Non-redundant Dimensionality Reduction Methods and Applications in Image Compression

Math 797 Spring 2020 Final Project Report
Student Name: Hoang Bao Linh Tran (Linh Tran)

Abstract

This report aims to give a summary and some critical comments of the paper titled "Non-redundant Spectral Dimensionality Reduction" by Yochai Blau and Tomer Michaeli [1]. We reproduced an experiment in the paper and produced several new ones to verify the redundant dimension phenomenon discussed in the paper. In the same experiments, we also tested our own redundancy removal algorithm, based on the same mathematical foundations as the authors' algorithm.

1 Summary of the Paper

Popular dimensionality reduction algorithms such as Isomap [7], Locally Linear Embedding (LLE) [10], Hessian LLE [6], Locally Tangent Space Alignment (LTSA) [9] and Laplacian Eigenmaps (LEM) [8] share a common characteristics: Their solutions are either the top or bottom eigenvectors of some kernel matrix built from the input data. Blau and Michaeli in their paper [1] observed a common problem of these algorithms, where on certain datasets they produce *redundant projections*, which undermines the quality of the output data.

1.1 The redundancy phenomenon

The output of dimensionality reduction algorithms consists of components (equivalently, projections), each corresponding to one new dimension in the reduced data. A dimension is redundant if it is heavily correlated to one or many dimensions before it. In many cases, the correlation is so high that the redundant dimension approximates a deterministic function of previous ones. In particular, for two-dimensional embeddings, redundancy causes the shape of the output data to look narrow, more resemblant of a one-dimensional strip than the supposedly two-dimensional shape of the input data.

The authors demonstrated this redundancy phenomenon in results from LLE, HLLE and LTSA on a Swiss Roll dataset and showed that their non-redundant versions avoid it. Although dimensionality reduction algorithms are guaranteed to produce orthogonal projections, meaning their correlation is zero, this does not guarantee non-redundancy. We came up with a minimalistic example to illustrate this point. Consider a circle dataset, generated by $\langle \cos(2\pi X), \sin(2\pi X) \rangle$ for $X \sim \mathrm{Uni}\,(0,1)$. Clearly the x and y components are functions of each other, indicating absolute redundancy. However, their correlation is approximately 0, since

$$\mathbf{Cov}\left[\cos(2\pi X), \sin(2\pi X)\right] = \int_0^1 \cos(2\pi x)\sin(2\pi x)dx - \int_0^1 \cos(2\pi x)dx \int_0^1 \sin(2\pi x)dx = 0$$

Preprint. Under review.

when $X \sim \text{Uni}(0,1)$. In the paper, the authors give a more detailed version of this argument. They considered the Narrow Swiss Roll manifold, whose length (as a rectangular strip in 2D) is at least 2.5 times its width. With sufficiently many samples, the first two components of LEM will approximate the top two eigenfunctions of the Laplace-Beltrami operator ∇^2, which are $\varphi_{1,0}(x_1,x_2) = \cos\left(\frac{2\pi x_1}{L_1}\right)$ and $\varphi_{2,0}(x_1,x_2) = \cos\left(\frac{4\pi x_1}{L_1}\right)$. The relation $\varphi_{2,0} = 2\varphi_{1,0}^2 - 1$ indicates redundancy, while the equation

$$\int_0^1 \cos\left(\frac{4\pi x_1}{L_1}\right)\cos\left(\frac{2\pi x_1}{L_1}\right)dx_1 = \frac{1}{2}\int_0^1 \left(\cos\left(\frac{6\pi x_1}{L_1}\right) + \cos\left(\frac{2\pi x_1}{L_1}\right)\right)dx_1 = 0$$

shows that orthogonality is still present. Thus *orthogonality does not imply non-redundancy*.

Note that the degree of redundancy, i.e. the number of redundant components may be arbitrarily high, as the first k eigenfunctions of ∇^2 are $\cos\left(\frac{2\pi x_1}{L_1}\right), \cos\left(\frac{4\pi x_1}{L_1}\right), \ldots, \cos\left(\frac{2k\pi x_1}{L_1}\right)$ for $k = \lfloor L_2/L_1 \rfloor$. This invalidates the naive approach of computing a large constant number of components and picking out non-redundant ones, as no theoretical upper bound on the number of components exists, hence the need for non-redundant dimensionality reduction algorithms.

