Indian Institute of Technology Ropar, Rupnagar, Punjab-140001

Foundations of Data Science

MA515

Noise Reduction in Speech Signals using Singular Value Decomposition (SVD)

By:

Saurav Ray

(2022CEB1029)

Rajat Gupta

(2022CEB1026)

Rishi Gautam

(2022CEB1028)

Madhav Bareja

(2022EEB1188)

Instructor:

Prof. Arun Kumar

Table of Contents

Table of Snippets

List of figures

1. Introduction

The background noise is usually more than the speech recording and overpowers the signal being spoken and thus deteriorates the clarity of the speech. Moderate noise levels may increase the difficulty of the speech, and decrease its intelligibility. Indeed, it has been known that noise significantly affects the functionality of speech recognition systems: the speech recognition performance in noisy conditions is significantly lower in noise than in quiet conditions. Such problems prompt the need to have proper noise elimination.

The elimination of noise is essential in improving speech intelligibility and performance of the system. Denoising enhances the effective signal- noisy ratio (SNR) of the speech signal, and thus speech content becomes more prominent over the background. Better SNR equates to a higher perceived voice quality and a more trustworthy automated processing. Practically, clean speech is required in the applications like hearing appliances, voice communication and automatic speech recognition, all of which perform poorly under high-noise condition. Denoising algorithms are capable of enhancing the listener comprehension and recognition by suppressing noise and, therefore, can considerably improve the recognition rate.

In this project we consider a denoising method with a mathematical background, which is based on Singular Value Decomposition (SVD). SVD is a matrix factorization method which breaks down a data matrix into orthogonal elements. In the case of speech data, SVD is capable of successfully subtracting the speech subspace made up of the structured subspace and the speech noise subspace made up of the unstructured subspace. In a more practical manner, the speech spectrogram (organized in the form of a matrix) will be split in such a way that the largest singular values and their vectors will represent the most descriptive speech phenomena, and the smaller ones of the singular values will represent noise. Through retention, we extract the main elements and discard the others to achieve a low-rank approximation of the data to preserve the speech and reject noise. This subspace filtering method has been demonstrated to have the effect of improving the quality of the voice by isolating noise and speech signal.

2. Dataset

In this project, we have utilized the Google Speech Commands Dataset, which is a publicly available popular benchmark corpus of speech recognition research and signal processing research. The dataset was created by Google TensorFlow team and it consists of a big set of brief audio clips of spoken words that can be used mainly to train and evaluate small-footprint speech recognition models.

2.1. Dataset Overview

The sample includes some 65,000 one-second audio samples, one spoken word in each. Thousands of speakers across the globe recorded these utterances and thus, natural differences in accent, speaking style, loudness and recording conditions were achieved. This diversity renders the dataset to be very useful in speech enhancement, denoising, and robustness analysis experiments. It includes 30 primary command words, such as:

- yes, no, up, down, left, right, on, off, stop, go
- digits like zero, one, two
- and other simple interaction commands.

These commands serve as a standardized set for evaluating speech recognition systems and noise reduction pipelines.

2.2. Audio Specifications

Each file in the dataset follows consistent technical specifications:

- **Audio Format:** 16-bit PCM(Pulse-Code Modulation) wav
- **Sampling Rate:** 16,000 Hz (16 kHz)
- **Channels:** Mono
- **Bit Depth:** 16-bit linear PCM
- **Target Duration:** ~1.0 second
- **Typical Sample Count:** ~16,000 samples

Because the sampling rate and encoding format are uniform throughout the dataset, it becomes straightforward to apply signal processing methods—such as STFT, spectrogram generation, and SVD—to all recordings without additional resampling or normalization.

