Iterative Feature Normalisation - Report

Brihi Joshi

1 TL;DR

- Iterative Feature Normalisation (IFN) works better than the global normalisation techniques for emotion recognition from speech data.
- Most of the **trends (not numerical values)** on a different dataset like RAVDESS match with that in the original paper [1], which supports their findings.
- However, the two major discrepancies as highlighted in Section 4.3 are justified with the help of the author's comments in the paper and the analysis of the data at hand. It appears that IFN is **not generalisable** to smaller datasets with lesser utterances per speaker.

2 Introduction

Iterative Feature Normalisation (IFN) was proposed in [1] for the task of emotional speech recognition. The primary idea behind the method is simple – in order to reduce speaker variability while detecting emotional speech, the neutral 'style' of each speaker is normalised using reference neutral speech. This can be seen as *leveling* the neutral speech of each speaker, which in turn highlights the emotional 'style' of the emotional speeches, while at the same time, minimizing inter-speaker differences.

3 Setup

IFN requires **two** corpora - a **reference** corpus and an **emotional** corpus. The **reference** corpus is used to calculate the normalisation constants that remove speaker variability. It consists of only neutral utterences. The **emotional** corpus is used to perform the actual classification task. The problem statement setup is only for **binary classification** into neutral and emotional speech.

3.1 Dataset

Since the datasets used in the original work ([1]) are not available, the dataset used for this experiment is the RAVDESS dataset [2]. It contains 2 unique textual utterances spoken in 8 different emotional categories, including neutral, spoken by 24 actors. Each such utterence is

spoken in 2 different intensities and each utterance is repeated twice. Since for the purpose of this assignment, the reference corpus was not provided separately, the RAVDESS data itself was split into the two different corpora.

3.2 Dataset Splits

- **Reference Corpus** - 1 neutral sample from each of the 24 actors is separated out at the beginning of the experiment. This makes up the 24 utterances in the reference corpus.
- **Emotional Corpus** - The rest of the RAVDESS data makes up the emotional corpus. Stratified Sample operation is then applied – to make sure for each speaker, equal number of neutral and emotional instances are present. After sampling, the data is split into train and test sets (the details for which are given in the pseudocode.

3.3 Feature Extraction

The features that are extracted are same as that in [1]. They are as follows -

- **SQ25** - 1st Quartile of the F0 contour
- **SQ75** - 3rd Quartile of the F0 contour
- **IDR** - Inter-quartile range of the F0 contour
- **F0 median** - Median of the F0 contour
- **sdmedian** - Median of the gradient of the F0 contour
- **SVMeanRange** - Mean range of the voiced region of the F0 contour
- **SVMaxCurv** - Maximum of the voiced region curvature of the F0 contour

4 Algorithm

4.1 Pseudocode

Step I

The first step is to separate out the reference corpus and the emotional corpus. Then, the following steps are taken -

1. Extract the 7 features listed above on the reference corpus.
2. Train 7 **Gaussian Mixture Models** (one for each feature) from the features extracted above.

Step II - IFN

Then following steps are used to build the implementation of IFN.

1. Sample a stratified train and test split from the emotional corpus.
2. Calculate $S^s_{F0} = \frac{F0_{ref}}{F0^s_{neu}}$. Here, $F0_{ref}$ is the average F0 of the reference corpus. For every speaker s, $F0^s_{neu}$ is the average F0 of the neutral speeches of the emotional corpus.
3. Now, on the entire emotional corpus, **scale** the F0 contour with S^s_{F0} for each speaker. By scaling, we mean the division operation. We thus get a modified/scaled F0 contour that we call $F0_{mod}$.
4. On $F0_{mod}$, do the following -
   (a) Calculate the 7 features described above
   (b) On the trained GMMs in Step I, infer the features to obtain **likelihood values**. This will result in 7 likelihood values for each utterance.
5. On these likelihood values, train a **Linear Discriminant Analysis** Model. This will result in a predicted probability value for each class.
6. The stopping criteria of the training is as follows -
   (a) If the predicted probability of the neutral class for an utterance is **greater than** 0.7, classify it as neutral. Else, it is classified as emotional.
   (b) We will not have a predicted set of labels.
   (c) For each speaker, find the percentage of utterances predicted as neutral (let's say this is p). If p is **greater than** 20%, we can continue. If not, we need to ensure atleast 20% of the files are in the neutral class. So we order the utterances predicted as emotional by the predicted probability for the neutral class and keep removing the utterance with the highest predicted probability, till we reach the 20% value, and adding them to the neutral class. This would give us the final predicted set of labels for this iteration.
   (d) If the percentage of labels changed (this would not be calculated for the 1st iteration) is **greater than** 5%, we again go to step 1.
   (e) Else we stop the algorithm.