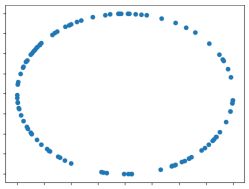

Figure 1: The circle.

1.2 Solution, Implementation and Experiments

To address non-redundancy, the author first converted the dimensionality reduction problems into a standard form of an optimization problem:

$$\text{For } i = 1,2,\ldots,d, \quad \max \mathbf{f}_i^T \mathbf{K} \mathbf{f}_i, \quad \text{subject to} \quad \begin{cases} \mathbf{f}_j^T \mathbf{f}_i = 0 & \forall j < i \\ \mathbf{f}_i^T \mathbf{f}_i = 1 \end{cases}$$

where $\mathbf{f}_0 = \mathbf{1}_n = [1\ 1\ 1 \ldots 1]^T$, with n being the number of samples in the input data and d being the dimension of the target space, i.e. the desired number of components. Note that in some methods such as LLE, HLLE and LTSA, the aim is to minimize $\mathbf{f}_i^T \mathbf{K} \mathbf{f}_i$, in which case re-defining $\mathbf{K} \leftarrow \lambda_{max}\mathbf{I}_n - \mathbf{K}$ where λ_{max} is the largest eigenvalue of $\mathbf{K}$ will turn the original problem to a maximizing one. The author formed the theoretical framework of their solution by first translating this problem to the language of probability theory:

$$\text{For } i = 1,2,\ldots,d: \quad \max \mathbf{E}\left[K(X_1,X_2) f_i(X_1) f_i(X_2)\right] \quad \text{s.t.} \begin{cases} \mathbf{E}\left[f_j(X) f_i(X)\right] = 0 \ \forall j < i \\ \mathbf{E}\left[f_i(X)\right] = 0, \ \mathbf{E}\left[f_i^2(X)\right] = 1 \end{cases}.$$

To ensure non-redundancy, the authors proposed changing the orthogonality condition $\mathbf{E}\left[f_j(X) f_i(X)\right] = 0 \ \forall j < i$ to *unpredictability*:

$$\mathbf{E}\left[f_i(X) \mid f_1(X), f_2(X), \ldots, f_{i-1}(X)\right] = 0. \tag{1}$$

This condition ensures that f_i cannot be a function of $f_1, f_2, \ldots, f_{i-1}$, with detailed proof in [1]. We checked and verified that the proof is correct. The authors then translate the problem back to the language of Linear Algebra by providing an approximation for Condition (1).

$$\mathbf{P}_i \mathbf{f}_i = 0, \quad \text{where} \quad [\mathbf{P}_i \mathbf{f}_i]_k \approx \mathbf{E}\left[f_i(X) \mid f_j(X) = \mathbf{f}_{jk} \ \forall j < i\right] \ \forall k \leqslant n. \tag{2}$$

The matrix $\mathbf{P}_i \in \mathbb{R}^{n\times n}$ is a smoothing matrix that can be constructed by

$$\begin{aligned}\mathbf{F}_i \triangleq \begin{pmatrix}\mathbf{f}_1 & \dots & \mathbf{f}_{i-1}\end{pmatrix} = \begin{pmatrix}\mathbf{f}_1^* \\ \vdots \\ \mathbf{f}_n^*\end{pmatrix}, \quad [\mathbf{P}_i^*]_{jk} \triangleq \exp\left(-\frac{1}{2h_i^2}\left\|\mathbf{f}_j^* - \mathbf{f}_k^*\right\|^2\right) \ \forall j,k \leqslant n, \\ [\mathbf{P}_i]_{jk} \triangleq \frac{[\mathbf{P}_i^*]_{jk}}{\sum_{k=1}^n [\mathbf{P}_i^*]_{jk}} \ \forall j,k \leqslant n.\end{aligned} \tag{3}$$

Note that $\mathbf{P}_0 = 0$. The *bandwidth parameter* h_i is chosen adaptively by $h_i \triangleq \frac{\alpha}{n}\|\mathbf{F}_i\|_F$ where $\|\cdot\|_F$ is the Frobenius norm. The translated linear algebra problem has the form

$$\text{For } i = 1,2,\dots,d, \quad \max \mathbf{f}_i^T\mathbf{K}\mathbf{f}_i, \quad \text{subject to} \quad \begin{cases}\mathbf{1}_n^T\mathbf{f}_i = \mathbf{P}_i\mathbf{f}_i = 0 \\ \mathbf{f}_i^T\mathbf{f}_i = 1\end{cases}. \tag{4}$$

