Report on TMW coding challenge

The following is a report on the results of my work on the TMW coding challenge. For ease of reference, I have I have pasted the posed tasks and questions into the text below and answered each of them right beneath each item. The original posted tasks and questions are marked in blue text to make it easier to separate them from my answers.

Part 1: Tasks

Task

Your challenge is to extract the audio features you believe are most important to distinguish between child and adult speech. This should be a set of independent features that are robust to who the speaker is and to environmental noise.

Submit your code, the final dataset of extracted features, and answers to the questions below.

My code for part 1 can be found in the following folder in the repo:"..\Part 1\KKJ_Contribution". The final dataset is provided as "features.csv" in the same folder.

Part 1: Questions

Question 1.1

What audio features did you choose to extract and why?

In general I would prefer to let the data to speak for itself and collect an initial broad suite of audio features for closer analysis before focussing on any particular family of features. However, for the purpose of this question the following is what I have chosen to consider:

I am under the assumption that formants or spectral shaping will be important in distinguishing between children and adults as opposed to for instance fundamental frequency, mean frequency and the like given that the fundamental frequency (F0) in the voice of children appear not as distinct from that of women. Moreover, parents often modify their pitch when talking to children which may render the F0 and related measures less robust. On that account I have chosen to extract the following features that provide detailed information about the broad frequency spectrum:

- LPCC (Linear Predictive Coding), *or*
- MFCC (Mel-Frequency Cepstral Coefficients)

For the above features I would analyse what subset (what frequency bans) are in combination most significant (if any) in making a distinction between child and adult voices and thus reduce the number of features to alleviate the computational budget (same is true for the delta features below)

The above can be susceptible to noise and it is possible an alternative may need to be. One might be the Mean Hilbert Envelope Coefficients (MHEC; not extracted here).

The following features were chosen in order to capture temporal changes in the formant structure, given the assumption that this may help in general to distinguish between children and adults and perhaps in specific to capture turn taking, as well as perhaps when both (or more) talkers are active at the same time:

- Delta MFCC
- Delta, delta MFCC
- Spectral entropy
- Spectral flux

In order to better separate when noise is dominant or to signal non-voiced speech I would as a minimum add:

- ZCR (Zero crossing rate)

In real world environments there are of course a lot other sounds than speech which the model will need to be able to handle robustly without getting “confused”. Given that many of the selected features should give a somewhat rich information on spectro-temporal information I assume they should work for classification of such sounds as well, but it is possible that additional or alternative features might be needed.

Question 1.2

Are there any features you intentionally didn't use?

Again, I have chosen to focus on features that capture rich information on the spectrum under the assumption that formant related cues will be key in differentiating between child and adult speech, and avoided more narrow banded features like e.g. fundamental frequency or mean frequency assuming these will be harder to make a distinction from. Normally, however, I would initially include a broader suite of feature to make sure I am correct in my assumptions and to let the data “speak for itself”.

Question 1.3

What steps did you take to transform the original .wav files to prepare for feature extraction?

In the current case I did nothing. It would all depend on what the target device is and the ultimate selection of features are. If low frequency noise is usually prominent one could apply low pass filtering of course. Downsampling should be done to match that of the target device as well. Any known spectral distortions imposed by the recording equipment should be corrected if possible. Similarly, any known spectral shaping of the target device should be implemented in order to transform the data to be as similar to that ultimately collected by the target device.

Question 1.4

How well do you think this audio reflects real-world conditions? What types of sounds might be harder to handle, and what edge cases would you prioritize detecting or filtering first?

In real life conditions there are of course a wide range of things that might render the classification task rather challenging. As is heard in some of the provided recordings there can be a lot of microphone rubbing and tapping noises with a lot of high frequency content. Since we are talking about children, there are also likely to be toy sounds/noises of all kinds present. TV or other media may be playing in the room. Other people might be talking nearby. Children/adults are not taking turns as much as talking on top of each other at times. Low frequency HVAC noise may be present. Outside wind noise might be present and when riding in a car a lot of road noise will be present. In the case of HVAC and car noise a significant amount of nearly inaudible, very low frequencies might be present and cause some interference.