3. Methodology and Terminologies

3.1. Preparing the Speech and Noise Signals

A clean speech waveform $x[n]$and a noise signal $n[n]$are first loaded and trimmed to the same length. Both signals are sampled at 16 kHz, giving:

$$x= [x_1, x_2, \ldots, x_N] \text{ and } n= [n_1, n_2, \ldots, n_N], N=16000.$$

The noisy speech signal is created using the additive model:

$$y[n] = x[n] + \alpha\, n[n]$$

where α controls the noise strength.

```
clean, _ = librosa.load(clean_path, sr=sr)
noise, _ = librosa.load(noise_path, sr=sr)

noise = noise[:len(clean)]
noise_factor = 0.01
noisy = clean + noise_factor * noise
```

Snippet 1:Preparing the speech and noise signals

3.2. Short-Time Fourier Transform (STFT)

Because speech is non-stationary, its frequency content changes over time. The STFT analyzes local time segments using a sliding window.

For a window $w[n]$of length L, each frame's STFT is:

$$Y\ (f, t) = \sum_{\{n=0\}}^{\{L-1\}} y[n + tH]w[n]e^{-\frac{j2\pi fn}{L}}$$

where

- t is the frame index,
- H is the hop size,
- f is the frequency bin.

Stacking all frames gives the complex STFT matrix Z:

$$Z = \begin{bmatrix} Y(1,1) & Y(1,2) & \cdots & Y(1,T) \\ Y(2,1) & Y(2,2) & \cdots & Y(2,T) \\ \vdots & \vdots & \ddots & \vdots \\ Y(F,1) & Y(F,2) & \cdots & Y(F,T) \end{bmatrix}$$

The magnitude spectrogram is:

$$S = | Z |$$

```
f, t, Zxx = stft(noisy, fs=sr, nperseg=512)
spectrogram = np.abs(Zxx)
```

Snippet 2:Applying STFT

3.3. Singular Value Decomposition (SVD)

The spectrogram $S$is factorized using SVD:

$$S = U \Sigma V^T$$

Where:

- $U \in \mathbb{R}^{F \times F}$contains frequency basis vectors
- Σ contains singular values $\sigma_1 \geq \sigma_2 \geq \cdots$
- $V \in \mathbb{R}^{T \times T}$contains time basis vectors.

The singular values show how much "energy" each rank-1 component contributes. Speech energy is concentrated in the first few singular values; noise is spread out.

```
U, S, Vt = np.linalg.svd(spectrogram, full_matrices=False)
```

Snippet 3: Using SVD

3.4. Low-Rank Approximation using leading singular values

To reduce noise, only the largest $k$singular values are kept:

$$\Sigma_k = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{bmatrix}$$

The denoised spectrogram is:

$$S_k = U_k \Sigma_k V_k^T$$

This removes components associated with smaller singular values, which are mostly noise.

```
k = 1  # number of singular values to keep

S_reduced = np.zeros_like(S)
S_reduced[:k] = S[:k]

spectrogram_denoised = (U * S_reduced) @ Vt
```

Snippet 4: Removing components with smaller singular values

3.5. Reconstruction using Inverse STFT

To get back the time-domain signal, the denoised magnitude spectrogram is combined with the original phase:

$$\tilde{Z}(f,t) = S_k(f,t) \cdot e^{j \cdot \angle Z(f,t)}$$

Applying ISTFT produces:

$$\hat{y}[n] = \text{ISTFT}(\tilde{Z})$$

```
Zxx_denoised = spectrogram_denoised * np.exp(1j * np.angle(Zxx))
_, denoised = istft(Zxx_denoised, fs=sr)

# Match lengths
if len(denoised) > len(clean):
    denoised = denoised[:len(clean)]
else:
    denoised = np.pad(denoised, (0, len(clean) - len(denoised)))
```

Snippet 5:Applying ISTFT

3.6. SNR Calculation

The Signal-to-Noise Ratio is computed using:

$$\text{SNR} = 10\log_{10}\left(\frac{\sum x[n]^2}{\sum (x[n] - \hat{y}[n])^2}\right)$$

```
def snr(clean, signal):
    power_clean = np.sum(clean**2)
    power_noise = np.sum((clean - signal)**2)
    return 10 * np.log10(power_clean / power_noise)

snr_noisy = snr(clean, noisy)
snr_denoised = snr(clean, denoised)
print("SNR (Noisy):    ", snr_noisy)
print("SNR (Denoised): ", snr_denoised)
```

Snippet 6:SNR calculation

4. Results

4.1. Time-Domain Waveform Comparison

Figure 1 shows the clean, noisy, and SVD-denoised waveforms plotted over a 1-second duration.