A new problem arises: The solution to this problem is no longer the top d eigenvectors of the double-centered matrix $(\mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)\mathbf{K}(\mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)$. To bypass this, author claimed that the following problem yields the same solution (proof in [1, Appendix A]):

$$\text{For } i = 1,2,\dots,d, \quad \max \mathbf{f}_i^T(\mathbf{I} - \mathbf{V}_i\mathbf{V}_i^T)\mathbf{K}(\mathbf{I} - \mathbf{V}_i\mathbf{V}_i^T)\mathbf{f}_i, \quad \text{subject to} \quad \begin{cases}\mathbf{1}_n^T\mathbf{f}_i = 0 \\ \mathbf{f}_i^T\mathbf{f}_i = 1\end{cases}, \tag{5}$$

where $\mathbf{P}_i = \mathbf{U}_i\mathbf{\Sigma}_i\mathbf{V}_i^T$ is the *compact singular value decomposition* of $\mathbf{P}_i$, meaning $\mathbf{\Sigma}_i$ contains only non-zero singular values of $\mathbf{P}_i$. Their final non-redundant dimensionality reduction algorithm is:

Algorithm 1. Input. A data matrix $\mathbf{X}$ and a dimensionality reduction algorithm A.

1. Follow A to compute the kernel matrix $\mathbf{K} = \mathbf{K}(\mathbf{X})$.
2. If A aims to minimize $\mathbf{f}^T\mathbf{K}\mathbf{f}$, find $\lambda_{max} \triangleq \max \text{Eig}(\mathbf{K})$ and set $\mathbf{K} := \lambda_{max}\mathbf{I}_n - \mathbf{K}$.
3. Set $\mathbf{f}_1$ to be the top (non-trivial) eigenvector of $\mathbf{K}$.
4. For $i = 2,3,\dots,d$ do:
   - 4.1. Compute $\mathbf{P}_i$ from the procedure (3).
   - 4.2. Compute compact SVD: $\mathbf{P}_i = \mathbf{U}_i\mathbf{\Sigma}_i\mathbf{V}_i^T$.
   - 4.3. Set $\mathbf{f}_i$ to be the top eigenvector of $\widetilde{\mathbf{K}}_i \triangleq (\mathbf{I} - \mathbf{V}_i\mathbf{V}_i^T)\mathbf{K}(\mathbf{I} - \mathbf{V}_i\mathbf{V}_i^T)$.
5. **Output.** $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_d$.

With the above algorithm, the authors ran three experiments on three datasets and compared the performances usual dimensionality reduction methods with their non-redundant versions.

1.3 Contributions

In this report, aside from summarizing the paper, we point out several unconvincing points and missing details in Section 2, and give comments, based on our best understanding of the paper.

Perhaps the greatest disagreement we have with the authors is in their design of the non-redundant dimensionality reduction algorithm, particularly the choice of the new kernel $\widetilde{\mathbf{K}}_i := (\mathbf{I} - \mathbf{V}_i\mathbf{V}_i^T)\mathbf{K}(\mathbf{I} - \mathbf{V}_i\mathbf{V}_i^T)$. We emphasize that we do agree entirely with the underlying mathematical idea. We give a brief critical view of the algorithm in Section 2 and our revised algorithm in Section 3.

To verify our revised algorithm, we benchmark the non-redundant version it outputs against the one in the paper and the original for three datasets and several choices of original methods in Section 4.

Our experiments are design to mimic the procedures in the paper, with a different dataset, to validate the robustness of their algorithm and ours.

2 Critical Analysis

The content of this critical analysis is our subjective assessment of the paper, based on our best effort to be objective.

2.1 Missing details

The greatest missing details is perhaps the source code to run the experiments and produce the plots showed in the paper, including the ones not in Section 5, Experiments. We could find neither the source code nor the link to an online repository in the paper and the supplementary materials. This leads to difficulties in reproducing and verifying the results and aggravation of the absence of other details below.

In Fig. 1 of Section 1, the authors included a description of the Swiss Roll dataset being used as the benchmark, nor the parameters of the models being benchmarked. An adequate description should include the ratio of length versus width and spiraling rate of the roll. The best way to describe it should be a code snippet to generate the data and run the models on it. The same problem occurs for the Torus experiment in Section 2, Fig. 4. In Section 4 of this report, we reproduce the Swiss Roll experiment with our own code and give critical remarks.