Some simple low or high pass filtering might be an option to reduce some impact (depending on the features used). One could add “Noise” as a category in the model in order to better separate out such cases (as well as other categories if necessary). It might be possible to construct a separate “noise detector” based on fewer and simpler features that could act as a gate to indicate when/if to engage a more sophisticated and computationally demanding model.

In general, it is extremely important to test the model(s) in real life conditions on the actual (or similar prototype) devices in order to evaluate the robustness and performance. More often than not, real-life conditions/data has a different distribution/character than the necessarily limited training data and it is important to catch that early on in order to modify the model and/or how the training data is processed/augmented.

Lastly, I will mention that if it proves very difficult to reliably classify children and adult voices based on audio alone and thus estimate turn taking, it might be a possible solution to place an accelerometer on the child's neck (“band aid”) to indicate when the child is speaking vs when

not. Alternatively, such a construct might be useful when verifying system performance during the research phase.

Question 1.4

Is there anything you would want to add, optimize, or improve if you had more time?

- Look more carefully though the literature to establish a firmer set of features for the task
- Do some initial annotations of the provided audio and look at how well the selected features perform, and re-evaluate my assumption if they appear performed poorly.
- Write up the software to automatically order a long list of features according to their ability to separate the classes.

Question 1.5

If you were to annotate these audio files to train a supervised learning model, what labels would you include? Assume each row of the dataset represents a 10-second audio snippet, and provide an example column header.

I would annotate such that male and female speech were broken out separately even though they will be lumped into the same class in order to be able during an analysis phase to see if there are different overlap between female vs child and male vs child. This might guide feature selection. I would add “background speech” as a separate annotation given that this might result in a challenging interferer.

Finally I would add “Noise” as a category to indicate any other sound. This could in principle be broken out into a number of specific noise classes like “Toy”, “Car”, “Wind”, “Music”, “TV”, etc. depending on how much time it might add and feasible it is at annotation time. This could help in understanding the model’s performance in different environments better, guide feature selection, and help determine if any additional class needs to be taken into account for classification.

Below I have provided an example datasheet of an annotation. The annotated classes are shown in the column. A “*one*” or a “*zero*” indicate if the given class is present within the time frame or not. As is shown, more than one class might be present within the same time frame.

Table 1.5.1

Time frame	Male	Female	Child	Background speech	Noise
1	1	0	0	1	0
2	0	1	0	0	0
3	0	1	1	0	0
4	0	0	1	0	1
5	1	1	0	0	1

Part 2 Tasks

Task 2.1

Determine which properties are statistically significant for determining gender. Report your results.

Histograms for each gender and feature are shown in the figure below. The feature histograms are displayed in order of the divergence between the female and male distributions (see below). The female histograms are shown in red and the male ones in blue. The feature name is indicated as the x-axis label. All histograms have been normalized according to probability.

For some of the features a log transform (log2) were done in order to both attempt to make the distributions more normal shaped and to better be able to visually inspect the data. In those cases the axis and associated text are shown in red.

As it turned out, the features “meanfreq” and “centroid” were exact copies of each other and I therefore removed the “centroid” feature from the analysis.