4.1.1 Clean Speech

The clean waveform exhibits the natural amplitude envelope of the spoken word.

- The speech energy is concentrated in the initial 0.3 seconds because the word was spoken for this long.
- After 0.3 seconds, the signal gradually decays with decreasing oscillations.
- The waveform is smooth with periodic structure, typical of voiced speech.

4.1.2 Noisy Speech

After adding white noise, the waveform becomes visibly distorted:

- High-frequency fluctuations are seen across the entire duration.
- The noise is especially noticeable in the low-energy tail of the signal (after 0.35 s).
- Noise masks the fine periodicity of the speech, reducing visual clarity and signal contrast.

4.1.3 Denoised Speech (SVD)

The denoised waveform retains the overall shape of the clean signal:

- High-frequency noise fluctuations are reduced.
- The voiced region (0–0.3 s) becomes smoother.
- The tail of the signal still contains residual noise, but its intensity is lower than in the noisy signal.

Observation:

The denoising effect is not much clearly visible here but it can be seen clearly in spectrogram and SNR values. The SVD reconstruction preserves the main amplitude envelope of the speech while removing much of the random noise added earlier.

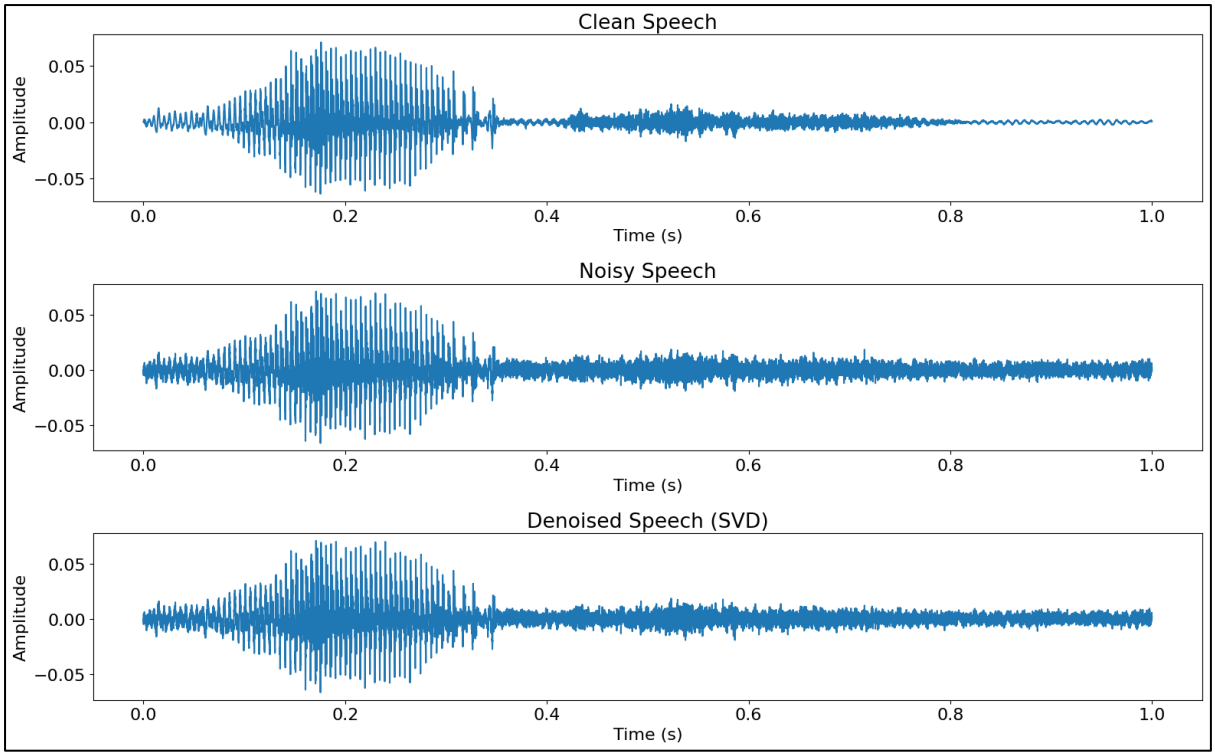

Figure 1:Waveform Comparison n=0.01, k=20

4.2. Spectrogram Analysis

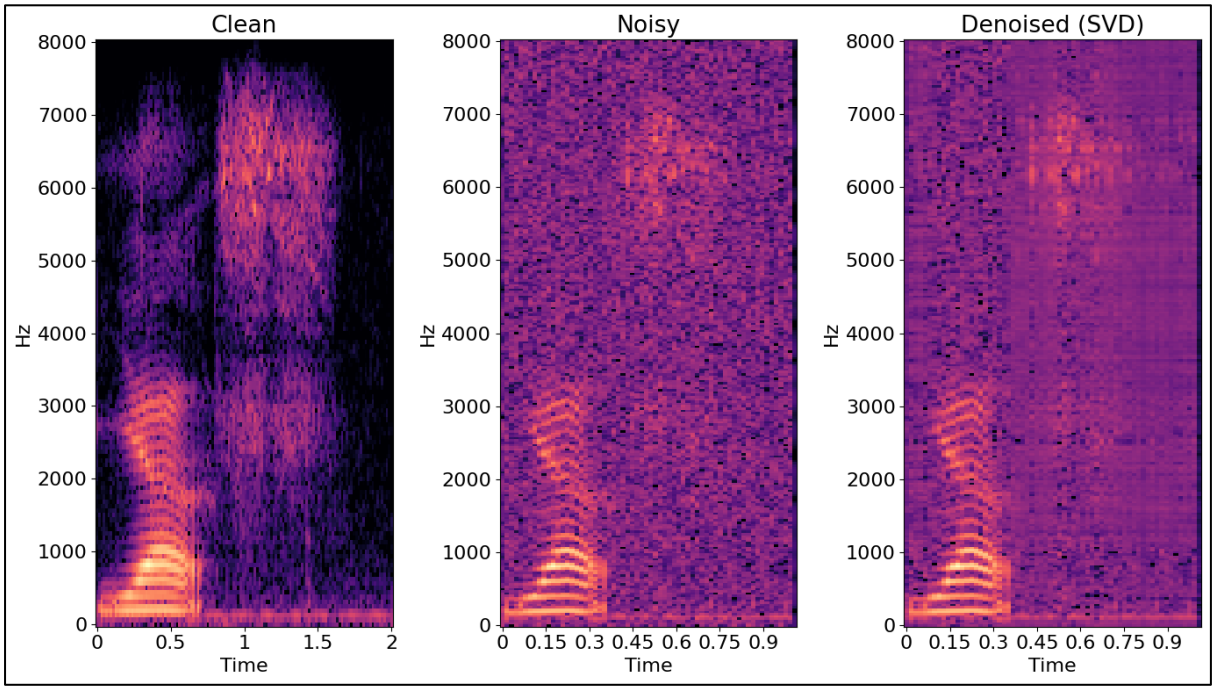

Figure 2:Spectrogram Comparison n=0.01 and k=20

Figure 2 shows the magnitude spectrograms of the clean, noisy, and SVD-filtered signals.

4.2.1 Clean Speech Spectrogram

- Clear harmonics are visible in the frequency range below 4000 Hz.
- The formant structure of the word is distinctly observable.
- The energy is concentrated in smooth frequency bands, typical of clean speech.

