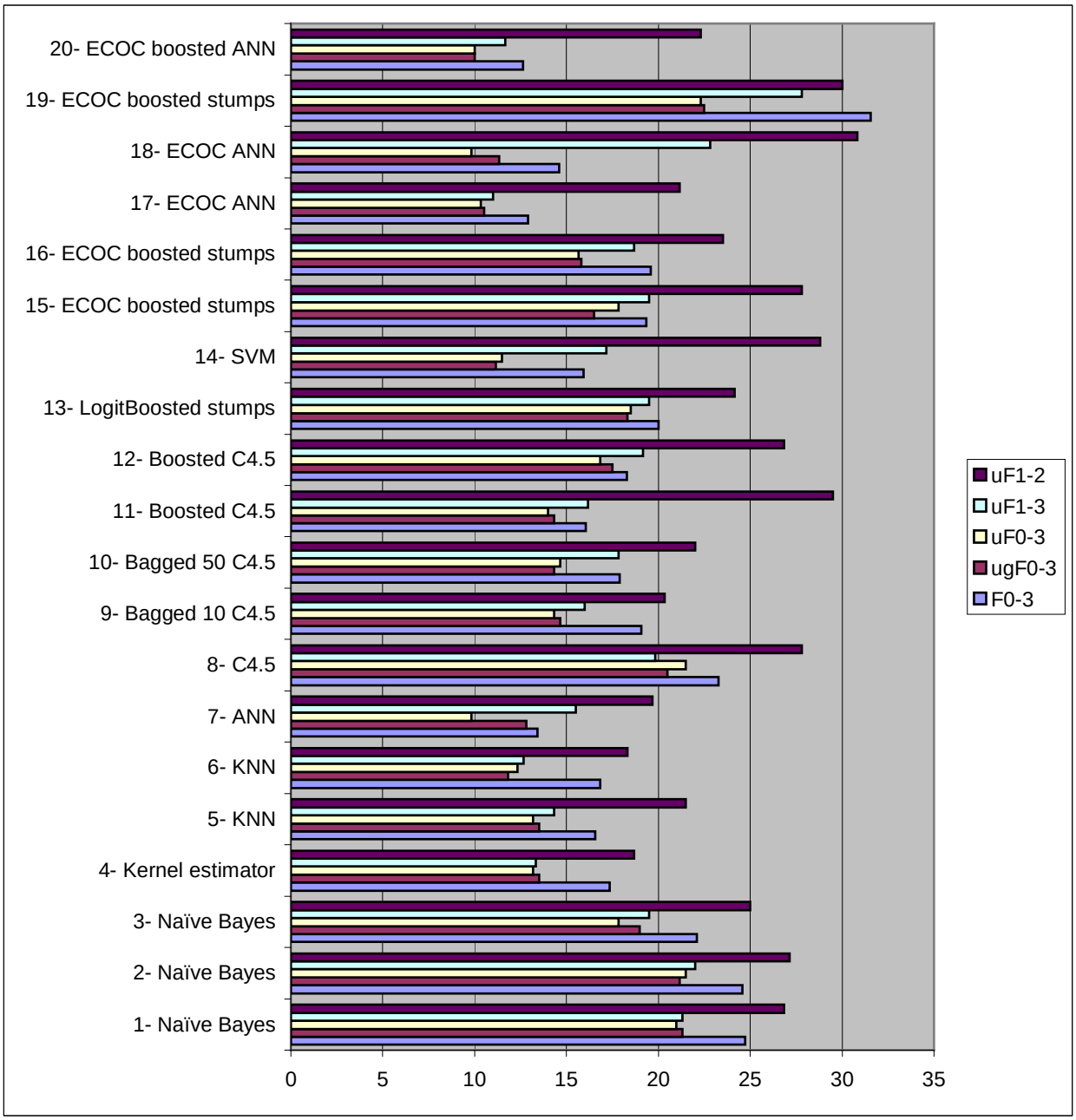

Fig. 6. Illustration of misclassification rates shown in Table VIII.

Fig. 7 compares the results obtained with the neural network classifier correspondent to number 17 in Table VIII with the listening experiments in [1]. The pattern of errors is not exactly the same, but there are coincidences as, for example, vowels IY and ER are recognized with high accuracy in both cases. In relative terms, the machine has more troubles with UH, while listeners have with AA.