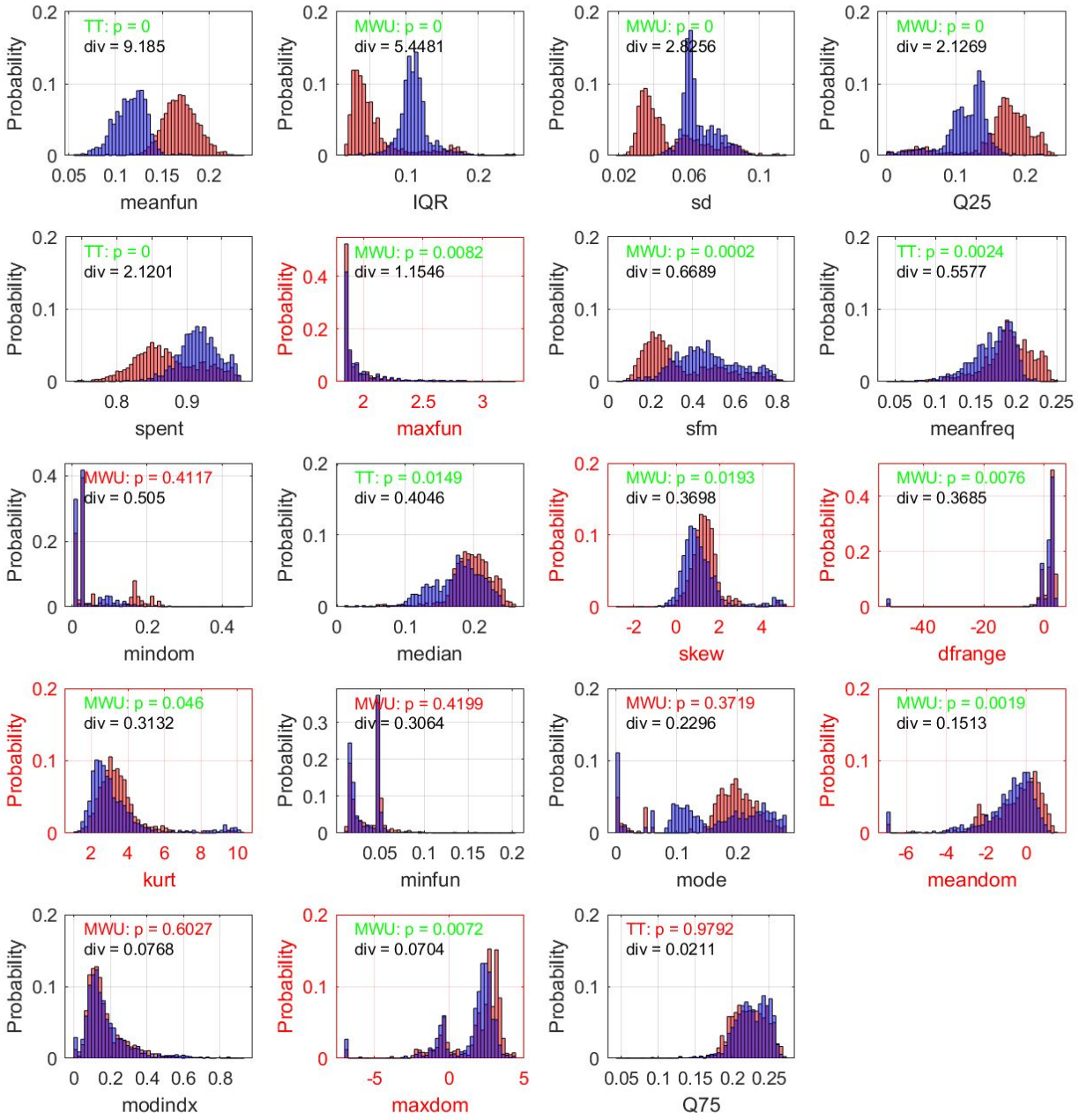
A text label top right in each panel shows the p-value of a statistical test, shown in green if a statistical difference at the alpha level (0.05) were found, and red if not. A crude test for normality of the data was performed by a mean-to-median ratio. If the mean was within +/- 5% of the median the data was taken as being normally distributed and a t-test for performed to test for a significant difference between the female and male means. If the mean was more than 5% different from the median, the data was taken as

not being normally distributed and a non-parametric Mann-Whitney U-test was performed instead.

The provided data had a sample size of N=1584 which results in a large statistical power and the ability to find a statistical difference even for a very small effect size. Given that the classifier will not have that many samples available at its decision time, it is more meaningful to look at how many samples are necessary to establish a significant difference with a power of 80% and alpha at 0.05. The panels therefore show the number of samples necessary in parentheses after the p-values.

In order to get an objective measure of how different the female and male distributions were for a given feature, their divergence were calculated. The divergence is closely correlated to the statistical significance and the number of samples, but is better at handling non-normal distributions, and is probably the best overall measure. The features in the figure have therefore been sorted in order of the size of the divergence.