4.2.2 Noisy Speech Spectrogram

- Noise introduces nearly uniform speckling across all frequencies.
- Harmonic patterns are partially masked; the background becomes much brighter.
- High-frequency regions (4000–8000 Hz) contain dense random noise, overpowering speech components.

4.2.3 Denoised Speech (SVD) Spectrogram

- Noise is visibly reduced across the full spectrum.
- Harmonics and formant structures reappear more clearly compared to the noisy spectrogram.
- Lower frequencies recover their smooth structure.
- High-frequency noise (above 5000 Hz) is significantly suppressed, though some residual noise remains.
- The spectrogram looks smoother overall due to the low-rank approximation.

Interpretation:

The SVD-filtered spectrogram demonstrates that the largest singular component captures the essential temporal and spectral characteristics of the speech signal, while smaller singular values (dominated by noise) are effectively suppressed

4.3. Quantitative Evaluation (SNR Improvement)

Using the SNR formula:

$$\text{SNR} = 10\log_{10}\left(\frac{\sum x[n]^2}{\sum (x[n] - \hat{y}[n])^2}\right)$$

the following values were obtained:

- SNR (Noisy): 9.419912
- SNR (Denoised): 10.0189

Since:

$$\text{SNR}_{\text{denoised}} > \text{SNR}_{\text{noisy}}$$

the SVD-based filtering provides a measurable improvement in noise suppression. This confirms that low-rank approximation effectively reduces noise while preserving important speech features.

