Dimensionality Reduction Techniques: 
An Algorithmic Analysis

Justin W. Rachels

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2004
Abstract

The fields of dimensionality reduction and manifold learning are becoming increasingly important in our data-rich world. Though many excellent algorithms exist with which to pursue the goals of these fields, there is little in the way of comparative analysis in the current body of literature. We posit a framework for considering the limitations, weaknesses, and strengths of the algorithms within these fields. We then derive neutral evaluative metrics based upon this framework and compare two major core algorithms, the Isomap and LLE non-linear dimensionality reduction techniques, with an experimental suite. Using this experimental data and guided by our analysis framework, we posit useful extensions and applications for the algorithms and offer, as well as test, modified versions of the algorithms.
Acknowledgements

I would like to very sincerely thank Graham McNeill for his extremely helpful guidance, especially concerning background research, and for keeping my focus well-directed as I developed, and moved outwards from, the project’s foundational perspectives.

I would also like to thank Mykel Kochenderfer for his suggestions and critiques and, of course, my supervisor, Dr. Sethu Vijayakumar, without whom this project would not have been possible.
Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Justin W. Rachels)
# Table of Contents

I. Introduction . . . . . . . . . 1
  I.1. Motivation and Perspectival Scope . . . . . . . . . 1
  I.2. Dissertation Overview . . . . . . . . . . . . . 3

II. Algorithmic Overview . . . . . . . . . . . . . . . . . 4
  II.1. Dimensionality Reduction and Manifold Learning—Definitions 4
  II.2. Linear Dimensionality Reduction Techniques . . . . . . 6
    II.2.1. MDS . . . . . . . 6
    II.2.2. PCA . . . . . . . 7
  II.3. Non-Linear Dimensionality Reduction Techniques . . . . 8
    II.3.1. Isomap . . . . . . . 8
    II.3.2. LLE . . . . . . . 9

III. Scenario Analysis . . . . . . . . . . . . . . . . . . . . 12
  III.1. General Application Concerns . . . . . . . . . . . 12
  III.2. Input Concerns . . . . . . . . . . . . . . . . . . 14
  III.3. Output Concerns . . . . . . . . . . . . . . . . . . 16
  III.4. Pathological Situations . . . . . . . . . . . . . . . 18

IV. Gauging Performance . . . . . . . . . . . . . . . . . . . 20
  IV.1. Mean Squared Error (MSE) . . . . . . . . . . . . . 20
  IV.2. Normalised MSE (nMSE) . . . . . . . . . . . . . . 21
  IV.3. Auto-Transformed MSE . . . . . . . . . . . . . . . 22
  IV.4. Data Coherence Measure (DCM) . . . . . . . . . . . 23

V. Core Experiments . . . . . . . . . . . . . . . . . . . . . . 24
  V.1. Experiment Outline . . . . . . . . . . . . . . . . . . 24
  V.2. Neighborhood Size . . . . . . . . . . . . . . . . . . 27
  V.3. Expected Error Values . . . . . . . . . . . . . . . . . 30
  V.4. Core Experiment Results . . . . . . . . . . . . . . . 30
  V.5. Extrapolations from Core Experiments . . . . . . . . . 47