Figure 2.1.1


TT: p = 0
div = 9.185
meanfun
MWU: p = 0
div = 5.4481
IQR
MWU: p = 0
div = 2.8256
sd
MWU: p = 0
div = 2.1269
Q25
TT: p = 0
div = 2.1201
spent
MWU: p = 0.0082
div = 1.1546
maxfun
MWU: p = 0.0002
div = 0.6689
sfm
TT: p = 0.0024
div = 0.5577
meanfreq
MWU: p = 0.4117
div = 0.505
mindom
TT: p = 0.0149
div = 0.4046
median
MWU: p = 0.0193
div = 0.3698
skew
MWU: p = 0.0076
div = 0.3685
dfrange
MWU: p = 0.046
div = 0.3132
kurt
MWU: p = 0.4199
div = 0.3064
minfun
MWU: p = 0.3719
div = 0.2296
mode
MWU: p = 0.0019
div = 0.1513
meandom
MWU: p = 0.6027
div = 0.0768
modindx
MWU: p = 0.0072
div = 0.0704
maxdom
TT: p = 0.9792
div = 0.0211
Q75
Probability

Task 2.2

Build a full logistic regression model and evaluate its performance. Report your results, including accuracy on the training and test set.

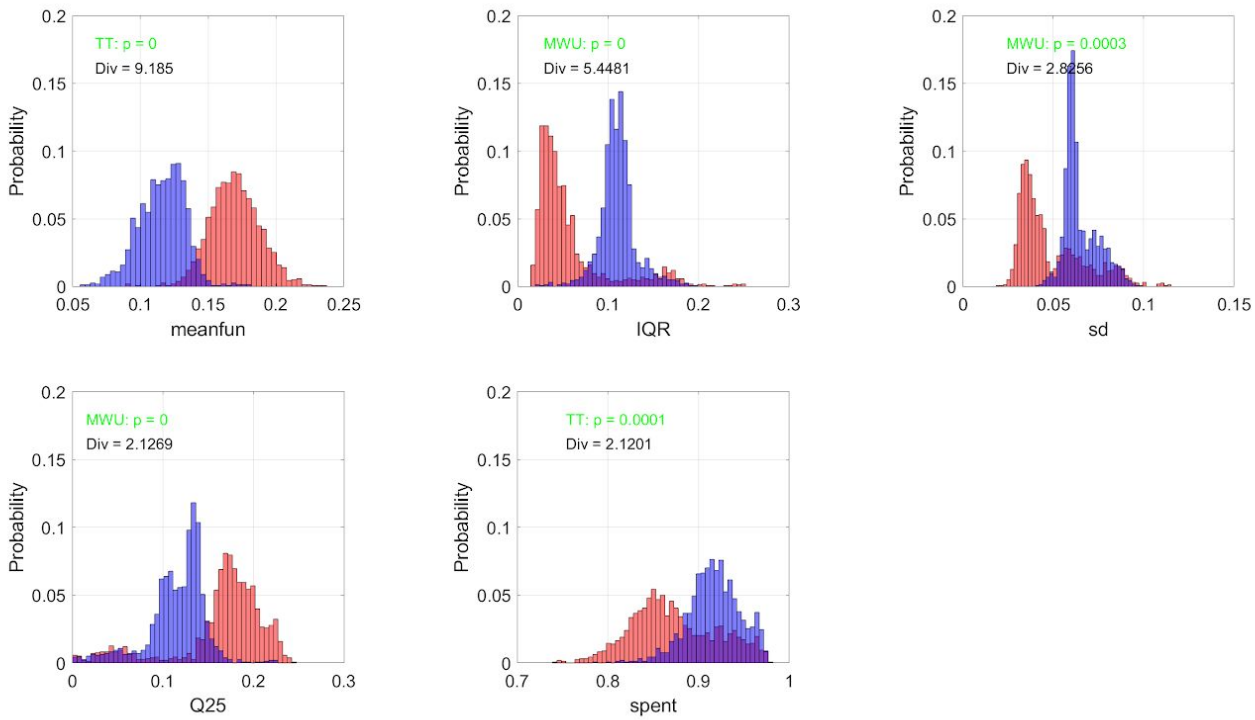
Selecting features

Selecting the most discriminant subset

Before building the logistic model, a subset of features were selected that seemed the most promising. At first I selected a subset of features that needed 25 or less samples in order to show a significant difference and had a divergence of 2 or more. This is more or less arbitrary and could be changed depending on the data and the specifics of the problem and the computational constraints of the device.