4.4. Comparison of SNR Gain and rank k

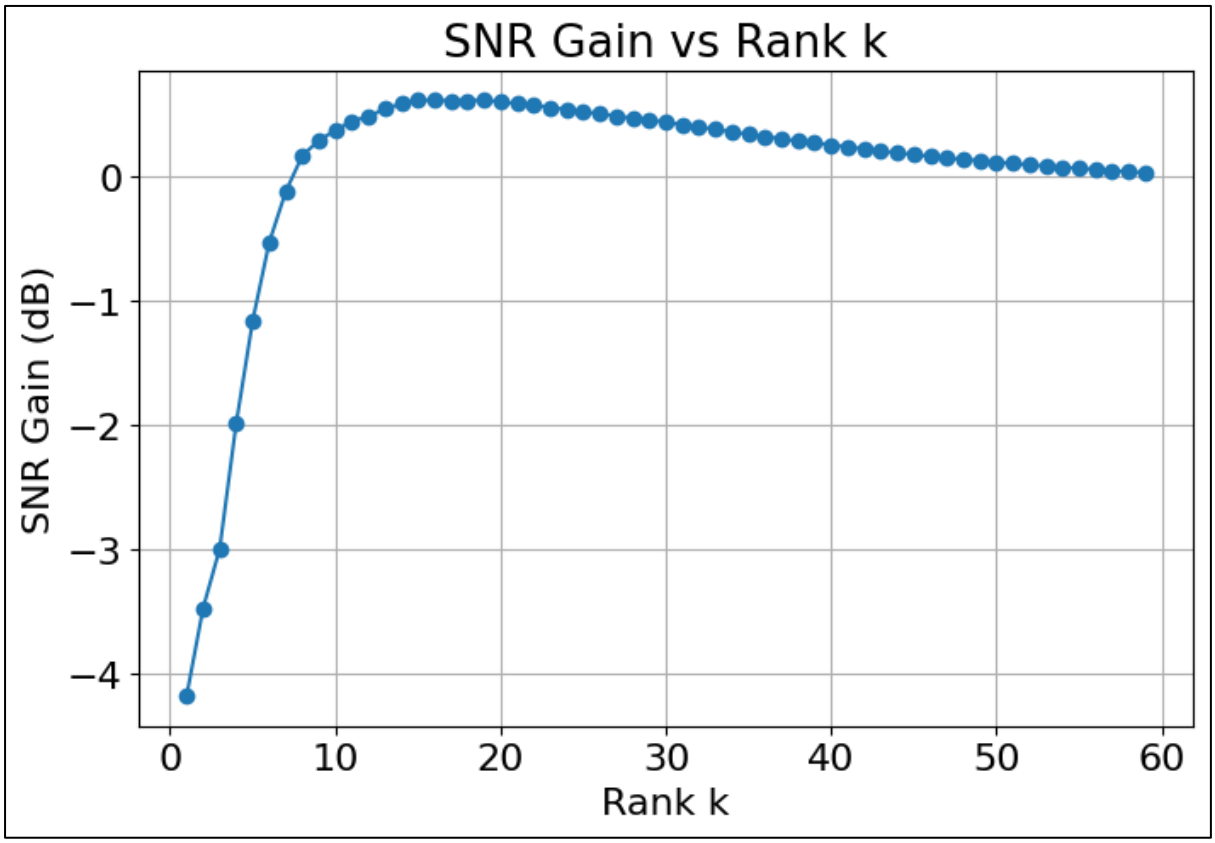

Figure 3: SNR Gain vs k

Interpretation:

- For very small values of k, the SNR gain becomes negative because only a small portion of the speech content is retained, resulting in noticeable distortion.
- As k increases, more meaningful speech information is preserved while noise components are still removed, leading to an improvement in SNR.
- For larger k, the additional components mainly contain noise, so the SNR gain decreases and gradually approaches zero as the reconstruction becomes similar to the original noisy signal.

5. Conclusions

In this project, we explored how Singular Value Decomposition (SVD) can be used to reduce noise in speech recordings by taking advantage of an important property of speech: most of its meaningful information is concentrated in only a few dominant patterns. By converting the audio into a spectrogram using STFT, applying SVD, and reconstructing a low-rank version of the spectrogram, we were able to separate the structured speech components from the more random, unstructured noise.

The results clearly show that SVD does help clean the signal. Although the difference in waveforms is subtle at first glance because speech naturally masks noise during voiced segments the spectrograms tell a much clearer story. The noisy spectrogram is filled with scattered energy across all frequencies, while the denoised spectrogram shows smoother, more defined speech patterns, especially in the lower and mid frequency regions. This visual improvement is also reflected numerically: the SNR increases after denoising, confirming that SVD succeeded in reducing noise.

A deeper analysis using a rank sweep revealed an important finding: the choice of rank $k$plays a critical role. Too small a rank removes not only noise but also parts of the speech itself, leading to a drop in SNR. As the rank increases, speech components begin to reappear and the denoised signal improves, eventually reaching an optimum region where SNR gain is the highest. Beyond this point, however, increasing $k$reintroduces noise because the smaller singular values mostly represent noisy components. This behavior matches the theoretical understanding that speech is low-rank while noise is high-rank.

Limitations

While SVD performed well overall, this approach also comes with several limitations:

1. **Loss of fine speech details at low ranks:**
   Very aggressive low-rank filtering removes important high-frequency details such as fricatives (“s”, “sh”), making speech sound slightly smoother or muffled.
2. **Sensitivity to rank selection:**
   Choosing the right value of $k$is not trivial. A fixed $k$does not work equally well for all speakers, words, or noise levels. In practical applications, this would require adaptive or automated rank selection.

3. **Phase is not denoised:**

   The method only filters the magnitude of the spectrogram. The noisy phase is reused during ISTFT reconstruction, which limits how "clean" the final output can be.

Final Thoughts

Despite these limitations, this project demonstrates that SVD is a simple, elegant, and surprisingly powerful method for speech denoising. It provides a clear improvement in SNR, visibly cleans the spectrogram, and offers an intuitive way to understand how speech and noise behave in the time–frequency domain. While modern deep learning methods can outperform SVD, this classical linear-algebra approach remains an excellent starting point for understanding the fundamentals of noise reduction and signal representation.