

Visual Analytics: Project Report

A Visual Exploration of Manifold Learning for Images

Mattia Bruno Stellacci

stellacci.1992018@studenti.uniroma1.it

Abstract

The following work is a report on the development and implementation of a visual analytics system intended to aid the understanding of dimensionality reduction techniques. In creating an interactive interface to various embeddings of the well-known MNIST handwritten digit dataset we propose a system which allows users to manipulate the embedded representations using a web-based application. The system further allows its users to select the embedded images and to aggregate them using standard image processing algorithms to emphasize the technique's inner workings and visually aid the development of intuitions.

1 Motivation, Intended users and preliminary Research

1.1 Motivation

The creation of the system was motivated by the desire to create an aid to understanding dimensionality reduction and the manifold hypothesis, as these topics seem to be strangely ubiquitous in current development in the field of Computer Science.

Reducing the extrinsic dimensionality of data has many advantages, not only for processing efficiency, but also for visualization and algebraic manipulation of higher-dimensional manifolds. Images are a prime example of the discrepancy that can exist between the intrinsic and extrinsic complexity of data. Handwritten digits specifically lend themselves to a use as toy data. Especially, because as humans we can easily comprehend them, despite hundreds of degrees of freedom. The intention was to create a system, that would not only visualize the embeddings generated by various dimensionality reduction techniques in a common cartesian chart, but would further allow the user to select points in the embedded space. The user could then manipulate them and aggregate the instances they

represent using image processing techniques such as morphology. This last aspect was based on the expectation, that visualizing aggregations of the embedded instances, while knowing their position in the embedded space, could yield insights on the inner workings of the techniques employed. Exposing characteristics such as linearity in the embedded space or the features represented by the degrees of freedom in the embedding.

The Manifold Hypothesis

In referring to the *Manifold Hypothesis* we intend the belief, that data recorded from the natural universe, despite a high dimensionality (think of the approx. 48000 *DOF* in a single sample of the 48KHz audio recording or 3 million *DOF* in a 1MP color image) lies on a lower-dimensional manifold. This belief has been extensively studied by papers such as (Fefferman et al., 2016) (which cites a plethora of other papers on the field) and though fascinating and interesting in its own right, it is the basis of many techniques used in Machine Learning and Data science as these fields tend to deal with highly dimensional data. Notable examples are the random sampling over a latent space used in generative models (Razavi et al., 2019) or generating plausible interpolations/deformations of instances, exploiting the euclidean nature of the latent space (Cosmo et al., 2020). While these are very interesting examples of advanced applications of the manifold hypothesis, the application proposed here intends to lay the groundwork to its understanding/intuition.

1.2 Goals

In an effort to create an accessible educational tool, one of the main goals was to build a powerful application that could run within the browser. As there would be image processing required for the various image aggregation techniques, the idea was

to incorporate an OpenCV build for *WebAssembly* to move any image processing to the frontend application. More on this in the section on the implementation.

1.3 Target Audience

Being a educational tool at it's core, the application is geared towards anyone interested in deepening their understanding of dimensionality reduction or even comparing existing techniques with a specific use-case in mind. This includes Machine Learning Engineers/Data Scientists and students of the STEM fields.

Though the prototype of this application developed in the course of this project uses the well-known MNIST handwritten digit dataset as toy data to demonstrate the working principle with, the approach and large parts of the codebase could easily be adapted for custom image datasets and/or tailor made dimensionality reduction techniques in a productive machine learning environment

1.4 Related Work and Preliminary Research

While searching for related work in the field of Visual Analytics, it immediately became apparent that dimensionality reduction is a very common tool for data visualization which given it's expressive potential is unsurprising. Having a closer look at publications mentioning dimensionality reduction, a certain timeline emerges with respect to the role that it has played in scientific publications:

- 2000-2010: While Principal component analysis and Multi-dimensional scaling are methods that have been around since the last century, many publications in this period motivate/discover the techniques themselves and their potential for various applications. Many of these papers explore the mathematical properties of the lower dimensional embeddings. Examples include: The original paper presenting the Isomap embedding (Tenenbaum et al., 2000), or the Locally Linear embedding (Lle)(Roweis and Saul, 2000). Contrary to this earlier focus on the techniques involved, towards the end on this decade, dimensionality reduction seems to have become pretty much standard practice for data visualization. Papers such as this survey (Zhang et al., 2010) suggest they are frequently used to visualize data in scientific publications.