The resulting 5 features are shown in figure 2.2.1 below

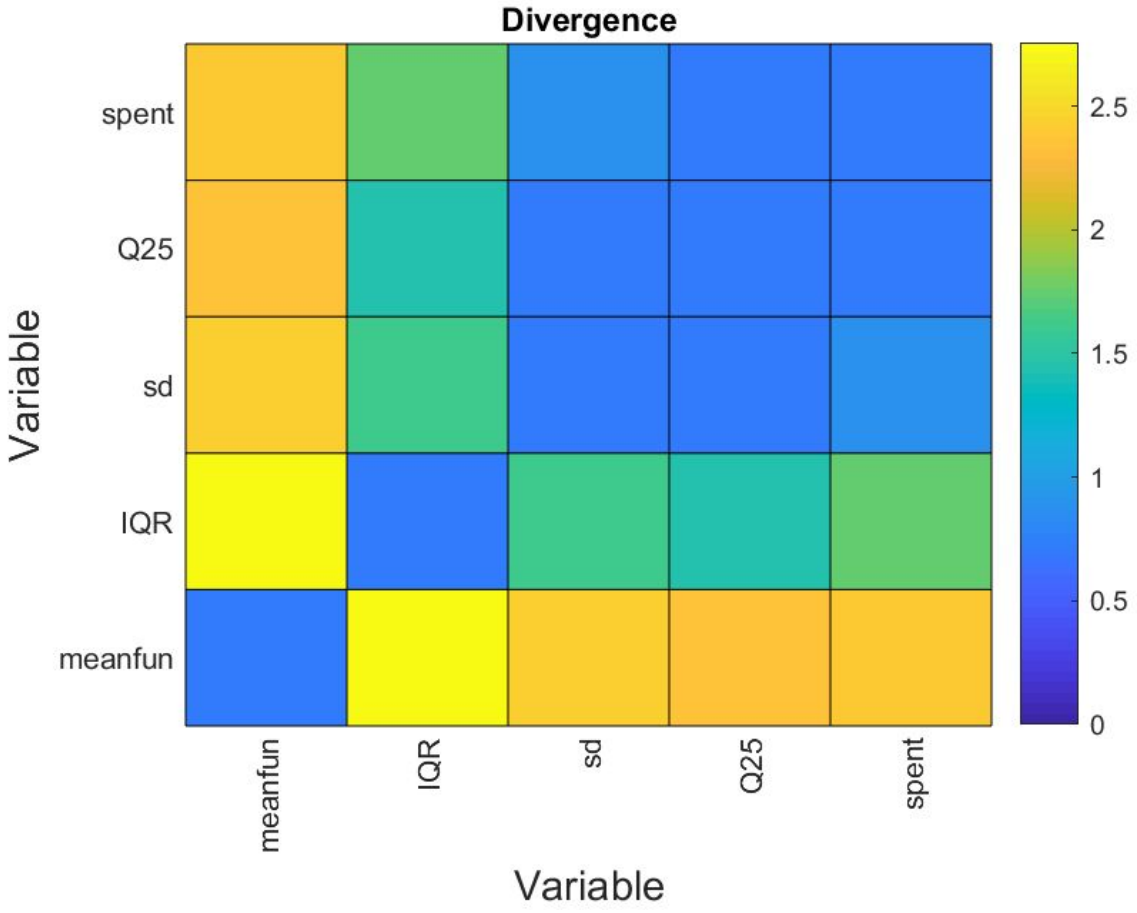
Figure 2.2.1



Pairwise divergence

In order to examine what features best combine I calculated the divergence of the pairwise combination of each feature as shown in figure 2.2.2 below. The natural logarithm was applied to the divergence for display purposes in order to best indicate the pattern across combinations though the color coding. As is seen, and not surprisingly, *meanfun* and *IQR* for the combination with the highest divergence.