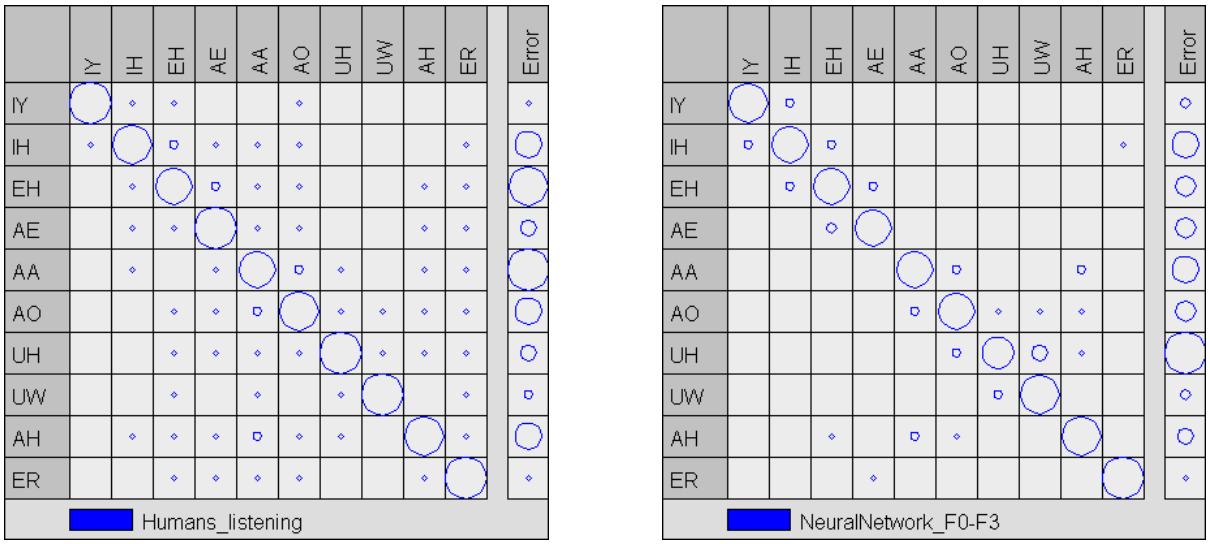

Fig. 7- Confusion matrices for listening test (Fig. 5) at the left and results with neural network (number 17 in Table VIII) at the right. The columns indicate the recognized vowel. The *radii* are proportional to the entries.

6. Visualization of results on F1 x F2 plane

As shown in this section, illustrative plots can be obtained when using only the first two formants. The figures in this section use color for an easier visualization.

Fig. 8 shows the instances used for training and testing the classifiers discussed in this section (see the appendix for other plots) Note that the regions overlap significantly in the F1 x F2 plane, especially for vowel ER. In spite of being perceptually less important than F1 and F2, the third formant F3 is useful for distinguishing ER from the others. In fact, according to Table II, ER can be easily recognized by listeners.

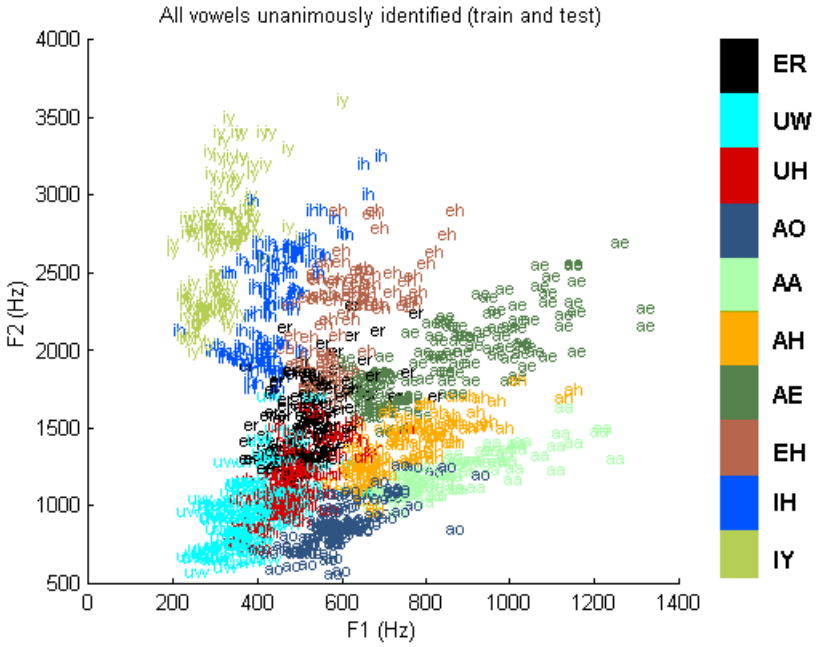

Fig. 8. Plot of all instances for which *confidence* is *high* (identified unanimously).

