# Small Project
# Random projection in dimensionality reduction: Applications to image and text data

Group 1
Alice De Schutter, Giovanni Castellano, Jakub Reha and Yuru Liu

October 3, 2022

**Abstract**

In this project, the results of the paper "Random projection in dimensionality reduction: Applications to image and text data" [1] are replicated. In particular, different dimensionality reduction techniques are compared based on the amount of distortion caused by the method and its computational complexity. For completeness, the methods were tested on different types of data (images, text, audio).

# 1 Introduction

Transforming data from a high- to a low-dimensional space, while retaining some important properties of the original data, is known as dimensionality reduction. Mathematically, linear dimensionality reduction can be expressed as:

$$X_{k\times N} = R_{k\times d} X_{d\times N},$$

where $X_{d\times N}$ is the original d-dimensional data matrix, $X_{k\times N}$ is the reduced k-dimensional data matrix and $R_{k\times d}$ is the mapping between the two spaces.
Principal Component Analysis (PCA) is one of the most used dimensionality reduction techniques. It projects data onto a lower-dimensional subspace while preserving as much of the data's variance as possible. Nevertheless, PCA has computationally intensive steps.
Random projection (RP) is an alternative and computationally efficient method. It is based on the Johnson-Lindenstrauss lemma [2], which states that there exists a mapping from a higher- to a suitable lower-dimensional space such that euclidean distances between the data points are nearly preserved.
Due to the lack of experimental results on RP, the study "Random projection in dimensionality reduction: Applications to image and text data" [1] compared RP to other dimensionality reduction methods. Comparisons were based on the amount of distortion caused by each method and on their computational complexity. From the paper, a method causes distortion when similarity between pairs of data points is not preserved. In particular, the authors decided to use the Euclidean distance and the dot product as measures of similarity. The study showed RP to be the best method of dimensionality reduction when projecting the data onto a low dimensional space.

In the following, we replicate the results of the paper implementing the same dimensionality reduction algorithms (RP, SRP, PCA, and DCT), and comparing the distortion of the produced embeddings on similar datasets.

# 2 Datasets

A collection of datasets was built by searching the internet. The text [1] and image [2] datasets used in the original paper were found and included in our study [1]. Moreover, additional datasets from each category (text and image [3] [4]), as well as one audio dataset [5], were also incorporated in the study.

## 2.1 Text

A detailed text pre-processing description was omitted from the original paper. Therefore, in this study, the pre-processing steps could not be replicated exactly. However, in an attempt to arrive at similar pre-processing results, as mentioned in the original paper, stop words were removed from the text datasets and no stemming was used. In addition, the header, numbers, names, email addresses, punctuation and words with length $> 12$ or length $< 2$ were also eradicated. Thereafter, the remaining text was transformed to lowercase form. Finally, in order to arrive at the same vocabulary size as in the original paper, 5000 of the most frequent words were selected (d = 5 000). From this, the normalized frequency matrix was created.

[1] http://www.cs.cmu.edu/ textlearning/
[2] https://web.archive.org/web/20150412005848/https://research.ics.aalto.fi/ica/data/images/
[3] https://www.kaggle.com/dipankarsrirag/topic-modelling-on-emails
[4] https://www.kaggle.com/msambare/fer2013
[5] https://www.kaggle.com/uwrfkaggler/ravdess-emotional-speech-audio

### 2.2 Images

As done in the original paper, for each image dataset, 1000 windows of sizes $50 \times 50$ $px$ were randomly drawn from the initial images. These were stored as noiseless image data, where each image window is represented as a d-dimensional column vector (d = 2 500). Furthermore, the images were corrupted with salt-and-pepper impulse noise (with a probability of 0.2 that a pixel in the image was turned white or black) and stored as noisy image data.

### 2.3 Audio

The recordings were each trimmed to the same length, corresponding to 100 000 samples (d = 100 000). No further pre-processing was performed on the audio dataset.

## 3 Methods for dimensionality reduction

### 3.1 PCA - Principal Component Analysis

To implement PCA, we computed the eigendecomposition of the data covariance matrix $E\{XX^T\}$:

$$E\{XX^T\} = E\Lambda E^T$$

where the columns of $E$ are the eigenvectors of the data covariance matrix and $\Lambda$ is the diagonal matrix which contains the respective eigenvalues.
Subsequently, we projected the data onto a subspace spanned by the most important eigenvectors:

$$X^{PCA} = {E_k}^T X,$$

where $E_k$ is a $d \times k$ matrix containing the k largest eigenvectors of $E$.
However, such implementation requires computationally intensive steps. These are:

- Finding the covariance matrix: $\mathcal{O}(Nd^2)$
- Finding the eigenvalue decomposition of the covariance matrix: $\mathcal{O}(d^3)$

The computational complexity of such implementation is $\mathcal{O}(Nd^2)+\mathcal{O}(d^3)$. Due to this high complexity and the size of the datasets the authors of the original paper used an Expectation-Maximization version of PCA. For the same reason, we decided to provide a second implementation which is based on the sklearn's Truncated SVD to compute PCA. This gave us a way to compute the running time for different values of $k$.