In Section 5.1. and 5.2., the author mentioned the method *leave-one-out prediction* with a *Nadaraya-Watson regressor* [4, 12] to reconstruct images. In both places the authors did not include a description or a citation to a source for leave-one-out prediction, making it impossible to determine the precise meaning of the term. Most materials we found online only refers to "leave-one-out" as a cross validation method, indicating that "leave-one-out prediction" is not a sufficiently well-known method to nullify the need for a reference. Our extensive search led us to a description of the method in [3], which is indeed closely related to the cross validation method of the same name. Regarding the Nadaraya-Watson regressor, if the kernel is Gaussian, there is a crucial parameter known as the *bandwidth*, which dictates the outcome of the predictor. The sources cited by the authors seem to not specify how to choose the bandwidth, Watson [12] even proposed a more general regressor where the kernel is not fixed as Gaussian. Therefore, in our image reconstruction experiment (Section 4), we treat the bandwidth as a tunable hyper-parameter.

2.2 Critical comments on the algorithm

The version of the algorithm the author provided aims to solve the problem (5). They claimed the solution to be the top eigenvector of $\widetilde{\mathbf{K}}_i$, but this is not true. From the constraint $\mathbf{1}_n^T \mathbf{f}_i = 0$, the solution should be the top eigenvector of the matrix double-centered matrix $(\mathbf{I} - \mathbf{1}_n \mathbf{1}_n^T)\widetilde{\mathbf{K}}_i(\mathbf{I} - \mathbf{1}_n \mathbf{1}_n^T)$. The corrected algorithm is below.

Algorithm 2 (Correction of Algorithm 1)**.** Input: data matrix $\mathbf{X}$ and dimensionality reduction algorithm A.

1. Follow steps 1 and 2 of Algorithm 1 to get $\mathbf{K}$.
2. Set $\mathbf{H} := \mathbf{I} - \frac{1}{n}\mathbf{1}_n \mathbf{1}_n^T$.
3. Set $\mathbf{f}_1$ to be the top eigenvector of $\mathbf{HKH}$.
4. For $i = 2, 3, \ldots, d$ do:
   4.1. Compute $\mathbf{P}_i$ from the procedure (3).

4.2. Compute compact SVD: $\mathbf{P}_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^T$.

4.3. Set $\mathbf{f}_i$ to be the top eigenvector of $\mathbf{H}(\mathbf{I} - \mathbf{V}_i \mathbf{V}_i^T)\mathbf{K}(\mathbf{I} - \mathbf{V}_i \mathbf{V}_i^T)\mathbf{H}$.

5. **Output.** $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_d$.

Furthermore, we are not entirely convinced by the proof of correctness for this algorithm, in particular the transition from Problem (4) to Problem (5). The author claimed in its proof that if a vector $\mathbf{f}_i$ is a solution to Problem (4) that does not satisfy $\mathbf{V}_i^T \mathbf{f}_i = 0$, the solution

$$\widetilde{\mathbf{f}}_i \triangleq \frac{(\mathbf{I} - \mathbf{V}_i \mathbf{V}_i^T)\mathbf{f}_i}{\|(\mathbf{I} - \mathbf{V}_i \mathbf{V}_i^T)\mathbf{f}_i\|}$$

also satisfies the conditions $\widetilde{\mathbf{f}}_i^T \widetilde{\mathbf{f}}_i = 1$ and $\mathbf{1}_n^T \widetilde{\mathbf{f}}_i = 0$. However, we could not deduce that $\mathbf{1}_n^T \widetilde{\mathbf{f}}_i = 0$ by any logical argument. In fact, we performed a simple experiment, where we set $n := 1000$, generated $\mathbf{f}_0$ as a random vector from $(0, 1)^n$, compute $\mathbf{P}_1$ with $\alpha = 10$ and $\mathbf{V}_1$, then truncate $\mathbf{V}_1$ to singular values higher than 1% of the largest, and compute the value $\mathbf{1}_n^T(\mathbf{I} - \mathbf{V}_i \mathbf{V}_i^T)\mathbf{f}$ for a random vector satisfying $\mathbf{1}_n^T \mathbf{f} = 0$. The value we got was 0.386, which is not near 0 enough to justify the authors' claim. It seems that there were a few details missing in the proof and we need to contact the authors for clarification. Fortunately, we invented a modification to Problem (5) and Algorithm 1 whose correctness (as an approximation to Problem (4)) we were able to prove. Details are in Section 3. Note that we do not claim that the authors' algorithm or Problem (5) are wrong.