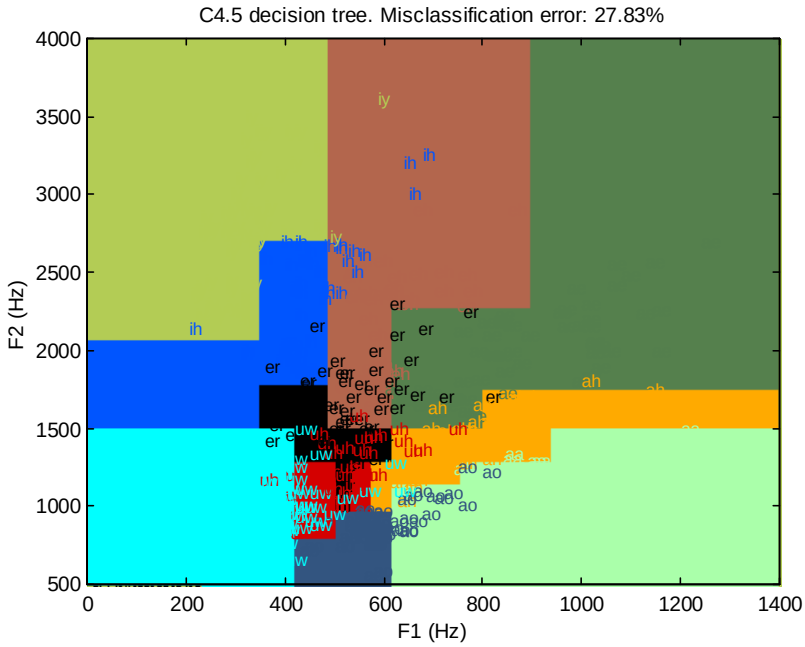

Fig. 9. Decision boundaries obtained with a C4.5 decision tree (number 8 in Table VIII). The colors are the same as in Fig. 8. The test instances are superimposed with their corresponding colors, such that errors can be identified.

A decision tree as C4.5 leads to boundaries that are parallel to the axes as shown in Fig. 9. This limitation is clearly problematic in this case. Some classifiers that are capable of arbitrary decision boundaries as *K*-nearest neighbor and neural networks led to better results, as shown in Fig. 10 and Fig. 11, respectively. Note that the *K*-nearest neighbor established a non-contiguous region for ER.

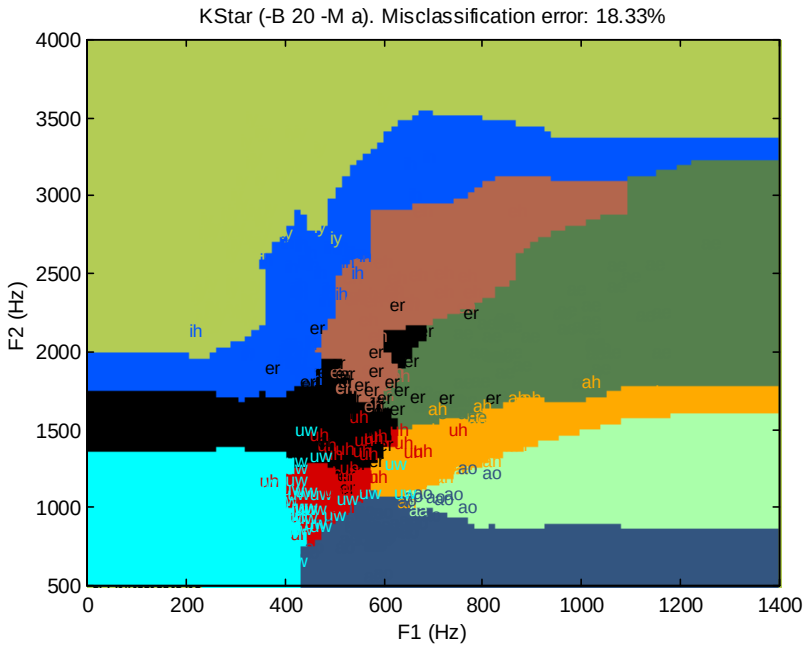

Fig. 10. Decision boundaries and errors obtained with a K-nearest neighbor classifier (number 6 in Table VIII).