Figure 2.2.2

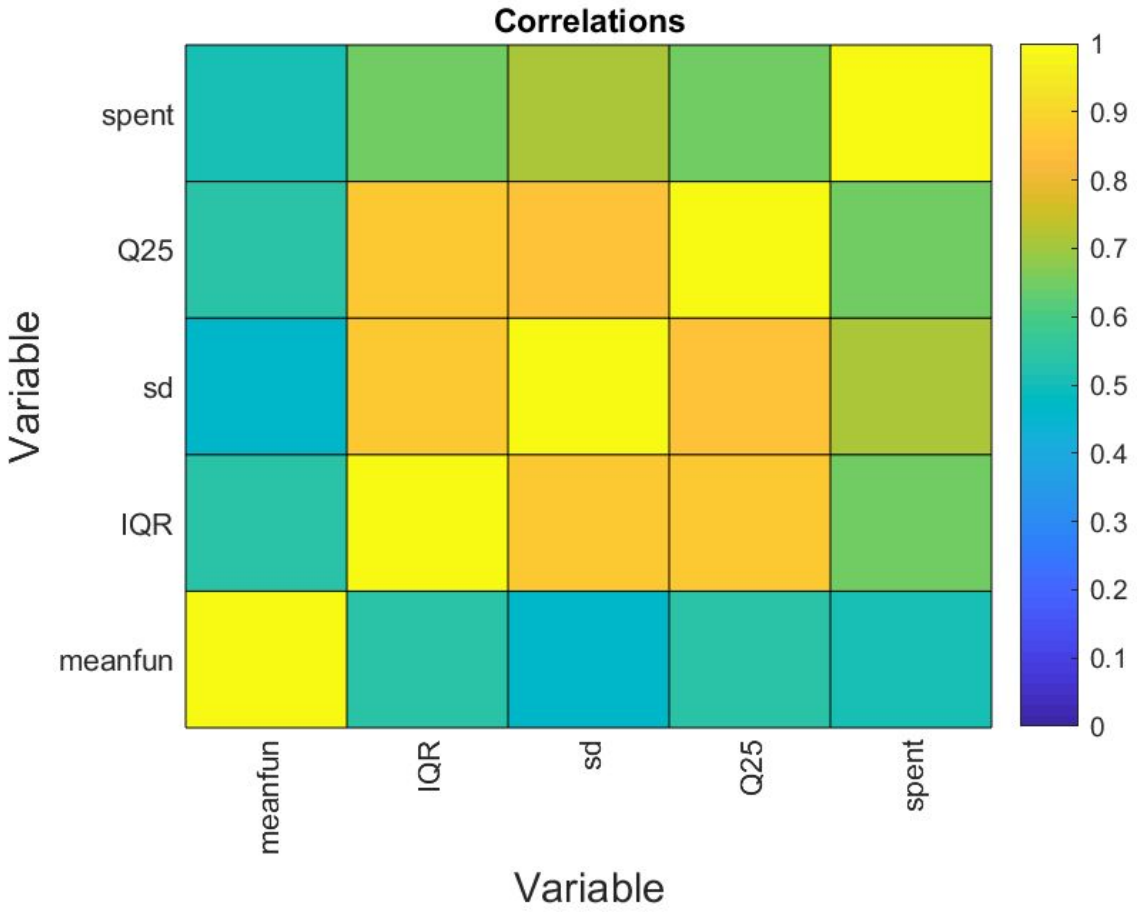


Determining feature correlations

If two or more features are correlated little or nothing may be gained to include it into the model and it would therefore be a waste of computations on a low resource device to include all and one should pick the one with the highest divergence or lowest computations needed. Below in figure 2.2.2. is shown the correlation between each pair of the selected subset of features. Only

the absolute value of the correlation is selected since we do not care about the direction of the correlation.