This entire run of IFN is performed 400 times, as given in [1] for different random splits of train and test set. This is done to prove the **Robustness** of the method.

4.2 Hyper-parameters and other setups

All implementations are done in Python. For extracting audio features, a freely available Python API for Praat [3] called Parselmouth [4] is used.

Hyper-parameter settings chosen are -

1. For the classification threshold, instead of **0.7**, I have chosen **0.5**. This is because given that RAVDESS is a small dataset, in comparison to those given in [1], the number of neutral samples are also less. While taking a threshold of 0.7, the model was not predicting any neutral samples. As the paper states - *Preliminary analysis suggested that setting this threshold equal to* 0.7 *yields to enough neutral samples, maintaining an acceptable accuracy rate.* Similarly, grid search methods showed that 0.5 is an acceptable hyper-parameter for our dataset.

2. An upper-cap of 100 and lower-cap 0.00001 is set on the scaling factor S^s_{F0}, to prevent it from blowing up or vanishing.

5 Results and Analysis

5.1 Comparison with Variants and Ablation

Four variations of IFN are attempted, given below -

- **Optimal Initialisation** - The initialisation of the scaling factor S^s_{F0} is from the ground truth labels.
- **Without Normalisation** - The F0 contour is not normalised before inferring from the GMM
- **Global Normalisation** - The scaling factor for each speaker S^s_{F0} is calculated by both neutral and emotional speeches, not only neutral (like in IFN).
- **Ablation - Only LDC** - The features are directly fed into the LDC for a vanilla classification setup.

The results for these variants are given in Table1.

Method	**Precision**	**Recall**	**F1**	**Accuracy**
Optimal	0.762	0.599	0.521	0.704
Without	0.772	0.665	0.625	0.75
Global	0.680	0.609	0.563	0.604
IFN	0.685	0.617	0.578	0.625
Ablation	0.688	0.669	0.660	0.645

Table 1: Classification results for IFN, its variations and ablation study.

5.1.1 Comparison with the original trends

As it is visible from the Table 1, the general trend as produced by my implementation is - **Without Normalisation > Optimal Initialisation > Ablation > IFN > Global Normalisation**. A comparison with the author's trend in [1] is as follows -

- The trend **Optimal Initialisation > IFN > Global Normalisation** holds for both the implementations. This shows that normalising just from the neutral samples gives better results that when normalised from all the samples.
- The trend with **does not match** is the one where **Without Normalisation > all variants**. A reason for why this might be happening is given in Section 4.3.

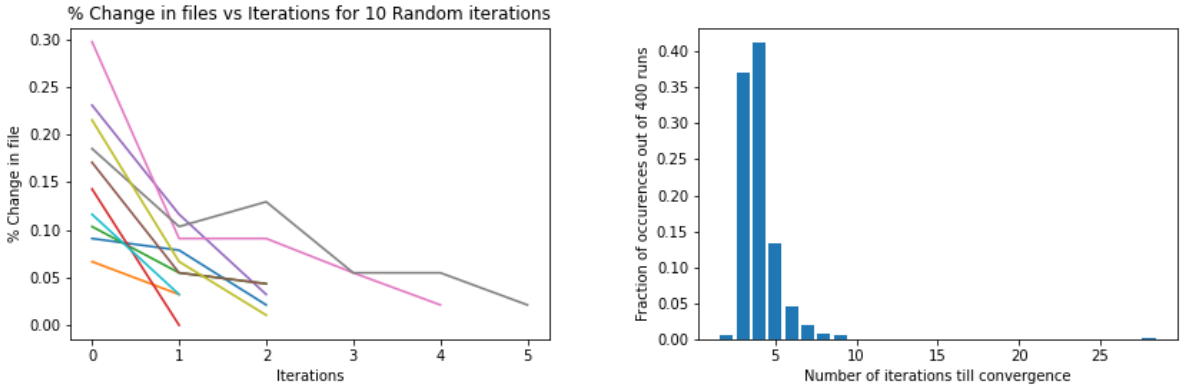
5.2 Convergence Analysis

Figure 1 shows that IFN converges in finite number of steps – The maximum number of iterations taken to converge were **28**.