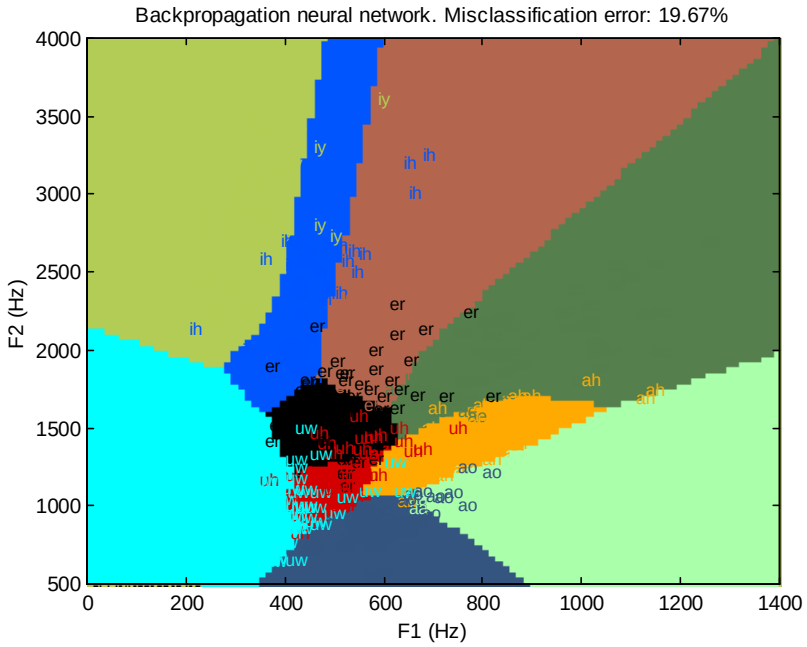

Fig. 11. Decision boundaries and errors obtained with a multilayer perceptron (number 7 in Table VIII).

Some classifiers, as the implementation of support vector machines (SVM) in Weka, can not deal with more than two classes. One alternative is to break down the multiclass problem into several binary problems. Fig. 12 shows the results obtained by training 10 binary classifiers using a *one-versus-all* methodology [11].

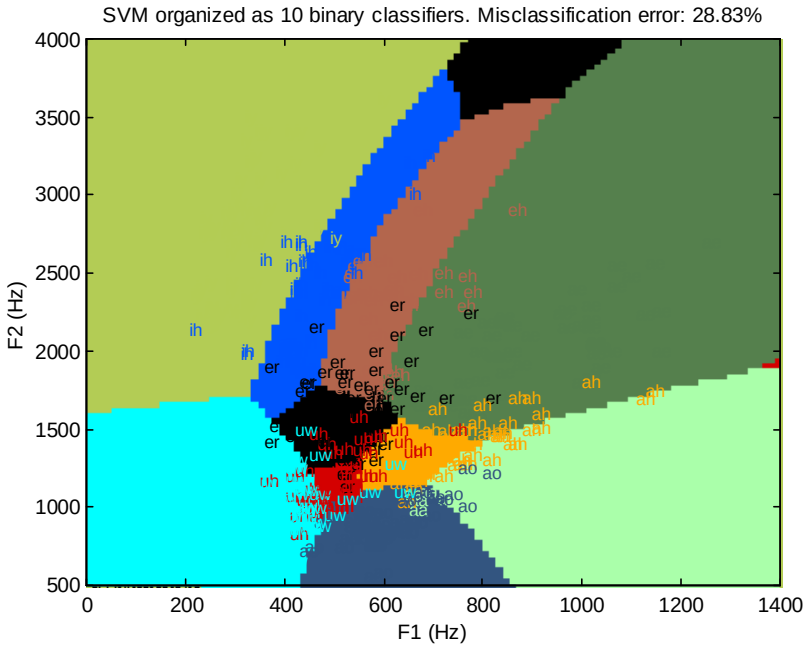

Fig. 12. Decision boundaries and errors obtained with support vector machines (number 14 in Table VIII).