- 2010-Present: With Big Data and Machine Learning becoming ever more relevant topics, dimensionality reduction gained more traction not only in it's capability to make data more comprehensible/intuitive to humans, but also as a means of counteracting the proverbial *curse of dimensionality*. Even more recently the focus seems to have once again shifted, as new models allow the enforcing of certain properties in the latent space, such as the preservation of semantics in *word2vec*(Mikolov et al., 2013) word embeddings. These machine learning based techniques, allow the transformation to be conditioned on desired properties, thereby extending the realm of possible applications.

While we couldn't find a work in the field of Visual Analytics that combines dimensionality reduction and image processing in the interactive way envisioned here, it is apparent that the underlying ideas are not new and that the techniques are still very relevant.

2 Implementation

2.1 Tech stack

In the following we will briefly address the technologies chosen to implement the prototype of the system proposed.

- Dataset We chose the well-known MNIST handwritten digits dataset as toy data, as the images are small and the classification problem of handwritten digits is intuitive.
- Frontend As we decided to build the application to run in the browser, a powerful framework was required to enable the features we envision. It was therefor decided that the advanced functionality of `ReactJS` would offset the additional effort involved in the initial setup.
- Charting Library: As a solution to creating charts `react-vis` was chosen. First published by Uber in 2016 and sadly deprecated in 2021, `react-vis` is a simple charting library, built for extensibility and interactivity.
- Backend: In an effort to prototype rapidly, there seemed to be no incentive to handle the dimensionality reduction in the browser, as far more powerful and versatile solutions exist for

Python. We therefor decided to build a simple HTTP-server based on the `bottle` package to serve the data and files. The dimensionality reduction techniques included were sourced from the `scikit-learn` library.

- **Image processing** To perform the image aggregation on the original instances we chose to use `OpenCV`. Initially it was our intention to perform all the processing in the browser, using a `WebAssembly` build of `OpenCV`. For practical reasons, it later became apparent that it would be much more efficient to simply refer to images by a common index(between frontend and backend) and let the frontend trigger the aggregation of instances in the backend as required.

2.2 Dimensionality Reduction Techniques

While we set out with the initial intention of incorporating approx. 5 different dimensionality reduction techniques, we limited ourselves to 3 in the prototype. We included

- Principal Component Analysis
- Locally linear embedding
- Isomap embedding

While it is desirable to incorporate and showcase as many different techniques as possible, it became apparent in the development process, that having too many different techniques wouldn't benefit the app as much as focussing on specific techniques and tailoring the application towards showcasing their properties as well as possible.

A property that wasn't sufficiently considered during planning, was whether the incorporated techniques have an inverse transformation or not. More on this in the section on insights gained.

3 The Prototype

3.1 Usage

When navigating to the page of the prototype the user immediately sees the user interface which can roughly be separated into 5 parts. In the following I will briefly address their function:

Search field

The page's header contains a field allowing the user to select the classes of digits they desire the backend to sample from, before calculating and

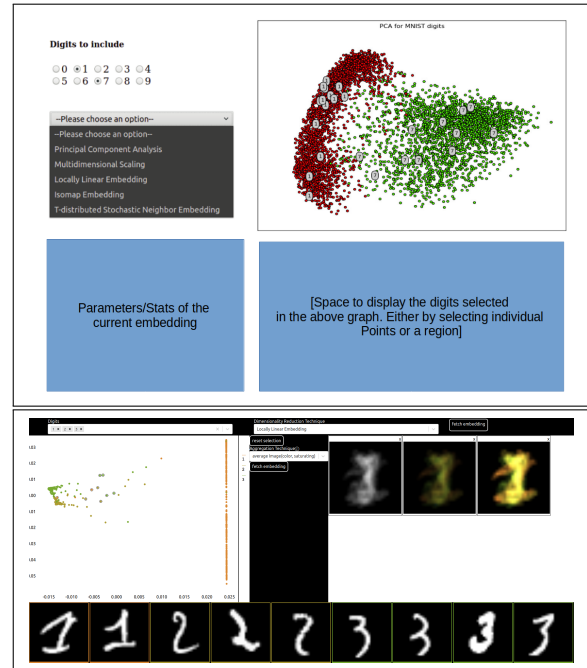


Figure 1: A picture of the proposed mockup vs. screenshot from the prototype of the application

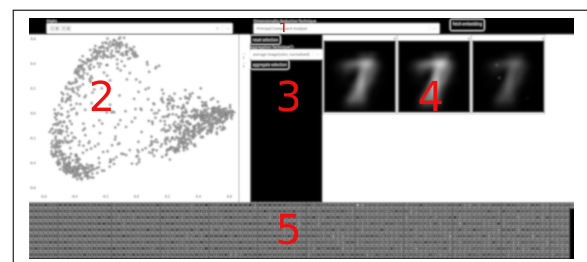


Figure 2: The 5 main components of the application, as described in section 'Usage'

returning a 2-dimensional embedding. Leaving the field empty will result in all 10 digit-classes being included. A drop-down field allows the user to select the dimensionality reduction technique they wish to employ.