3 Our Redundancy Removal Algorithm

Now we present our version of non-redundancy removal algorithm. Our first step is to modify Problem (5). Our goal remains to solve Problem (4). The two constraints can be combined to become

$$\mathbf{Q}_i f_i = 0 \quad \text{where} \quad \mathbf{Q}_i \triangleq \begin{pmatrix} \frac{1}{n}\mathbf{1}_n^T \\ \mathbf{P}_i \end{pmatrix},$$

Consider the *full* (non-compact) SVD $\mathbf{Q}_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^T$. If $[\Sigma_i]_{kk} \approx 0$ for $k \geqslant k_0$, the rows of $\mathbf{W}_i^T = [\mathbf{V}_i^T]_{k \geqslant k_0}$ form an approximate orthogonal basis for the nullspace of $\mathbf{Q}_i$. Therefore Constraint (3) becomes $\mathbf{f}_i = \mathbf{W}_i \mathbf{g}_i$ for some $\mathbf{g}_i$. Our new problem is

$$\text{For } i = 1, 2, \ldots, d, \quad \max \mathbf{g}_i^T \left(\mathbf{W}_i^T \mathbf{K} \mathbf{W}_i\right) \mathbf{g}_i, \quad \text{subject to } \mathbf{g}_i^T \mathbf{g}_i = 1. \quad \text{Return } \mathbf{f}_i = \mathbf{W}_i \mathbf{g}_i. \quad (6)$$

Note that $\mathbf{W}_i^T \mathbf{W}_i = \mathbf{I}$, so $\mathbf{f}_i^T \mathbf{f}_i = 1 \Leftrightarrow \mathbf{g}_i^T \mathbf{g}_i = 1$, thus the new problem is indeed equivalent to Problem (4). The solution is simply the top eigenvector of $\widetilde{\mathbf{K}}_i \triangleq \mathbf{W}_i^T \mathbf{K} \mathbf{W}_i$. Our revised algorithm is given below:

Algorithm 3. Input. data matrix $\mathbf{X}$ and dimensionality reduction algorithm A.

1. Follow steps 1 to 3 of Algorithm 2 to get kernel matrix $\mathbf{K}$, centering matrix $\mathbf{H}$ and $\mathbf{f}_1$ as the top eigenvector of $\mathbf{HKH}$.
2. For $i = 2, 3, \ldots, d$ do:
   2.1. Compute $\mathbf{P}_i$ from the procedure (3). Set $\mathbf{Q}_i \triangleq \begin{pmatrix} \frac{1}{n}\mathbf{1}_n^T \\ \mathbf{P}_i \end{pmatrix}$.
   2.2. Compute full SVD: $\mathbf{Q}_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^T$.
   2.3. Set $\mathbf{W}_i^T$ to be the rows of $\mathbf{V}_i^T$ corresponding to singular values below a threshold.
   2.4. Set $\mathbf{g}_i$ to be the top eigenvector of $\widetilde{\mathbf{K}}_i \triangleq \mathbf{W}_i^T \mathbf{K} \mathbf{W}_i$ and $\mathbf{f}_i := \mathbf{W}_i \mathbf{g}_i$.
3. **Output.** $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_d$.

Beside being able to see immediately that this algorithm approximates the solution to Problem (4), it apparently does slightly fewer computations than Algorithm (2). The computations are nearly identical up to Step 2.2 (computing the SVD of an $n \times n$ matrix is approximately the same as that of an $(n+1) \times n$ matrix), after which Algorithm (2) does four matrix multiplications and one subtraction, namely $\mathbf{A} \leftarrow \mathbf{V}_i \mathbf{V}_i^T$, $\mathbf{A} \leftarrow \mathbf{I} - \mathbf{A}$, $\mathbf{A} \leftarrow \mathbf{HA}$, $\mathbf{T} \leftarrow \mathbf{AK}$ and $\mathbf{T} \leftarrow \mathbf{TA}$, while Algorithm 3 does two matrix multiplications, namely $\mathbf{T} \leftarrow \mathbf{W}_i^T \mathbf{K}$ and $\mathbf{T} \leftarrow \mathbf{TW}_i$, and one matrix-vector multiplication, $\mathbf{f}_i \leftarrow \mathbf{W}_i \mathbf{g}_i$. Thus we believe Algorithm 3 will have a slightly better running time.