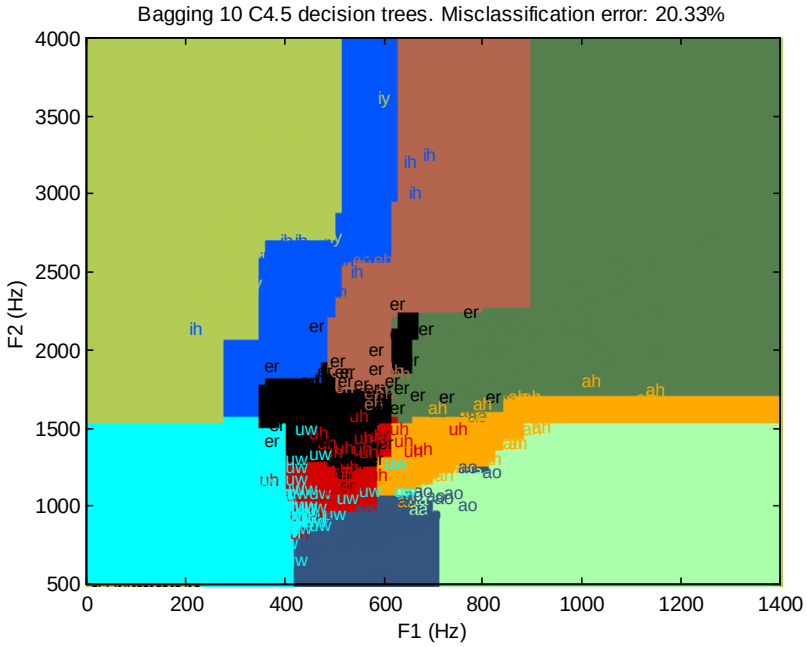

Fig. 13. Decision boundaries and errors obtained by bagging 10 decision trees (number 9 in Table VIII).

Another approach for circumventing limitations of classifiers is to build ensembles with methods as boosting or bagging. Fig. 13 illustrates how bagging can improve results, obtaining decision boundaries not as limited as the ones of the basic decision tree in Fig. 9. There is even a

non-contiguous region for ER in Fig. 13, similar to the one suggested by the *K*-nearest neighbor of Fig. 10, which obtained the best result in terms of classification based only on F1 and F2.

7. Conclusions

This report reviewed the work described in [1]. The concepts of formants and fundamental frequency were discussed in order to avoid confusions with the nomenclature F0 and F1-F3. A flexible partition of the database was proposed, such that results obtained by different researchers can be easily compared. Results of vowel classification experiments conducted with Weka were presented, with some plots on the F1 x F2 plane to help interpreting results.

References

[1] G. Peterson and H. Barney, "Control methods used in a study of vowels," *Journal of the Acoustical Society of America*, vol. 24, pp. 175-184, 1952.

[2] R. L. Watrous, "Current status of Peterson-Barney vowel formant data," *Journal of the Acoustical Society of America*, vol. 89, pp. 2459-60, 1991.

[3] E. Frank and e. al, "Weka [http://www.cs.waikato.ac.nz/ml/weka/]," The University of Waikato, 2002.

[4] A. Klautau, "http://speech.ucsd.edu/aldebaro/repository," 2002.

[5] C. L. Blake and C. J. Merz, "UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]." Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[6] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*: Prentice-Hall, 1978.

[7] A. M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communication Systems*: John Wiley & Sons, 1994.

[8] X. Huang, A. Acero, and H.-W. Hon, *Spoken language processing*, 1 ed: Prentice-Hall, 2001.

[9] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive Mixture of Local Experts," *Neural Computation*, vol. 3, pp. 79-87, 1991.

[10] R. S. Shadafan and M. Niranjan, "A dynamic neural network architecture by sequential partitioning of the input space," Cambridge University, Cambridge May 13 1993.

[11] P. Clarkson and P. J. Moreno, "On the use of support vector machines for phonetic classification," presented at IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999.

[12] G. Fant, *Acoustic theory of speech production*, 1 ed: Mounton & CO, 1960.

[13] S. Stevens and J. Volkman, "The relation of pitch to frequency," *Journal of Psychology*, vol. 53, pp. 329, 1940.

[14] R. K. Potter and J. C. Steinberg, "Toward the specification of speech," *The Journal of the Acoustical Society of America*, vol. 22, pp. 807-820, 1950.

[15] K. N. Stevens, *Acoustic Phonetics*. Cambridge, Mass.: The MIT Press, 1999.

[16] R. Kent and C. Read, *The Acoustic Analysis of Speech*: Singular, 1992.

