Final Project

Matthew Markowitz, Lifu Xiao

November 27, 2018

1 Introduction

The database is a set of noisy recordings, which have poor quality for further usage. So it make sense to improve them. In order to remove the noise, we propose to use online dictionary learning.

There were many challenges involved with this problem. One problem involved the large number of samples that were needed to begin the dictionary updates. Unfortunately, we can not begin training the dictionary until there are no zero elements in our diagonal matrix A. This means that until every single dimension must have at least one alpha with a non-zero element in it. To overcome this, we refused to update the dictionary until all the zero elements were filled. However, this is also not ideal because this requires substantially more training data as the window size for our audio increased. Experimentally, this can be overcome by adding a random small constant to our A before we start (such as 0.000000001), however, this is little to no theoretical backing for this, so we did not take this route.

Deep recurrent neural networks[1] have good performance on extracting acoustic features from noisy data. But it have a high computation cost. Another widely used method is spectral subtraction.

2 Problem Statement

We used a python library known as librosa to import our audio[2]. The audio recordings found in our test set had a sampling rate of 22050. This meant that every second of audio held approximately 22,000 numbers to represent it. For this reason, down sampling became necessary. Although some sacrifice in audio quality was necessary, we were able to reduce the sampling rate to 5,000, which made our calculations more feasible. The 5,000 points per second was still computationally intensive, but we found that we could break each second into X millisecond windows to ease computation further without sacrificing much quality. We found that a window size of 50 points or 50/5,000 = 1/100 second windows worked well for our dataset.

3 Algorithm

3.1 Data Preparation

Initializing the $\boldsymbol{A}_0 \in \mathbb{R}^{k \times k}$ and $\boldsymbol{B}_0 \in \mathbb{R}^{m \times k}$ as $\vec{\mathbf{0}}$
k is atoms number and m is the dictionary size.

3.2 Sparse Coding

When each x_t come, using LARS[3] to calculate

$$\alpha_t \triangleq \arg\min_{\alpha \in R^k} \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{D}_{t-1}\alpha\|_2^2 + k \cdot \lambda\|\alpha\|_1$$

where $\boldsymbol{x} \in \mathbb{R}^m, \boldsymbol{D} \in \mathbb{R}^{m \times k}$ and $t \leq T$(maximum number of iterations)
Then updating $\boldsymbol{A}, \boldsymbol{B}$ by

$$\boldsymbol{A}_t \leftarrow \boldsymbol{A}_{t-1} + \alpha_t \alpha_t^T$$
$$\boldsymbol{B}_t \leftarrow \boldsymbol{B}_{t-1} + \boldsymbol{x}_t \alpha_t^T$$

3.3 Dictionary Update

$$\boldsymbol{D}_t \triangleq \arg\min_{\boldsymbol{D} \in C} \frac{1}{t} \sum \frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{D}\alpha_i\|_2^2 + \lambda\|\alpha\|_1$$

Where $C \triangleq \boldsymbol{D} \in \mathbb{R}^{m \times k}$ s.t. $\forall j = 1, ..., k, \boldsymbol{d}_j^T \boldsymbol{d}_j \leq 1$ to ensure the convex.
Using block-coordinate descent to update dictionary Extracting columns of $\boldsymbol{A}$ and $\boldsymbol{B}$

$$\boldsymbol{A} = [\boldsymbol{a}_1, ..., \boldsymbol{a}_k] \in \mathbb{R}^{k \times k}$$
$$\boldsymbol{B} = [\boldsymbol{b}_1, ..., \boldsymbol{b}_k] \in \mathbb{R}^{k \times k}$$

for each column from $j = 1 \Rightarrow k$

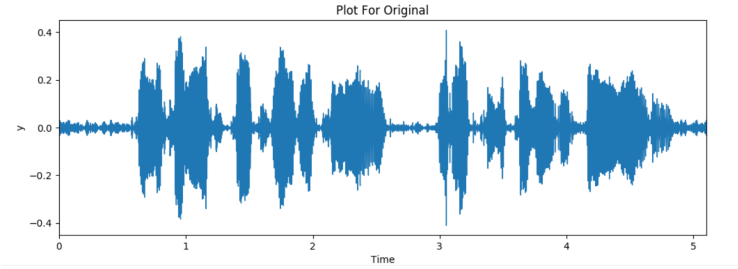
$$\boldsymbol{u_j} \leftarrow \frac{1}{A[j,j]}(\boldsymbol{b}_j - \boldsymbol{D}\boldsymbol{a}_j) + \boldsymbol{d}_j$$

$$\boldsymbol{d}_j \leftarrow \frac{1}{\max(\|\boldsymbol{u}_j\|_2, 1)}\boldsymbol{u}_j$$

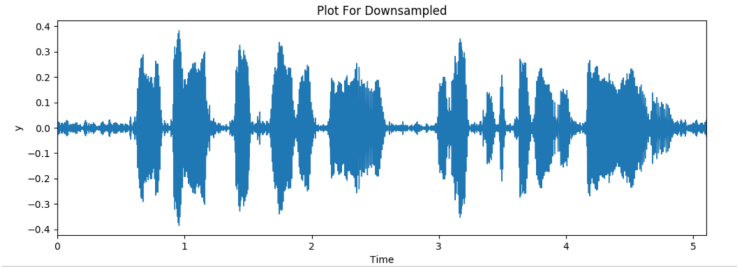
return $\boldsymbol{D}$ for next iteration