4 Experiments

We perform two experiments, the first is intended as a test to verify the redundancy phenomenon and our algorithm's ability to avoid it, the other is an attempt to reproduce Experiments 5.2 from the paper [1], with a different image to test for robustness of the model and our implementation (Algorithm 3).

4.1 Narrow Swiss Roll with noise

The dataset resembles a Swiss roll that is much narrower than the one provided by `scikit-learn`[5], with Gaussian noise of standard deviation 0.5. The underlying distribution is:

$$\begin{gathered} \left\langle \tfrac{L}{L_0}\theta\cos\theta + \xi_1,\ \tfrac{L}{L_0}\theta\sin\theta + \xi_2,\ W\cdot Z + \xi_3 \right\rangle, \\ \text{where } \ \theta \sim \mathrm{Uni}\left(\tfrac{\pi}{2}, 4\pi + \tfrac{\pi}{2}\right),\ Z \sim \mathrm{Uni}\,(0,1),\ \xi_i \sim \mathrm{N}(0,\sigma) \ \text{ independent}, \end{gathered} \tag{7}$$

where L and W are respectively the desired length and width of the roll, $L_0 = 2\pi\sqrt{16\pi^2+1} + \frac{1}{2}\ln\left(4\pi + \sqrt{16\pi^2+1}\right)$ being the length of the unscaled roll ($\langle \theta\cos\theta, \theta\sin\theta\rangle$ for $\theta \in [0, 4\pi]$). We choose $L = 60$, $W = 10$ and $\sigma = 0.5$. **Note:** All dimensionality reduction methods performed well with little redundancy for the Swiss Roll generated by `scikit-learn`, so we have to generate a narrow one. We sampled 2500 points from distribution (7) and benchmarked the performance of our non-redundant LLE, LTSA, HLLE against their original versions. The number of nearest neighbors to consider is 15 for LLE and 20 for the rest. Results are reported in Figure 2.

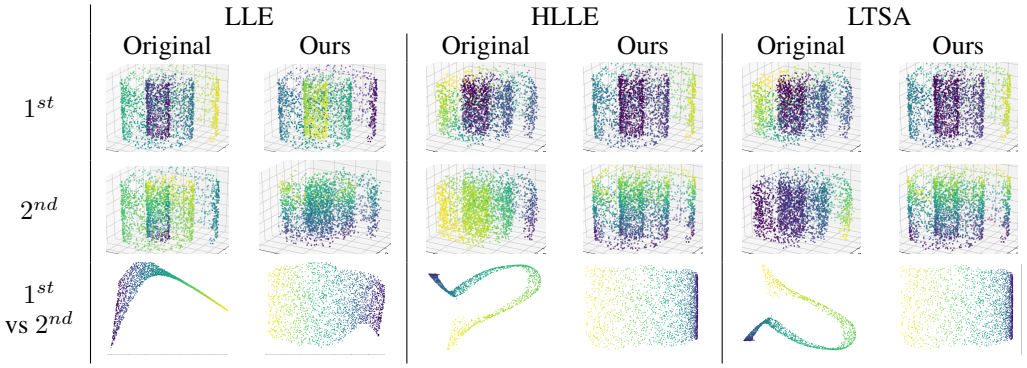

Figure 2: Performance of dimensionality reduction methods and their non-redundant versions

The first and second rows respectively plot the original data with each point colored by the value of its first and second projections. The third row plot first and second projections against each other, colored by the points' original color. Results show that original methods produce two components that both capture the variation along the length of the roll, while their non-redundant counterparts capture both the length and width. **Note.** Isomap performed well with low redundancy even for this narrow Swiss roll, so we did not observe an apparent improvement with the non-redundant version.

4.2 Image reconstruction

We took a colored image of the White House at [2], turned it into a greyscale image, extracted all 20×20 patches with 10×10 overlap and computed the three-dimensional embeddings of these patches with Isomap, LLE and their non-redundant versions. We then took the three resulting components as input to a leave-one-out prediction algorithm [3] to reconstruct the patches. This predictor uses the Nadaraya-Watson regressor with a Gaussian kernel [11]. We choose the bandwidth for this Gaussian kernel to be 0.01 for all experiments. The reconstructed patches are used to reconstruct the image by averaging overlapping patches. **Note:** higher values for the bandwidth tend to produce lower-quality reconstructed images. Figure 3 shows the top three components as images.