[17] J. L. Flanagan, *Speech Analysis, Synthesis and Perception*. New York: Springer, 1972.

[18] P. Ladefoged, *A Course in Phonetics*, 4 ed: Harcourt Brace, 2001.

[19] A. Klautau, "ARPABET and the TIMIT alphabet [http://speech.ucsd.edu/aldebaro/papers]," 2001.

[20] P. Ladefoged, "http://hctv.humnet.ucla.edu/departments/linguistics/VowelsandConsonants/course/chapter2/amerenglishvowels.html," 2002.

Appendix - Command lines for Weka

#	*Algorithm*	*Weka's command line*
1	Naïve Bayes - simpler implementation	weka.classifiers.NaiveBayesSimple
2	Naïve Bayes	weka.classifiers.NaiveBayes
3	Naïve Bayes with kernel estimation	weka.classifiers.NaiveBayes -K
4	Kernel density estimator	weka.classifiers.KernelDensity
5	K-nearest neighbor with K = 5	weka.classifiers.IBk -K 5 -W 0
6	KStar	weka.classifiers.kstar.KStar -B 20 -M a
7	Multilayer perceptron	weka.classifiers.neural.NeuralNetwork -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
8	C4.5 decision tree with reduced error pruning	weka.classifiers.j48.J48 -R -N 3 -M 2
9	Bagging 10 iterations using C4.5 with reduced error pruning	weka.classifiers.Bagging -S 1 -I 10 -P 100 -W weka.classifiers.j48.J48 -- -R -N 3 -M 2
10	Bagging 50 iterations using C4.5 with reduced error pruning	weka.classifiers.Bagging -S 1 -I 50 -P 100 -W weka.classifiers.j48.J48 -- -R -N 3 -M 2
11	AdaBoost M1 using C4.5 without reduced error pruning	weka.classifiers.AdaBoostM1 -P 100 -I 10 -S 1 -W weka.classifiers.j48.J48 -- -C 0.25 -M 2
12	AdaBoost M1 using C4.5 with reduced error pruning (only approximately 4 iterations were completed)	weka.classifiers.AdaBoostM1 -P 100 -I 10 -S 1 -W weka.classifiers.j48.J48 -- -R -N 3 -M 0
13	Boosted stumps with LogitBoost	weka.classifiers.LogitBoost -P 100 -I 10 -W weka.classifiers.DecisionStump --
14	10 binary classifiers (one-against-all): SVM with polynomial kernel of order 5	weka.classifiers.MultiClassClassifier -E 0 -R 2.0 -W weka.classifiers.SMO -- -C 1.0 -E 5.0 -A 1000003 -T 0.0010 -P 1.0E-12 -O
15	10 binary classifiers (one-against-all): 50 iterations of AdaBoost using stumps	weka.classifiers.MultiClassClassifier -E 0 -R 2.0 -W weka.classifiers.AdaBoostM1 -- -P 100 -I 50 -S 1 -W weka.classifiers.DecisionStump --
16	10 binary classifiers (one-against-all): 50 iterations of LogitBoost using stumps	weka.classifiers.MultiClassClassifier -E 0 -R 2.0 -W weka.classifiers.LogitBoost -- -P 100 -I 50 -W weka.classifiers.DecisionStump --
17	10 binary classifiers (one-against-all): multilayer perceptron	weka.classifiers.MultiClassClassifier -E 0 -R 2.0 -W weka.classifiers.neural.NeuralNetwork -- -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
18	20 binary classifiers (random code): multilayer perceptron	weka.classifiers.MultiClassClassifier -E 1 -R 2.0 -W weka.classifiers.neural.NeuralNetwork -- -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
19	20 binary classifiers (random code): 100 iterations of AdaBoost using stumps	weka.classifiers.MultiClassClassifier -E 1 -R 2.0 -W weka.classifiers.AdaBoostM1 -- -P 100 -I 100 -S 1 -W weka.classifiers.DecisionStump --
20	20 binary classifiers (random code): 100 iterations of AdaBoost using multilayer perceptrons	weka.classifiers.MultiClassClassifier -E 1 -R 2.0 -W weka.classifiers.AdaBoostM1 -- -P 100 -I 10 -S 1 -W weka.classifiers.neural.NeuralNetwork -- -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Appendix - F1 x F2 plots

