ACOUSTIC INDICES PT 2

Images used

For each spectrogram, 4 differentials were taken. These 4 channels were saved in the same dictionary format.
Dim1: spec_idx
Dim2: freq_steps
Dim3: time_steps
Dim4: ch (0, 1, 2, or 3 for the 4 channels)

Files

The files for each class - Good, Bad, Human, Maybe - were loaded as .npz files, uncompressed, and reloaded as 'good_preprocessed_uncompressed.npz' and so on - https://github.com/veenavijai/birds-mel-veena/blob/master/Acoustic%20Indices/pre_unc_load2.py

Function modifications

- calls_orig vs calls
- ch added to rename columns which is useful while later joining dataframes for different channels
- ACI implementation was un-vectorized - many of the rows in the differentials are 0, and to avoid divide by 0 error
- NaN values were removed for each and every feature using pd.notnull
    - Runtime warning of true divide pops up - needs to be checked
- The table looked like this after joining:

	ACI1	ADI1	ADI_even1	SH1	NDSI1	Class1	ACI2	ADI2	ADI_even2	SH2	...	ADI_even3
0	80.243289	-1.918789	-1.881208	0.233704	0.716630	Good	73.463291	-1.904153	-1.699613	0.265937	...	-1.045067
1	46.404186	-1.777213	-1.836112	0.354649	0.712668	Good	42.930161	-1.696032	-1.593020	0.372118	...	-0.927017
2	83.291242	-1.738360	-1.666841	1.067125	0.384758	Good	68.120467	-1.834686	-1.399250	0.855138	...	-0.878492
3	75.370427	-1.776804	-1.708756	1.078624	0.402326	Good	66.747861	-1.781149	-1.552843	0.850100	...	-1.047065
4	55.620199	-1.939896	-1.565449	0.502348	0.340162	Good	39.435161	-1.816832	-1.500386	0.314896	...	-1.292281

- 5 rows × 24 columns
- Now in one function - converts all data to one dataframe and prints its scatterplot matrix
- NDSI implementation - removed hard coding of freq bins

Experiments

Note: Conducted after joining all rows of all 5 acoustic indices and 4 channels and removing all rows where even one column has an NaN value.

For each spectrogram -

1. Normal log spec - done
2. Taking average of all indices - 4 channels - done
3. Taking max of each index - 4 channels - done
4. Take max pixel value in each spec for the 4 channels and get a new spec -
    a. https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.maximum.html
    b. Calc_plot modified with ch==4 condition
5. Take average pixel value in each spec - done
6. 2 with 2 channels - done
7. 3 with 2 channels - done
8. 4 with 2 channels - done
9. 5 with 2 channels - done

TODO

10. Thresholds with normal log spec
11. Weighted average of 4 channels with more weight to 1 & 2
12. Normalize both original spec and 4 channel spec to between 0 and +1 and repeat 1-11 (have a normalize parameter as 0 or 1 in all functions)
13. Try weighted average with 0.5 weightage to original and 0.25 to ch1, 0.25 to ch2

TODO: Metrics to quantify how good the clusters are

Scatter plots

https://github.com/veenavijai/birds-mel-veena/tree/master/Acoustic%20Indices/4_channel_scatter

Analysis of plots

Plots with good separation:

1. Scatter_good_bad:

a. ACI vs ADI: Bad is mostly ACI > -3.5 and ADI < -1.9430
b. ACI vs SH: Bad is mostly ACI > -3.5 and SH > 3.53
c. ADI_even vs ADI: Bad is mostly ADI_even < -1.8 and ADI < -1.9430
d. SH vs ADI: Bad is mostly SH > 3.52 and ADI < -1.9430
e. NDSI vs ADI: Bad is mostly 0.40 < NDSI < 0.55 and ADI < -1.9430