Free Spoken Digit Dataset Classification Problem

Flavio Giobergia
Politecnico di Torino
sXXXXXX
flavio.giobergia@polito.it

***Abstract*—In this report we introduce a possible approach to the *Free Spoken Digit Dataset* classification problem. In particular, the proposed approach consists in extracting the spectrogram of each audio signal and split it into a specific number of chunks. For each chunk, various summary statistics are computed and used as input for two classification models. The proposed approach outperforms a naive baseline defined for the problem and obtains overall satisfactory results.**

I. Problem overview

The proposed competition is a classification problem on the *Free Spoken Digit Dataset*, a collection of audio recordings of utterances of digits ("zero" to "nine") from different people. The goal of this competition is to correctly identify the digit being uttered in each recording. The dataset is divided into two parts:

- a *development* set, containing 1,500 recordings for which a label
- an *evaluation* set, comprised of 500 recordings.

We will need to use the development set to build a classification model to correctly label the points in the evaluation set.

We can make some considerations based on the development set. First, the problem is well-balanced: for each of the 10 classes (0 to 9), there are 150 recordings. Second, the sample width is of 16 bits for all files. Third, all recordings have been sampled at a frequency of 8 kHz. In telephony, the usable voice-frequency band ranges from approximately 300 Hz to 3400 Hz [1]. Due to Nyquist-Shannon sampling theorem, 8 kHz is therefore a reasonable sampling frequency for these recordings (though only by a small margin).

While all signals have been sampled at the same rate, recordings durations differ. Figure 1 shows how such durations are distributed. There are some outliers that have a length that significantly differs from the mean duration (0.4 seconds). Upon a manual inspection, those signals contain some silence following the utterance and should not pose any particular problem. However, since the entries have varying lengths, we needed to devise some approach to extract the same number of features from all data points (since most classification models require a fixed number of features).

To get a better understanding of the type of data at hand, we can inspect some signals visually, both in time and in frequency domains. These two domains introduce useful and complementary information on each signal. Figure 2 shows one signal in the time domain. We can immediately notice that there can be some pauses before and/or after the utterance.

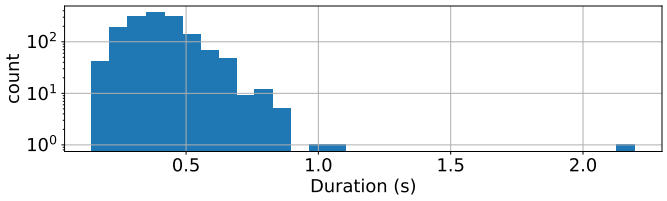


Fig. 1: Distribution of the durations of the recordings

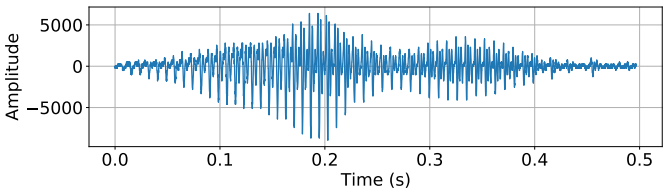


Fig. 2: Representation of a recording in the time domain

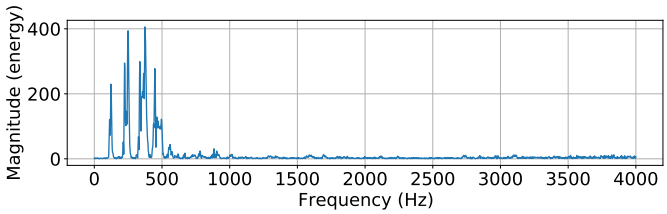
Also, as expected, the amplitudes are bound in the range of values [-32,768, 32,767] (16 bits per sample).

Figures 3a and 3b show the signal in the frequency domain (without a with a decibel scale). From this, we can observe that some frequencies carry more energy than others. The linear scale shows a mostly flat plot with only a limited number of high-energy frequencies. The decibel scale, on the other hand, is a logarithmic scale and highlights a wider range of involved frequencies. Since the human perception of sounds is logarithmic in nature (rather than linear) [2], we can consider the dB scale to be a more meaningful representation than the linear one. We will adopt this representation during the preprocessing phase.