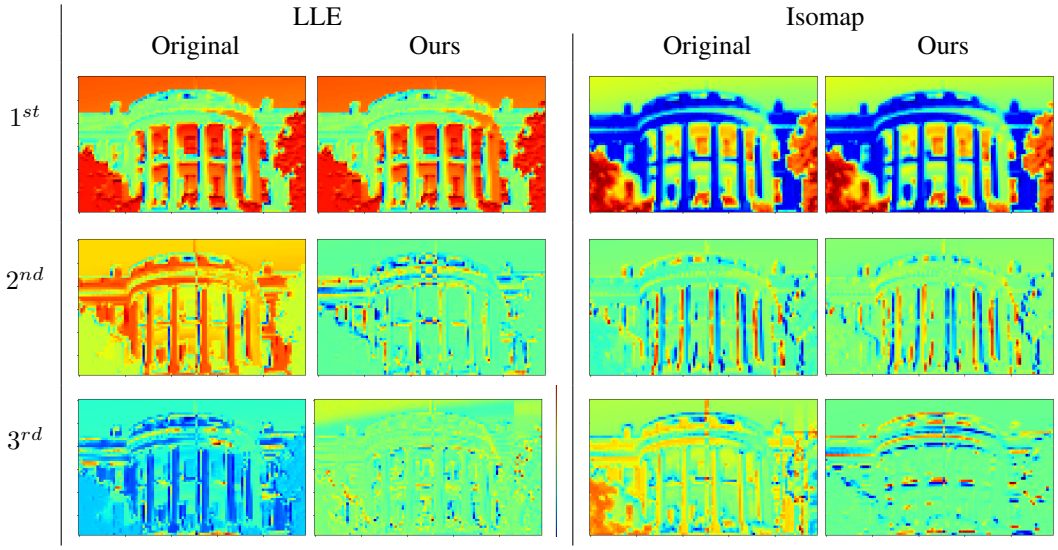

Figure 3: Top three components as produced by LLE, Isomap and their non-redundant versions

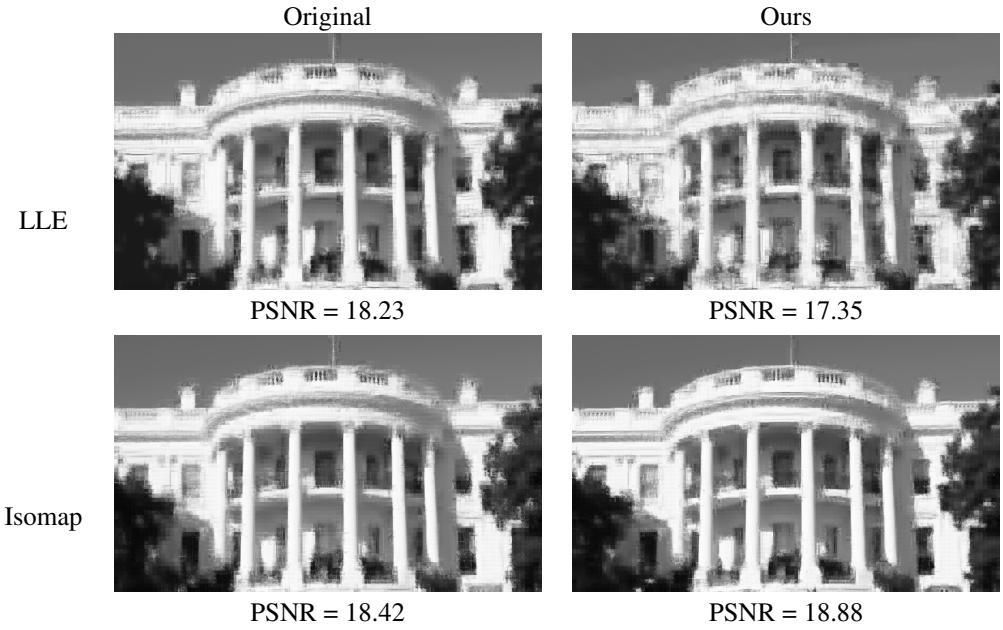

Figure 4: Reconstructed images from the top three components

From Figure 3, it is clear that our versions of LLE and Isomap achieve non-redundancy. The three components from normal LLE all capture lighting, with a few random edges. The second component from normal Isomap largely captures vertical edges with some lighting, while the three components from our version distinctly capture lighting, vertical edges and horizontal edges.