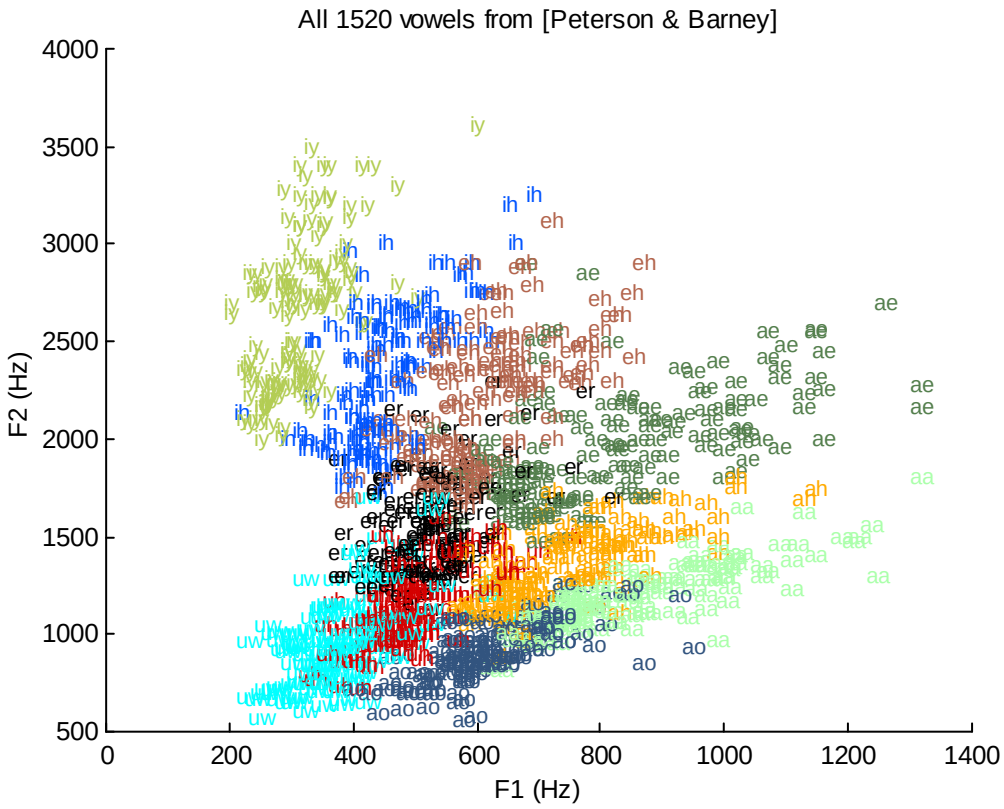

Fig. A.1- F1 x F2 for all 1520 vowels. This plot should be the same but for unknown reason it seems to differ from [6], page 44.

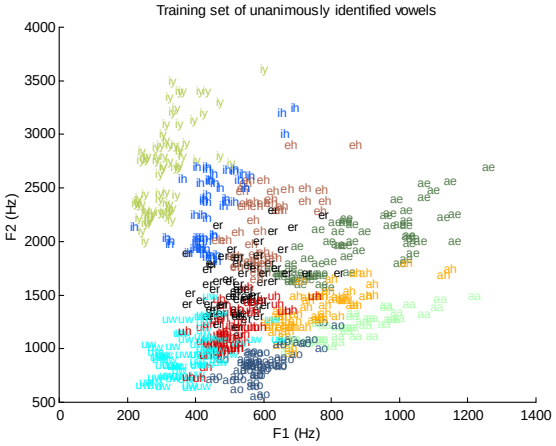

Fig. A.2- F1 x F2 for training set with only unanimously identified vowels.

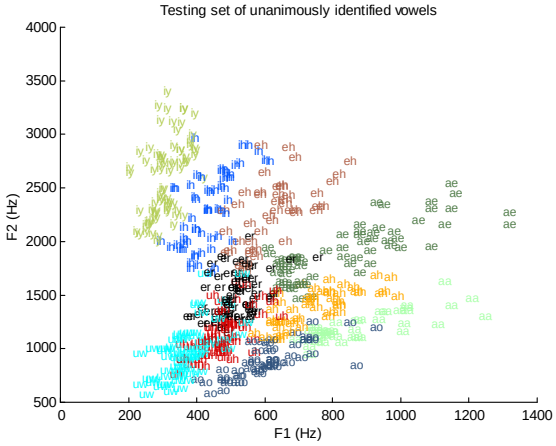

Fig. A.2- F1 x F2 for testing set with only unanimously identified vowels.