VI. Algorithmic Modifications . . . . . . . . . . . . . . . . . 49
  VI.1. A Conjectured Algorithmic Modification for Proximity Issues 61
Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LLE Description 1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>LLE Description 2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Illustrations for Pathological Transformation Functions</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>Illustrations of Effect of Error Introduction for Various Error Levels on $z = x^2$ Transformation</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>Auto-Transformation Functions Used in Core Experiments</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>Relation between $k$ and $\varepsilon$ for Increasing Error Levels</td>
<td>28</td>
</tr>
<tr>
<td>7</td>
<td>Unscaled and Scaled MSE Measures for k-LLE over $z = x^2$ Transformation</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>$n$MSE Measures for Unscaled, Scaled, and Proportionately Scaled Data for k-LLE over $z = x^2$ Transformation</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>Alternate MSE Measures and Data Coherence Measure Graphs for k-LLE over $z = x^2$ Transformation</td>
<td>35</td>
</tr>
<tr>
<td>10</td>
<td>MSE Methods for k-LLE over $z = x^2$ Transformation</td>
<td>37</td>
</tr>
<tr>
<td>11</td>
<td>Emulated Best-Fit MSE for k-LLE over $z = x^2$ Transformation</td>
<td>38</td>
</tr>
<tr>
<td>12</td>
<td>Data Coherence Measure for k-LLE over $z = x^2$ Transformation</td>
<td>38</td>
</tr>
<tr>
<td>13</td>
<td>Comparative Algorithmic Performance on Different Data Set Transformations</td>
<td>40</td>
</tr>
<tr>
<td>14</td>
<td>Comparisons of Algorithmic Performance by Algorithm</td>
<td>43</td>
</tr>
<tr>
<td>15</td>
<td>MSE and DCM Values for Uniform and Variable Sampling Density Data Sets using k-LLE Algorithm over $z = x^2$ Transformation</td>
<td>53</td>
</tr>
<tr>
<td>16</td>
<td>DCM Values Per Point for Uniform and Variable Sampling Density Data Sets using k-LLE Algorithm over Noiseless $z = x^2$ Transformation</td>
<td>54</td>
</tr>
<tr>
<td>17</td>
<td>DCM Values Per Points Constrained by y-axis Range of [-2,2] for Uniform and Variable Sampling Density Data Sets using k-LLE Algorithm over Noiseless $z = x^2$ Transformation</td>
<td>55</td>
</tr>
<tr>
<td>18</td>
<td>Edge Effect Illustration— MSE and DCM Values Per Point for LLE and Isomap over Noiseless $z = x^2$ Transformation</td>
<td>57</td>
</tr>
<tr>
<td>19</td>
<td>Illustration for Momentum Algorithm Modification</td>
<td>62</td>
</tr>
</tbody>
</table>
Figure 20: Error Results for Momentum-Variant of k-LLE Algorithm for “Swiss Roll” Transformation . . . . . . . . . . . . . . . . . . . . 64

Figure 21: Error Results for Alternate-Neighborhood-Selection-Variants of k-LLE Algorithm for “Swiss Roll” Transformation . . . . . . . . . . . . . . . . . . . . 66
Chapter I. Introduction

Dimensionality reduction techniques (sometimes used seemingly interchangeably with the concept of “manifold learning”), are becoming increasingly important as the technologies and methodologies used for data gathering improve. Novel fields and new well-springs of rich data begging analysis are perfect focal points for the elucidating lens of these techniques. Basically termable as the algorithms used to make sense of high-dimensional raw data, converting it into a more accessible low-dimensional form, these procedures have found increasing application—especially across the fields of theoretical algorithmics, visualisation, and artificial intelligence.\(^1\) Many of these algorithms exist, from the classical linear methods of Multi-Dimensional Scaling (MDS) and Principal Component Analysis (PCA), to their popular non-linear extensions, Isomap [Tennenbaum] and Local Linear Embedding (LLE) [Roweis], as well as other extensions of the theories justifying these core techniques such as Charting algorithms [Brand], Eigenmap-based algorithms [Belkin], and Semi-Definite Embedding (SDE) [Weinberger]. We will, of course, be unable to analyse the entirety of the field here, but, at minimum, would like to introduce some potential guiding concepts for such an analysis and focus on the more well known algorithms and their extensions.

I.1. Motivation and Perspectival Scope

It is actually an interest in unsupervised online adaptive systems which has piqued our curiosity about dimensionality reduction and provided the starting vector for this project. Given an elementary structural black-box style theory of such a system, where environmental inputs are linked to a system's output spectra through

\(^1\) See [Hadid], [Riden], and [Yang] for examples of these applicative senses.
some learning algorithm\textsuperscript{2}, a direct network of mapping between the input and output
to whatever unit function descriptors are used as system components) function of
of such mappings\textsuperscript{3}. It would be practical, if not absolutely necessary due to
computational resource or memory storage concerns, to be able to describe these
high-dimensional mappings, where possible, in a more compact representation:
eliminating extraneous pseudo-relations, decreasing storage requirements, increasing
conjecturability towards relationships dealing with novel stimuli or uncharted
sections of the system's state space, and perhaps decreasing the opacity of the
system's internal structure due to this high level of inherent dimensional complexity.

This motivation (similar to the motivations driving the fields of both
visualisation and data compaction) has brought us to the field of dimensionality
reduction and manifold learning techniques as a viable candidate for driving this part
of our theoretical adaptive system. As it were, however, it is not immediately clear
whether or not any or all of these algorithms would be appropriate for our
purposes— and if so, in what cases might they be. Though there is much excellent
research done and many intriguing algorithmic papers with impressive results have
been presented, it is also not clear how these different algorithms relate, even though
many are indeed similar in their theoretical driving justifications. Each algorithm
has its own objective function, and though these may be compared and contrasted, it
may not be immediately evident what the implications of these choices of functions
may be in various situations. It is also not clear how a specific algorithm's
performance scales under different data conditions (curvature specifications of the
data curve, error levels in data collection, etc...). In order to evaluate the
appropriateness of these algorithms for our (or any) purpose with some level of
rigour, we would like to present a formal analysis of the algorithms, their conditions,
and their performance in such manners as are possible here.