From Figure 4, we observe that our version of Isomap yields the best-looking result with the highest peak signal-to-noise ratio (PSNR). However, our version of LLE has the worst performance, with the lowest PSNR. This is consistent with Figure 3, where its three components, despite being non-redundant, do not adequately capture edges. The normal LLE, despite redundantly capturing lighting, does capture edges fairly well. In general, Isomap performs better than LLE, which seems contradictory to the authors' claim [1] that the image patches lie on a linear manifold which Isomap are not equipped to handle well. We speculate that this manifold is highly non-convex, while LLE is only equipped to handle locally convex manifolds (since the weight matrix it constructs is constrained to be stochastic), hence the lower performance.

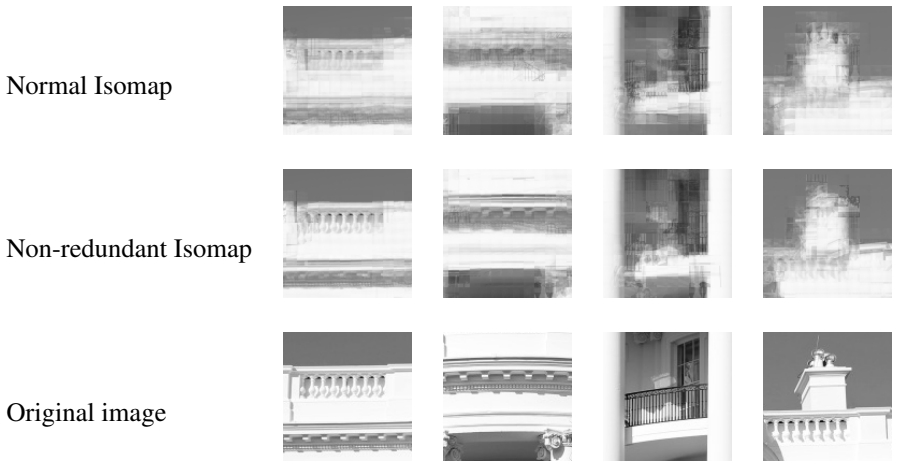

Figure 5: Comparison of Isomap and our version

Figure 5 showcases in more details the advantage of our non-redundant Isomap over the normal. The photos on the right show the same four areas extracted from respective reconstructed images. Not only does it show sharper horizontal edges, our Isomap produces some small details blurred by the normal Isomap.

5 Time Analysis and Conclusion

The table below shows the average running time for each model in Experiments 4.1 and 4.2.

Experiment	Isomap		LLE		HLLE		LTSA	
(seconds)	Original	Our	Original	Our	Original	Our	Original	Our
4.1	N/A	N/A	0.5	8.3	1.1	9.0	0.8	8.7
4.2	61	209	49	203	N/A	N/A	N/A	N/A

Our implementations of the non-redundant versions seem to run much slower than the original methods. This is understandable as computing each component separately while doing extra matrix multiplication and SVDs takes much more work than computing the same number of components all at once. Nevertheless, we will have to optimize our implementations significantly in order for it to be practical. The authors claimed to have a more optimized implementation [1], but since they did not discuss running time anywhere, it is unknown whether their non-redundant

version achieves reasonable running time. Overall, their results in the experiments, backed by mathematical justification, show great potential for widespread applications in manifold learning, provided advancements in computing technologies.

References

[1] Yochai Blau and Tomer Michaeli. Non-redundant spectral dimensionality reduction. *Lecture Notes in Computer Science*, page 256–271, 2017.

[2] Geoff Burke. President donald trump invites chiefs to visit white house, 02 2020.

[3] Wen Sui Liu. Calculate leave-one-out prediction for glm, December 2015.

[4] E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

[5] 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.

[6] Jianzhong Wang. *Hessian Locally Linear Embedding*, pages 249–265. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[7] Jianzhong Wang. *Isomaps*, pages 151–180. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[8] Jianzhong Wang. *Laplacian Eigenmaps*, pages 235–247. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[9] Jianzhong Wang. *Local Tangent Space Alignment*, pages 221–234. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[10] Jianzhong Wang. *Locally Linear Embedding*, pages 203–220. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[11] Larry Wasserman. Nonparametric regression, 36-708 statistical methods for machine learning, January 2019.

[12] Geoffrey S. Watson. Smooth regression analysis. *Sankhyā, Ser. A*, 26:359–372, 1964.