4 Experiments

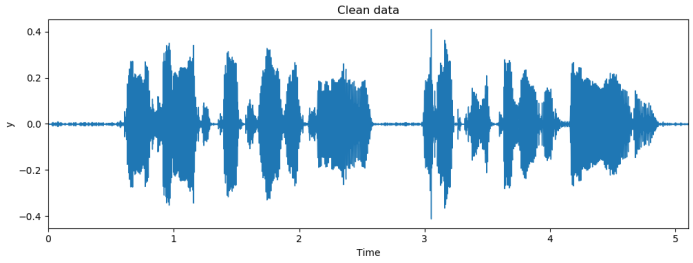
To visualize our result, we picked a 6 seconds segment of the output audio. The Original, Downsampled figures and the Clean data which is used for a baseline are as follows.



(a) Original



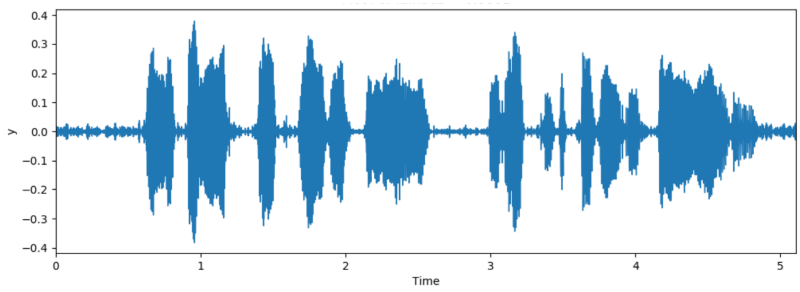
(b) Downsampled



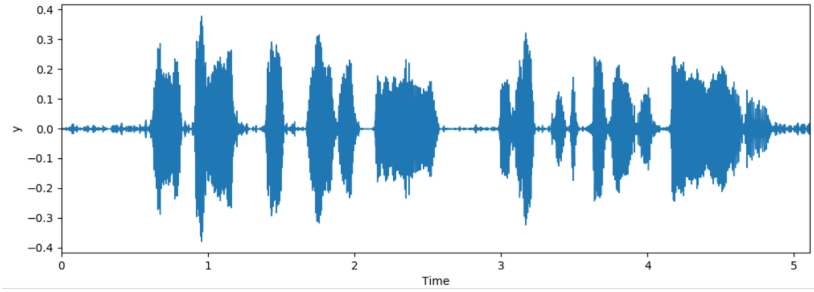
(c) CleanData

Figure 1: Original , Downsampled and the Clean data

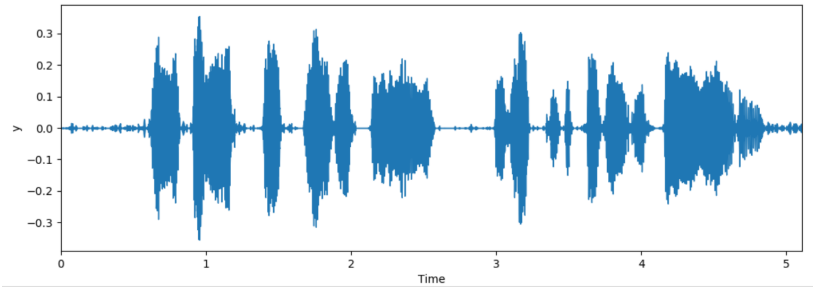
We choose $k = 500$ and $m = 50$ for the online dictionary learning step and the result generated by different λ are presented in Figure 2.



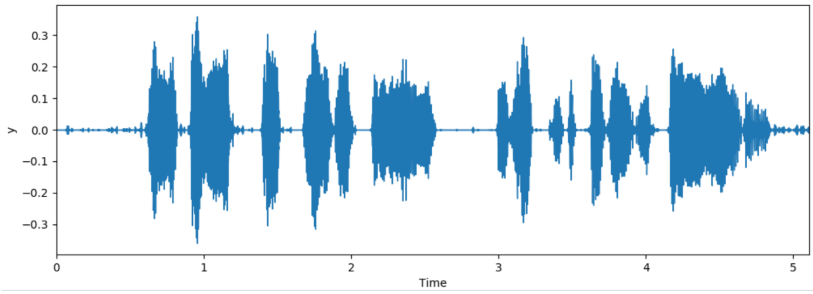
(a) $\lambda = 0.005$



(b) $\lambda = 0.025$



(c) $\lambda = 0.035$



(d) $\lambda = 0.05$

Figure 2: Output

References

[1] Cassia Valentini-Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi. Speech enhancement for a noise-robust text-to-speech synthesis system using deep recurrent neural networks. In *Interspeech*, pages 352–356, 2016.

[2] Cassia Valentini-Botinhao et al. Noisy speech database for training speech enhancement algorithms and tts models. 2017.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

[5] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.