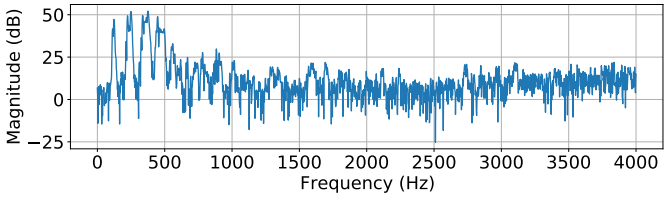
II. Proposed approach

A. Data preprocessing

We have seen that both time and frequency domains contain useful information regarding the recordings. We can leverage both by using the spectrogram of each signal. An example of a spectrogram of a signal is shown in Figure 4. A spectrogram can be seen as an $N \times M$ matrix (N frequency bins and M time bins). We can divide this matrix into n_ρ rows and n_γ columns. This produces $n_\rho \cdot n_\gamma$ blocks. For each block, then, we can compute some summary statistics (in this case, mean and standard deviation). Figure 5 exemplifies how these features are extracted from a single block.



(a) Linear scale



(b) dB scale

Fig. 3: Magnitude of a recording (linear and log scales)

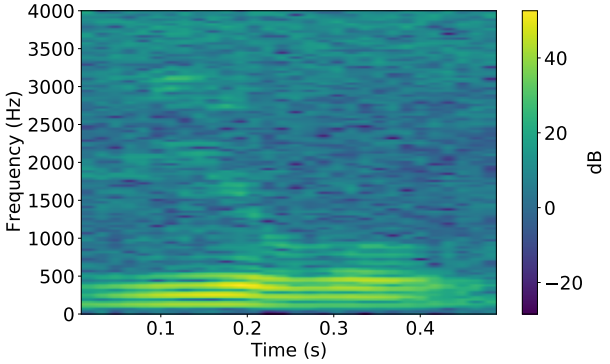


Fig. 4: Spectrogram of a signal

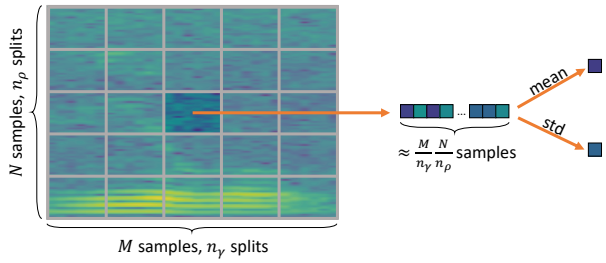


Fig. 5: Example of the extraction of mean and standard deviation for one of the blocks into which the spectrogram is split. Assuming that the spectrogram is represented by an $N \times M$ matrix and that $n_\rho = n_\gamma = 5$ splits are done along each axis

Defining our features as explained has the following benefits:

- *Reduction of the number of features:* if we select a number of blocks of $N \cdot M$, we get the same number of features as produced by the spectrogram. By reducing the number of blocks, the number of features is also reduced
- *Generation of a uniform number of features:* if we select a number of blocks that is not a function of N and, in particular, of M (since M varies across signals), we can generate the same number of features for all signals

The implication of selecting a number of features that does not depend on M is that blocks extracted from signals of different length will contain a different number of elements. This in turn means that we are "squeezing" (or stretching) the signals in time. Similar considerations would apply for the frequency dimension if different sampling frequencies were used.

We can consider n_ρ and n_γ as two hyperparameters of our preprocessing step. Section II-C explores how we can define meaningful values for n_ρ and n_γ.

The statistics extracted from each block are representative of a specific range of frequencies, as used at a specific moment of the recording.

B. Model selection

The following algorithms have been tested:

- *Random forest*: this algorithm leverages multiple decision trees (trained on various subsets of data and features) to make predictions. This typically avoids the overfitting problems of decision trees but still maintaining some degree of interpretability. The performance of a random forest scales with the number of estimators (up to a certain point [3]) and can be tuned on a similar set of parameters as a single decision trees. Random forests, like decision trees, work on one feature at a time, so no normaliziation is necessary. We chose to use random forests as they have been shown to perform well on audio signal classification problems [4]
- *SVM*: this model has also been shown to work well on audio signals [5]. This algorithm applies a (possibly non-linear) transformation to the data points and identifies the maximum-margin hyperplane that separates the classes. SVMs typically benefit from a normalization step.

For both classifiers, the best-working configuration of hyperparameters has been identified through a grid search, as explained in the following section.

C. Hyperparameters tuning

There are two main sets of hyperparameters to be tuned:

- n_γ and n_ρ for the preprocessing
- random forest and SVM parameters

By assuming that the two are orthogonal, we can first select n_γ and n_ρ and then move on to the classifiers parameters.