### 3.2 RP - Random Projections

Two different distributions were used for the RP mapping. The methods are differentiated and referred to as RP (using the Gaussian distribution) and SRP (using the simplified distribution according to Achlioptas [3]). The random mapping $R_{k \times d}$ is constructed by randomly drawing elements from these distributions and subsequently normalizing the rows of the mapping matrices $R_{k \times d}$. The computational complexity of random projections is $\mathcal{O}(dkN)$.

### 3.3 DCT - Discrete cosine transform

DCT is a Fourier-like transform widely used in images and audio compression. Many types of DCT exist. In the original paper, the authors did not mention which one they were using. For that reason, we decided to proceed with DCT-II, which seems to be the most common in the literature for this kind of application. Considering the relation between DCT and DFT, the DCT-II of a vector

$V_1 = [v_1, v_2, v_3, ..v_N]$ with length $N$ is computed in the following way:

1) Compute the DFT of the vector $V_2 = [v_1, v_2, v_3, ..v_N, v_N, v_{N-1}, v_{N-2}, ..., v_1]$ with size $2N$ (mirrored vector).
2) Multiply each component of $V_2$ by $e^{\frac{-j\pi k}{2N}}$ where $k$ is the index of the component.
3) Take the real part of the first $N$ components of the resulting vector.

In this way, we get a real vector with size $N$, which is the DCT [4].
The most efficient way to compute the DFT is by using the Fast Fourier Tranform (FFT). We provided the implementation of FFT as it was presented in the original paper [5]. Such implementation however, works only when the input vector has a number of components that is a power of 2. In order to reproduce the results of the paper, we decided to use the numpy implementation of the FFT, which is based on the FFTPACK (a package of Fortran subprograms). The use of the FFT reduces the computational complexity of DFT from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ [5].

# 4 Results

In this section, the methods mentioned above are compared by projecting the data on subspaces with different number of dimensions $k$. The mean error in the similarity between the lower dimensional and the original dimensional data points is presented together with the 95% intervals (for 100 pairs as in the original paper).

## 4.1 Results on Image Data

The measure of similarity for images was the Euclidean distance. Therefore, the term $\sqrt{\frac{d}{k}}$ is used to scale the distances in the lower dimensional space based on the JL lemma [2]. Figure 1 illustrates the average error of the methods and the confidence interval on the (small) dataset [6] used in the original paper. We note that the trends in this plot are very similar to those in the paper, showing that RP and SRP are sufficiently accurate and even outperform PCA and DCT in lower dimensions. In figure 2, the same algorithms are tested on a bigger dataset [7]; also in this case, the same trends are observed.

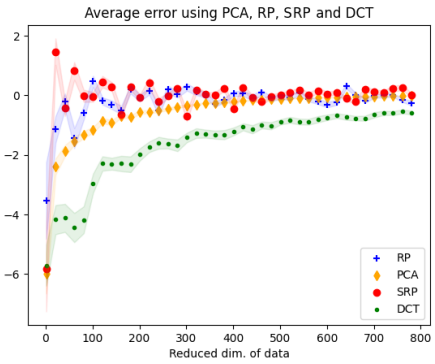

Figure 1: Image data (Small)

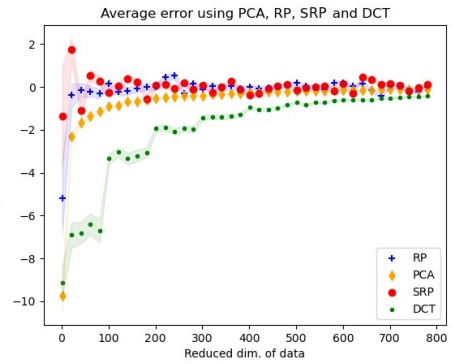

Figure 2: Image data (Big)

[6] https://web.archive.org/web/20150412005848/https://research.ics.aalto.fi/ica/data/images/
[7] https://www.kaggle.com/msambare/fer2013

In figures 3 and 4 the methods are tested on the same image datasets but the images are corrupted with noise. Again, the same trends are observed. However, this time SRP and RP perform better by an even bigger margin. The plots also show errors in the distances after applying the Median Filter to the noisy data as a baseline for comparison.