Figure 1(a) shows that the % of shift in labels (or as the authors call it, files) decreases as the iterations in IFN proceed. 10 random runs out of the 400 runs were plotted and it is seen that the graph is decreasing in nature. As soon as the % shift in labels is below 5%, the algorithm is terminated. This also proves the validity of the stopping criteria.

Figure 1(b) shows that most of the IFN runs (out of 400) finished before 5 iterations. This shows that IFN is fast in converging.

These convergence trends replicate those shown in [1].



(a) % Change in files vs iterations for 10 Random iterations

(b) Bar plot of the number of iterations taken in different IFN Runs

Figure 1: Convergence analysis as shown on IFN

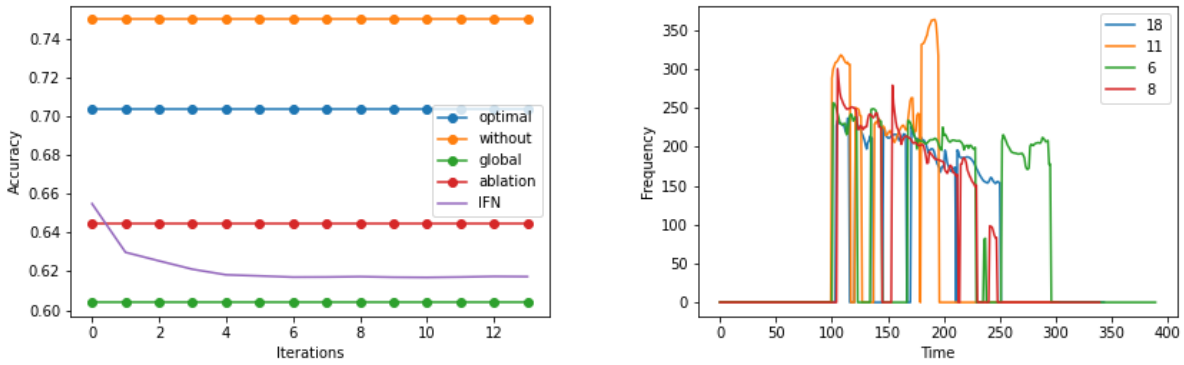
5.3 Why are the trends different from those in the original work?

While collecting the results in 1, one thing that was noted was that both **Without Normalisation** and **Ablative** variations provide better performance than IFN. On investigating why this might happen - I checked whether there actually exists stark speaker variability in the RAVDESS dataset. Figure 2 is a plot of the F0 Contours of 4 randomly chosen speakers, and within those randomly chosen speakers, randomly chosen neutral speech from 2 speakers and emotional speech from the other 2 speakers were taken. An interesting observation from the F0 plots is that - **RAVDESS doesn't have very prominent speaker variability**.

The authors state that the main motivation for creating IFN was to eliminate speaker variability while detecting emotions. Hence, IFN, or any normalisation based method, is actually not creating much difference in performance, and in hindsight, might also normalise some prosodic features which might be key in detecting emotions. This was one conclusion as to why the **Without Normalisation** method performs best in this case.

Other possible reasons for the difference in results can be due to -

- Data is very small, and is not of the scale of the original data, making it hard to replicate the results.
- Figure 2(a) shows the **Accuracy vs. Iterations** plot for the different versions of the model. As it can be seen that the accuracy trend for IFN doesn't follow that of the paper – initially increasing and then dipping. The reason for this is provided by the authors itself - *The improvement in the performance for the EMO-DB database is not as impressive as the improvement in other databases. Figure 3-b even shows that the accuracy decreases in early iteration, until it finally converges at* 73.6%. *This is the database with the lowest average number of samples per speaker (53.5), which may explain this result.* Since the test set while working on RAVDESS has **2 samples per speaker**, this reducing trend is observed, as written by the authors!



(a) Accuracy vs. Iterations plot for different models

(b) Inter speaker variability for Neutral and Emotional classes

Figure 2: Different trends for this experiment as compared to [1]

References

[1] C. Busso, A. Metallinou, and S. S. Narayanan. Iterative feature normalization for emotional speech detection. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5692–5695, 2011.

[2] Steven R. Livingstone and Frank A. Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PLOS ONE*, 13(5):1–35, 05 2018.

[3] Paul Boersma and David Weenink. Praat: doing phonetics by computer [Computer program]. Version 6.0.37, retrieved 3 February 2018 http://www.praat.org/, 2018.

[4] Yannick Jadoul, Bill Thompson, and Bart de Boer. Introducing Parselmouth: A Python interface to Praat. *Journal of Phonetics*, 71:1–15, 2018.