Another simplifying assumption we can make is that $n_\rho = n_\gamma$. This helps us significantly lower the number of combinations we need to consider, as we now need to try $n = n_\rho = n_\gamma$

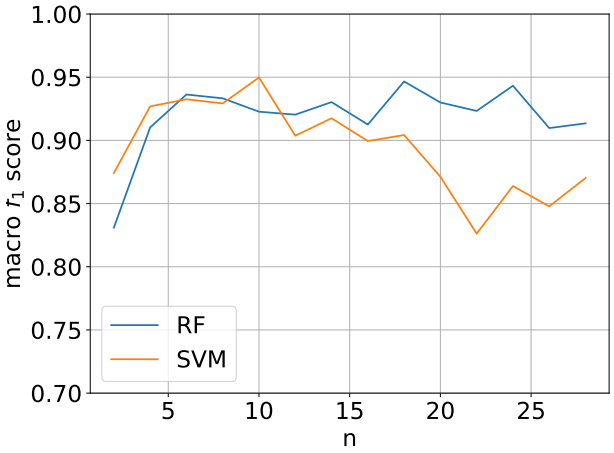


Fig. 6: Performance of default SVM and random forest as n varies

Model	Parameter	Values
Preprocessing	$n = n_\gamma = n_\rho$	$2 \rightarrow 46$, step 4
Random forest	*max_depth*	{None, 2, 5, 10, 50}
	n_estimators	100
	criterion	{gini, entropy}
SVM	*C*	{1, 5, 10, 50, 100, 500, 1,000}
	kernel	{rbf, linear, sigmoid}

TABLE I: Hyperparameters considered

configurations, instead of $n_\rho \cdot n_\gamma$. We can use an 80/20 train/test split on the development set and run a grid search on n. To this end, we will train a random forest and an SVM with their default configurations and assess their performance based on the resulting macro f_1 score.

After choosing a value for n, we can run a grid search[1] on both SVM and random forest, based on the hyperparameters defined in Table I.

III. Results

The tuning of the parameter n is summarized in Figure 6. We can immediately see that the random forest is more stable than SVM as n increases. Both classifiers achieve satisfactory performance for $n = 10$, which we can select as a reasonable value (larger values would affect the SVM with no meaningful improvement for the random forest). With this value of n, we ran the grid search for both random forest and SVM.

The best configuration for random forest was found for {*criterion=entropy, max_depth=None*[2]} (f_1 score $\approx$ 0.96), whereas the best configuration for the SVM was found for {*C=5, kernel=rbf*} (f_1 score $\approx$ 0.94). The two classifiers achieve satisfactory and comparable results.

We trained the best performing random forest classifier and SVM on all available development data. Then the models have been used to label the evaluation set. The public score obtained is of 0.972 for the random forest and of 0.988 for the SVM. Since these are the first scores obtained using the evaluation data, there should be no overfitting and we can reasonably assume that similar results can be obtained on the private score.

For comparison, a naive solution has also been defined. This solution carries out the following steps:

1) Split the signal into 20 chunks on the time dimension
2) For each chunk, extract mean and standard deviation of the samples
3) Train a default random forest on the obtained features

This solution obtains a public score of 0.658.

IV. Discussion

The proposed approach obtains results that far outperform the naive baseline defined. It does so by leveraging both time- and frequency-based features. We have empirically shown that the selected classifiers perform similarly for this specific task, achieving satisfactory results in terms of macro f_1 score.

The following are some aspects that might be worth considering to further improve the obtained results:

- Other feature extraction approaches may be considered. For example, the Mel Frequency Cepstral Coefficients (MFCC) are often used for speech recognition tasks [6]. Other approaches may leverage an automated feature extraction from the spectrogram of the signal (e.g. Convolutional Neural Networks). Both approaches are promising and could result in further improvements
- Run a more thorough grid search process. Only a limited set of hyperparameters has currently been studied. Exploring further options could result in an even better configuration. This includes considering additional classification models

The results obtained, however, are already very promising and there is little room for improvement. This classification problem is indeed quite easy and the datasets available are very limited.

[1] With another 80/20 train/test split

[2] This implies setting no limitation on the maximum depth of the trees

References

[1] (2019) Atis telecom glossary. ATIS. [Online]. Available: https://glossary.atis.org/glossary/voice-frequency-vf

[2] E. B. Goldstein, *Sensation and perception (3rd ed.)*. Wadsworth/Thomson Learning, 1989.

[3] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?" in *International workshop on machine learning and data mining in pattern recognition*. Springer, 2012, pp. 154–168.

[4] F. Saki, A. Sehgal, I. Panahi, and N. Kehtarnavaz, "Smartphone-based real-time classification of noise signals using subband features and random forest classifier," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 2204–2208.

[5] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, "Classification of audio signals using svm and rbfnn," *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 6069 – 6075, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417408004004

[6] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various mfcc implementations on the speaker verification task," in *Proceedings of the SPECOM*, vol. 1, no. 2005, 2005, pp. 191–194.