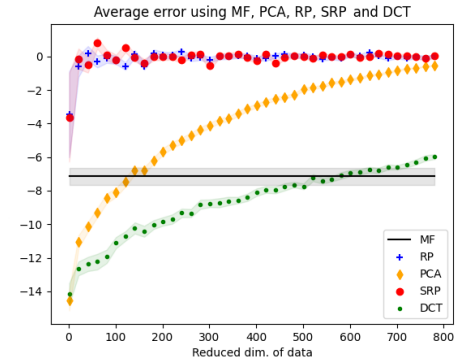


Figure 3: Noisy image data (Small)

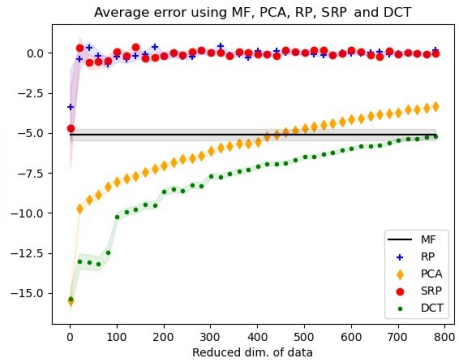


Figure 4: Noisy image data (Big)

## 4.2 Results on Text Data

For text data the measure of similarity was the Inner product. Figures 5 and 6 show the results for the two datasets [8] [9]. RP is not performing that well compared to SVD. However, this result agrees with the result of the same experiment in the original paper.

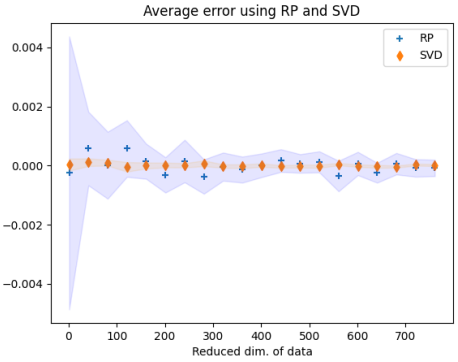


Figure 5: Text data (Small)


Figure 6: Text data (Big)

## 4.3 Results on Audio Data

The methods were also tested on an additional audio dataset, not present in the original paper. This dataset is also significantly different in the number of dimensions (d = 100 000). For this data, the Euclidean distance was used as measure of similarity. Also in this case, the same trend are observed: figures 7 and 8 show that RP and SRP outperform PCA and DCT in lower dimensions. Moreover, we notice that we need a much higher number of dimensions to get good results with DCT.

[8] http://www.cs.cmu.edu/ textlearning/
[9] https://www.kaggle.com/dipankarsrirag/topic-modelling-on-emails

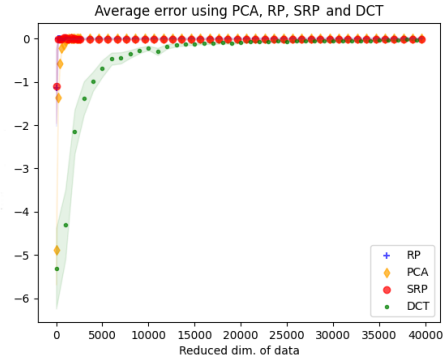


Figure 7: Audio data


Figure 8: Audio data zoomed

## 4.4 Runtime results

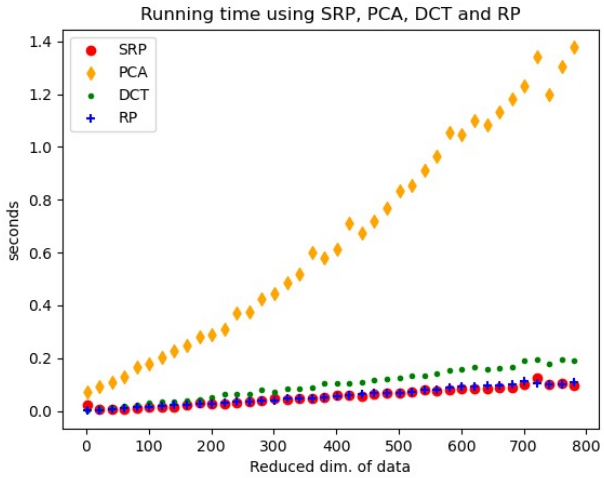


Figure 9: Runtime on image data (computed using time.process_time())

It is clear from figure 9 that PCA is much more expensive than the other methods. However, we cannot observe a significant difference between SRP and RP, this probably means that our implementation of SRP can be optimized. For instance, we did not use the Numpy methods optimized for sparse matrices, and the Scipy implementaion of SRP provides a more efficient way to sample the vectors from the Achlioptas [3] distribution. Moreover, we have to take in consideration that the authors used FLOPS to measure the performance, whereas we are using the running times.