\textsuperscript{2} Similar to the elementary framework for agent-based AI theory as presented in [Russell].
\textsuperscript{3} As examples: should the complexity of the function being described or learned be outside of the
bounds for simple description by the abstract unitary component of the AI system (as in a nonlinear
function being learned by a linear neural network) or should output behaviour be dependant upon
some unknown (directly unperceived) variable, the existence of which must be extrapolated by the
system.
I.2. Dissertation Overview

First, we would like to consider the general problems that are described by the terms "dimensionality reduction" and "manifold learning" and some of their more formal implications in Chapter 2. We will there continue in such a vein and formally describe the basic linear (MDS and PCA) algorithms and most common non-linear (Isomap and LLE) techniques on order to guide our preliminary choices in the analysis of the field of techniques.

Chapter 3 will outline a number of both theoretical and concrete concerns for applicative senses of the algorithms, concentrating on validity conditions for scenarios, expected behaviours, and pathological situations. In Chapter 4, we will use these outlined concerns in order to conjecture a spread of evaluative criterion for performance models of the algorithms which we will test in Chapter 5.

We will continue by addressing the applicative concerns of Chapter 3 enlightened by our experimental results and offer methods for dealing with some of these concerns as well as offering ideas for algorithmic improvements (both concrete and theoretical) which may improve performance over our metrics and applicability considerations in Chapter 6. Finally, in Chapter 7 we will attempt to offer potentially useful extensions of these algorithms guided by our findings before we conclude this presentation.
Chapter II. Algorithmic Overview

II.1. Dimensionality Reduction and Manifold Learning—
Definitions

The following formal definition of the aim of dimensionality reduction, gives a very concise idea of one conception of the goals of these algorithms:

“Given $N$ high dimensional inputs $X_i \in R^D$ (where $i = 1,2,\ldots,N$), the problem is to compute outputs $Y_i \in R^d$ in one-to-one correspondence with the inputs that provide a faithful embedding in $d < D$ dimensions.” [Weinberger]

A less formal but equally effective description of the technique is a method for rotating (transforming) untenably high-dimensional data to a best low dimensional fit in order to increase manageability and visualisablility. [Labelle]

A combination of the following definition for manifold learning,

“Manifold learning can be viewed as implicitly inverting a generative model for a given set of observations. Let $Y$ be a $d$ dimensional domain contained in a Euclidean space $R^d$. Let $f: Y \rightarrow R^D$ be a smooth embedding for some $D > d$. The goal is to recover $Y$ and $f$ gives $N$ points in $R^D$.” [Raykar]

, and the conception (motivated by theoretical geometry) of the data as a transformed latent manifold ($Y$, in the definition) gives rise to the idea that the goal of these techniques is to extract a simple low-dimensional manifold from an isometric or conformal transformation. [Raykar]
The differences in these conceptions, though slight, give rise to some issues when considering the scope of applicability and practicality of performance metrics. For example, manifolds are not directly comparable to a free functional description of a transformation—some transformations, though valid by the given definition, would change the nature of the manifold itself (a line transformed through a radial rose function results in a multiple torus, a non-equivalent manifold structure) and so may not be a valid transformation to posit under the definition. Also, the relevance of mapability (and re-mapability) between the latent and transformed manifolds is different between the definitions of the problems. Indeed, manifold learning can be considered as a subspace of the general dimensionality reduction problem. [Raykar]

The techniques we will discuss have been applied to widely different applications in widely varying scenarios, and as such, there are innumerable minor variations on implementation of each aspect of the algorithms. We will attempt to encompass these implementational variations as well as the basic applicability of the concerns which arise by consideration of the problem definition. To allow breadth of consideration, we will attempt to use a maximally broad conception of the joint issue posited by the two definitions in order to consider functionality in a general sense without specific constraint to either definition. This should be viable as long as we can avoid destroying the concreteness of our conjectured metric and do not limit the applicability of an algorithm in a given scenario.

Also worthy of note, within the scope of definition and mapability concerns, is the idea that raw data to which these algorithms are applied are generally considered as a point in high-dimensional space (for example, an 8x8 pixel greyscale image is considered a point in a 64-dimensional space) instead of a functional description in a lower dimensional space (the same image could be described as a general function in a 3-dimensional space with the specific image function $f: x,y \rightarrow \alpha$). This may seem a largely semantic issue, but is worth comment as we outline our problem definition.