The Chart

This section contains the main chart, displaying colour-coded dots representative of all instances embedded in the calculation. The dots in the chart can be selected, either by clicking on them individually, *dragging* a rectangular selection over a region of the graph(similar to selecting multiple files in a OS's file explorer) or by clicking the legend. Hovering a point displays the instance it represents, right-clicking it will remove it from the selection if it is currently selected. The scale of the axes is automatically set based on the output of the calculation and the classes are colour-coded. To the right there is a (small) legend containing all digits displayed and their color. Hovering an entry in the legend highlights only all instances of the class it represents. Clicking on an item adds all instances in that class to the current selection.

The Aggregation Parameter Section

To the right of the chart visualization there is a column that allows the user to select how they wish to aggregate the selected instances. Aggregation techniques in the prototype are currently limited to various types of average images(calculated over the user's current selection of instances). They are:

- **Average Image(Monochrome)** All images are added and the mean is calculated for every single pixel value. This can emphasize certain similarities in the input but, has a tendency to blur the results
- **Average Image(Colour, Saturating)** Mean images are calculated per class(as above). These are then converted to colour(the colour is coordinated with the colour their class is represented by in the chart) and the resulting average images are additively overlaid. As this method tends to 'blow out'/saturate high activity areas in the image, it becomes difficult to interpret. For large selections it might induce false colours.
- **Average Image(Colour, normalized)** Similar to above, mean images are calculated for every class. In the aggregation step any class's

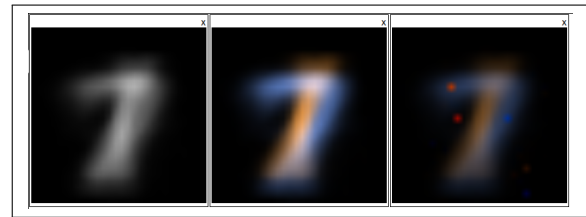


Figure 3: Examples of the three aggregation techniques over 1000 instances sampled equally from class 1 and 7

mean image is converted to HSL, it's luminance is scaled to be proportionate to the current class's relative representation in the user's selection, i.e. the ratio between images pertinent to that class and the total count of images in the selection. This approach avoids saturating the aggregated image, but results in visually dim results(particularly for the darker coloured classes. In some cases, stray pixels in the per-class averages seem to introduce false-color noise in the results. See Fig 3.)

The Aggregated Image View

In the right column of the application's main row there is field where the aggregated instances get displayed. The size of these images is calculated dynamically to best fit the content. Hovering over an image displays the technique it was calculated by and the amount of instances that went into it's calculation.

The Selected Instance Container

The bottom section of the application contains a field displaying the user's current selection of instances. As above, the size of individual entries is calculated dynamically for optimal layout. Instances are sorted by class(in alphabetical order) and within their class they are sorted by first their x and then their y coordinate in the embedded space. Hovering an instance will highlight it's position in the chart, right clicking it will remove it from the current selection.

Intended Usage

The app is intended to be used by first generating an embedding selected from the options available. The user can then visually/spatially explore the resulting embedding both with respect to the spatial distribution in the embedded space, as well as view

and aggregate the instances that led to the displayed mapping. Sufficient knowledge on the dimensionality reduction techniques is required to make sense of the application's output.

4 Insights

Considering the version 1 of the Visual Analytics system described in this work, it is the author's opinion, that the system, though promising in its interactive approach to dimensionality reduction, lacks in some aspects which could be improved upon in a second iteration. These aspects will be addressed in this section.

Interpretability

While there are many properties of dimensionality reduction techniques that the system is capable of showcasing, it takes some knowledge of these techniques to configure the system to make them emerge. The same is true for the general interpretability of the results. The system is therefore not self-explanatory. While a certain knowledge as prerequisite is inevitable when dealing with mathematical models, this aspect could possibly be improved upon by briefly explaining the algorithms used, the composition of the data and parameters that the embedding was calculated for. A possible solution would be employing a *onHover*/tooltip similar to the one describing the aggregation techniques in the app's central column

Scalability and Performance

In an effort to put the system online and thus make it available to anyone interested, it would have been desirable to do away with the need for a backend performing the heavy calculation on data and image aggregation. As this poses a significant challenge from a development point of view and would probably require an advanced use of sophisticated web technologies such as *WebAssembly*(WASM) and some sort of numerical JavaScript library, it was unfeasible as part of the work on this project. Pragmatically speaking, improved convenience to the interested user could likely be achieved by creating a docker-file that runs the entire application(frontend and backend).