# 5 Discussion

- **Discussion about the original paper:** The results were very hard to replicate since the authors did not provide a complete description of how the data was pre-processed. Moreover, it is not clear which version of the algorithms they used. We also observe that the number of flops is not a perfect measure of performance since it depends on the CPU.

  On the other hand, we believe it was a good idea to perform the algorithm on different types of datasets and to use different measures of similarity. Moreover, it is very interesting that they took in consideration the effect of noise on the results. We also find extremely important the presence of the confidence intervals in the plots, since without them the results would not be useful.

- **Recent developments:** The original paper does not bring anything new in terms of theory, it only demonstrates the effectiveness of using random projection in practice. This might have played a role in popularizing RP and other randomized algorithms such as the streaming algorithms. Random projections also found their place in many approximation algorithms such as [6], [7], [8]. Moreover, in a recent paper by Magen [9], the author shows how volumes and affine distances can also be preserved. Dmitriy Fradkin and David Madigan performed some experiments to compare the performance of different machine learning algorithms after projecting the dataset using PCA and RP. Interestingly, the results show that RP are best suited for use with Nearest Neighbor methods [10].

# 6 Conclusion

As seen in the figures, we obtained similar results to those presented in the paper. Moreover, the results hold even if we use a different type of dataset. Therefore, we can conclude that RP is the best dimensionality technique to use if we are interested in preserving the average similarity between points; in fact, we obtained very good results even if we projected the data onto low-dimensional spaces. Moreover, RP is much less expensive than PCA from a computational point of view. This does not mean however, that RP is always the best technique to use; we know for instance, that PCA always returns the best projection if we want to preserve the positions of the points (the mean square error) [11]. DCT on the other hand, is the best technique to use if we are interested in reconstructing the original data (for instance if we want to visualize the image or we want to listen to the audio). This makes DCT suitable for data compression [1].

# References


[1] Ella Bingham and Heikki Mannila. "Random Projection in Dimensionality Reduction: Applications to Image and Text Data". In: 2001. URL: https://doi.org/10.1145/502512.502546.

[2] W. B. Johnson and J. Lindenstrauss. *Extensions of Lipschitz mappings into a Hilbert space*. Contemporary mathematics 26(189-206). 1984.

[3] Dimitris Achlioptas. "Database-friendly Random Projections". In: *Proc. Principles of Database Systems (PODS)* (July 2001). DOI: 10.1145/375551.375608.

[4] endolith (stackexchange). *Fast Cosine Transform via FFT*. 2013. URL: https://dsp.stackexchange.com/questions/2807/fast-cosine-transform-via-fft.

[5] James W. Cooley and John W. Tukey. "An Algorithm for the Machine Calculation of Complex Fourier Series". In: (May 1965). DOI: 10.1090/S0025-5718-1965-0178586-1.

[6] Tamas Sarlos. "Improved Approximation Algorithms for Large Matrices via Random Projections". In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. 2006, pp. 143–152. DOI: 10.1109/FOCS.2006.37.

[7] Renchi Yang. *Fast Approximate CoSimRanks via Random Projections*. 2021. arXiv: 2010.11880 [cs.SI].

[8] Michael Kerber and Sharath Raghvendra. *Approximation and Streaming Algorithms for Projective Clustering via Random Projections*. 2015. arXiv: 1407.2063 [cs.CG].

[9] Avner Magen. "Dimensionality Reductions That Preserve Volumes and Distance to Affine Spaces, and Their Algorithmic Applications". In: *RANDOM*. 2002.

[10] Dmitriy Fradkin and David Madigan. "Experiments with Random Projections for Machine Learning". In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Apr. 2003). DOI: 10.1145/956750.956812.

[11] Michel Verleysen John A. Lee. *Nonlinear dimensionality reduction*. New York: Springer. 2007.

[12] R. Hecht-Nielsen. *Context vectors: general purpose approximate meaning representations self-organized from raw data*. Computational intelligence: Imitating life, 43–56. 1994.