Figure 2.2.3

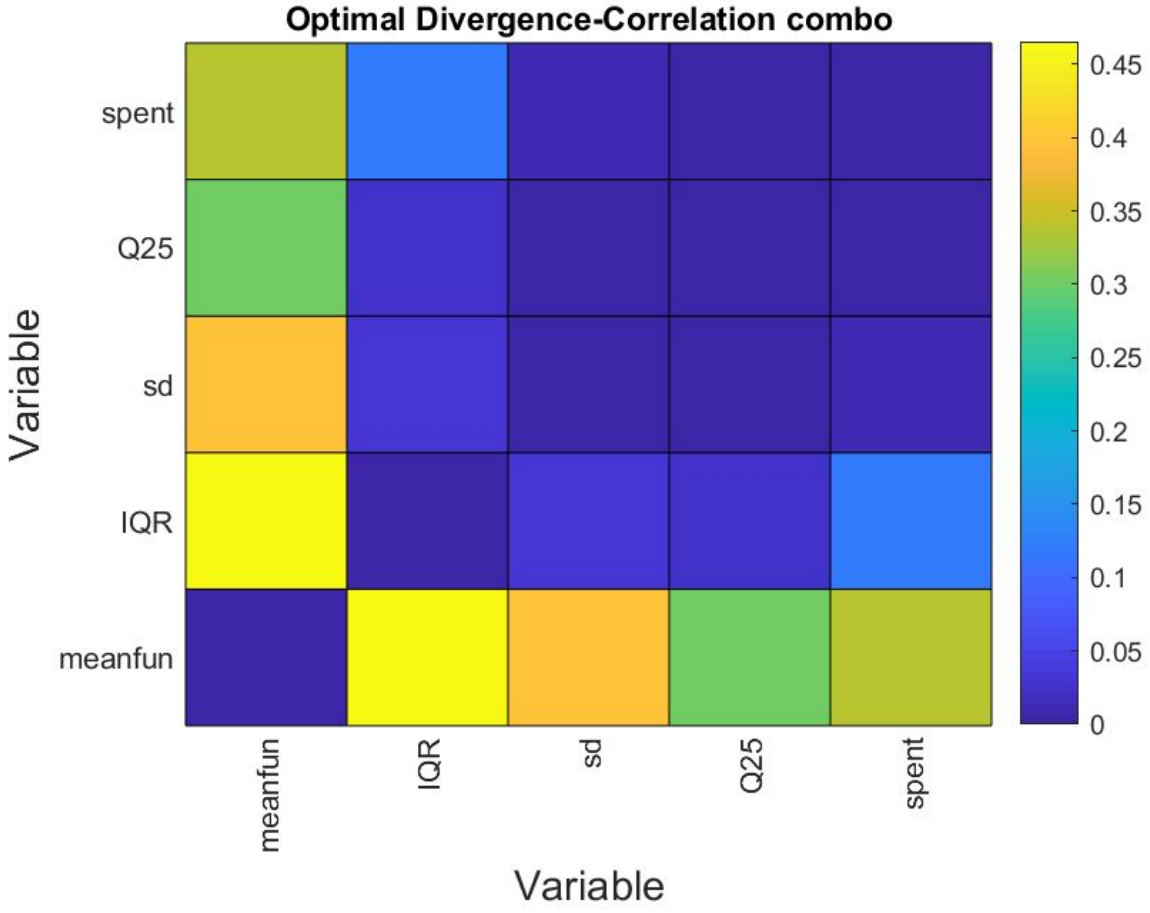


Optimal divergence-correlation combo

Finally, to find the best combination of the subset of features I calculated the optimal divergence-correlation combinations by multiplying the inverse of the correlations (1-r) with the divergence values (normalized btw 0 and 1). Doing so, the feature pairs with the highest divergence and the least correlation will stand out. As is seen in figure 2.2.3 below the best combo is *meanfun* and *IQR*. Next, now ignoring the *meanfun* feature (first column, or last row), we look for the next best combo which can be seen to be the *sp.ent* (spectral entropy). After that nothing stands out, and we therefore build the classifier based on the following 3 features:

1. Meanfun
2. IQR
3. Sp.ent

Figure 2.2.3



Logistic regression

As was concluded above the three features *meanfun*, *IQR*, and *sp.ent* were found to the best and hopefully sufficient combination of features to use in order to discriminate between female and male voices.

A logistic regression model was built and trained on the three features using Keras/Tensorflow. The script is called “logistic_regr.py” and is located in “..\Part 2\KKJ_Contribution\Model” . The data were split into a training set (70%), an evaluation set (20%), and a test set (10%). Please refer to the script for details on the training hyper parameters.