Comparability

One of the principal goals of this project, was to create a system allowing the user to compare various dimensionality reduction techniques, possibly

even to the end of determining the best-suited one for a given application. While it is possible to either open the web-page multiple times and run different algorithms in different tabs or by simply running the calculations one after the other in the same page, it takes some imagination and domain-specific knowledge to interpret the results correctly. The comparability of the results is limited by the following factors:

- **Sampling:** As it is impractical and unnecessary to run even the "easier"(computationally lighter) techniques on all 70K instances, a random subset is sampled every time the user calls the backend for a new embedding. This introduces variability between runs, due to a poor design decision which should be easy enough to correct, possibly by fixing the random seeds of the sampling based on a session ID.
- **Synchronization:** It would be desirable to have these graphs be synchronized, so that hovering the selection/legend reveals the pertinent instances in multiple embeddings being displayed at the same time. From this point of view both the temporal comparison(running the algorithms one after the other) as well as the multi-tab solution fail. While the modular nature of *ReactJS* and some foresight during development actually make this problem not too difficult to fix, the difficulty arises with respect to layout, as the user interface is already pretty busy and loaded with information. Toggling a multi-chart view vs. a chart+aggregated instances view would be a technically plausible and interesting modification to explore moving forward.

Consistency

Trying to be universal, i.e. suiting as many dimensionality reduction techniques as possible, while maintaining equal functionality for all, proved to be a great challenge. This is true both for technical reasons regarding *scikit-learn* as well as theoretical limitations inherent to the techniques involved.

The technical difficulties stem from a design decision to calculate the embedding based on a balanced subset of (currently) 3000 instances representing the user-selected classes and then returning a fixed amount of items per class. While this makes sense from the point of view of robustness, it requires first calculating a transformation and

then applying it, which is not supported by all classes in `scikit-learn`, as some only feature `.fit_transform()` as opposed to the separate `.fit()` and `.transform()` functions. This is just one example of (surpassable) technical difficulties arising while reconciling the various embedding techniques. Another would be the sample size that the transformation is determined for. While 3000 instances is reasonable (on my 2016 Thinkpad, i7, 16GB ram) for PCA, calculations are unacceptably slow for other methods such as Multidimensional scaling or the T-stochastic Neighbour Embedding.

The last and possibly most important aspect of diversity to address, is the invertibility of the transformations calculated. Being able to reverse the embedding for some techniques is a fundamental aspect that differs between embeddings, as some are well suited to distribute data in lower dimensions, but are not invertible. There are many features that could have enriched the application and seized its interactivity to the fullest, had it been designed with only invertible embeddings (e.g PCA, Variational Autoencoders etc.) in mind. Possible examples are clicking any point in the chart and viewing the instance that it would correspond to in the data's original space, viewing exaggerated (by means of image morphology) difference maps between input instances and their reconstruction loss, and many more. This highly interesting approach would require creating a separate application, just for invertible embeddings as to not make the app too confusing. Though absolutely feasible, given enough time, it would further increase the complexity of both the backend and frontend of the existing application.

5 Conclusion

While the work on this Visual Analytics system was interesting and rewarding, the prototype developed suffers from some shortcomings owing to a lack of "focus" in the development phase. It is the author's opinion, that *narrowing* the application down to either broad comparability of diverse embedding techniques (possibly at the expense of some aggregation features) or focussing solely on invertible transformations (and enriching the aggregating functionality accordingly) would make better use of this application's great interactive, visual potential.

References

- Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. 2020. [LIMP: Learning Latent Shape Representations with Metric Preservation Priors](#). volume 12348, pages 19–35. ArXiv:2003.12283 [cs, stat].
- Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. 2016. [Testing the manifold hypothesis](#). *Journal of the American Mathematical Society*, 29(4):983–1049.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient Estimation of Word Representations in Vector Space](#). ArXiv:1301.3781 [cs].
- Ali Razavi, Aaron van den Oord, and Oriol Vinyals. 2019. [Generating Diverse High-Fidelity Images with VQ-VAE-2](#). ArXiv:1906.00446 [cs, stat].
- Sam T. Roweis and Lawrence K. Saul. 2000. [Non-linear Dimensionality Reduction by Locally Linear Embedding](#). *Science*, 290(5500):2323–2326. Publisher: American Association for the Advancement of Science.
- J. B. Tenenbaum, V. de Silva, and J. C. Langford. 2000. [A global geometric framework for nonlinear dimensionality reduction](#). *Science (New York, N.Y.)*, 290(5500):2319–2323.
- Junping Zhang, Hua Huang, and Jue Wang. 2010. [Manifold Learning for Visualizing and Analyzing High-dimensional Data](#). *IEEE Intelligent Systems*, page 5401149.