Table 2.2.1 below is showing the resulting accuracy on the test set. As can be seen the initial model with the selected features gives 95% accuracy on the test set. Since the IQR distribution is a little more complex than a normal distribution I attempted with applying feature crossing to see if it helped the already high performance. This increased the accuracy by 1% to 96 which

isn't much and probably not significant. In fact, using the meanfun feature alone result in 96% test set accuracy as can be seen in the table.

Given the high performance on the single meanfun feature, there is presumably rather little to gain in building a more powerful model. Just to see I expanded the model to be a 3 hidden layer dense neural network with 10 units each. This resulted in a marginal increase of 2% to 98% accuracy. Whether this added complexity is worth it or not depends of course on how critical the accuracy of the classification is and how much computational power is available.

Table 2.2.1

Model	**Features**	**Comments**	**Feature crossing**	**Test set accuracy (%)**
Logistic regression	*meanfun*	3 most divergent features and least correlated features	No	95
	IQR		Yes	96
	spent			
Logistic regression	*meanfun*		No	96
Dense NN with 3 hidden layers of 10 units each	*All*			98

Task 2.3

Train any model of your choice to achieve an accuracy above 80%.

As described above, meanfun alone already gives 96% accuracy on the test set, so there is already a good performance from that one feature alone and only room for a marginal improvement.

I am guessing it wasn't intentional to include a single feature with such a high separability between classes. If I assume that this was not the case, I would use the same procedure above (which can be automated) to isolate the optimal features. If a simple logistic regression was showing less than 80% accuracy and I was trying to get above, a first natural step would be to expand the logistic regression to a dense neural network and see how many layers and units in each layer that would be necessary to get above 80% (if possible). One could apply hyper parameter tuning with a certain

computational constraint on the network architecture in order to find the optimal design within the computational budget.

If a neural network within the computational budget wasn't resulting in a high enough accuracy, due perhaps to complicated feature distributions and relations, a next step could be to attempt building, for instance, a Gaussian Mixture Model which has relatively low computational complexity and yet able to model complex feature distributions.

Part 2 Questions

Question 2.1

What changes did you make in Section 3 to increase the accuracy of your ML model? Include a description of the features you selected or created, parameters you tuned, and a discussion of tradeoffs.

Please see my discussion in the "Task 2.3" on increase of model accuracy and the "Selecting features" section for features I selected. Above I also discussed some tradeoffs. The computational requirements are likely increasing as you are trying to achieve higher accuracy which may not be supported on the device the model is supposed to be running on.

Question 2.2

Is there anything you would want to add, optimize, or improve if you had more time?

With the given data set there is not much left to optimize per se. A logistic regression model is already at a low computational cost and it performed rather well in this case. If it had not and it turned out to be hard to build a model within the computational budget that was based on the "static" distribution of features, I would attempt to build a model that took the temporal pattern of the feature values into account. This is my experience from previous work, that the information in the temporal pattern was a lot more useful than the more static picture. If computational affordable, it could be a "tiny" recurrent neural network would work, or some simpler integration of temporal patterns might be possible. .

Question 2.3

How might the model you built perform in the real world? What technical or ethical considerations would you weigh when deciding whether to deploy such a system?

This would to a large extent depend on the collected data set and how representative it is of the real world. It is also of great importance what equipment was used to record the data and how it compares to how the deployed device records data. Any difference in spectral response, bit depth, sampling rate, etc may render the model unable to make correct classifications.

It is important that the data cover as much of a variety as practically possible encountered in the real world and if the data is not directly recorded through the planned device to transform the data to be representative of the device.

Once that is taken into account, one can apply data augmentation to attempt to make the model more robust by e.g. adding noise, reverb/room responses, spectral distortions, and more.

From an ethical point, it is optimal if either the classifications are done on the device if raw audio is used as model input such that no identifiable information is sent to the cloud. As such, low level audio features are good to use since they contain fairly anonymous data.

Question 2.4

What would you do differently if you were trying to detect adult vs. child speech instead of male vs. female voices?

I would pay more attention to features that are informative about the overall spectral shaping assuming that features based on fundamental/mean frequency and the like are going to be tricky to use given that children and female fundamental frequencies can be rather similar, especially when mothers often changes their voice pitch when talking with their children. Features with information on overall spectral shaping, on the other hand, should give information on the formants which should be more divergent given the smaller size of children (this depends on their age of course).