---

4 Where $\alpha$ is the pixel’s alpha channel value.
II.2. Linear Dimensionality Reduction Techniques

These algorithms, though incompatible for our desired uses since they do not deal well with non-linear data, create the basis for non-linear techniques and are important as subalgorithms so are described in brief here.

II.2.1. MDS

The Multi-Dimensional Scaling (MDS) algorithm is a fairly direct reconstruction of coordinate points from a distance matrix. Weighted best-fit reconstruction can be used when the distance data describes a non-Euclidean structure for the number of dimensions desired. Alternatively, any such non-Euclidean structures could be modeled Euclideanly in a coordinate system of higher dimensionality.5

The following algorithmic description is the basic classical MDS method:

1. Obtain dissimilarities \( \{\delta_{rs}\} \).
2. Find matrix \( A = [-\frac{1}{2} \delta_{rs}] \).
3. Find matrix \( B = [a_{rs} - a_r - a_s + a_.] \).
4. Find the eigenvalues \( \lambda_1, \ldots, \lambda_{p-1} \) and associated eigenvectors \( v_1, \ldots, v_{p-1} \), where the eigenvectors are normalized so that \( v_i^T v_i = \lambda_i \). If \( B \) is not positive semi-definite (some of the eigenvalues are negative), either (i) ignore the negative values and proceed or (ii) add an appropriate constant \( c \) to the dissimilarities, \( \delta'_{rs} = \delta_{rs} + c(i-\delta_{rs}) \ldots \) and return to step 2.
5. Choose an appropriate number of dimensions \( p \). Possibly use \( \sum_i 1/\lambda_i \Sigma(\text{positive eigenvalues}) \) for this.
6. The coordinates of the \( n \) points in the \( p \) dimensional Euclidean space are given by \( x_{ri} = v_{ri} (r = 1, \ldots, n; i = 1, \ldots, p) \).” [Cox]

where the dissimilarities matrix is equivalent to the distance matrix between points which is used to create matrix \( A \), which is “doubly-centered” to create \( B \).6 The desired number of most-contributing dimensionalities then arise as the highest \( \lambda \)-

---
5 Justification via Nash embedding theorems. [Nash 1,2]
6 Alternately, \( B \) can be considered derived via the equation “\( B = HAH \) where \( H \) is the centering matrix \( H = I - n^11^T \), with \( 1 = (1,1,\ldots,1)^T \), a vector of ones.” [Cox]
valued eigenvectors of $B$ and are selectable according to absolute or value-dependant criterion as is appropriate for the situation.

II.2.2. PCA

The Principal Component Analysis (PCA) algorithm uses variance measures to extract the most relevant dimensional structures in a data set distribution. Thereby, all of the dimensions with a certain relevance level (variance measure) can be extracted, or a set number of most relevant dimensions can be delineated. Some non-linear dimensionality techniques use PCA as a final step in their algorithms, after processing the data to approximate linear relations from the non-linear ones present in the data.

A straightforward description of the algorithmic process follows:

“In practice, the algorithm proceeds by first computing the mean of the vectors $x^n$ and then subtracting off this mean. Then the covariance matrix is calculated and its eigenvectors and eigenvalues are found. The eigenvectors corresponding to the $M$ largest eigenvalues are retained and the input vectors $x^n$ are projected onto the eigenvectors to give the components of the transformed vectors $z^n$ in the $M$-dimensional space.” [Bishop]

where $x^n$ is the set of high-dimensional coordinate data and $z^n$ is the $M$-dimensional (low-dimensional) projection of the $n$ points in the data set. The data-relation analysis inherent to the covariance matrix’ eigenvalues allows the “intrinsic” dimensionality of the data distribution to be estimated by comparing the resultant eigenvalues and working under the assumption that particularly small values for such correlate to dimensionalities which can be removed with minimal loss to the data’s coherence and descriptive ability. [Bishop]
II.3. Non-Linear Dimensionality Reduction Techniques

These two non-linear techniques will create the basis for our algorithmic investigation. We hope to be able to analyse these algorithms, compare them, critique them, create a metric or set of metrics for gauging their performance, and, if possible, extend and/or enhance them.

II.3.1. Isomap

The Isomap algorithm uses a global theory for constraining non-linear data to a linear manifold, as opposed to the local theory used by the LLE algorithm. A distance matrix is acquired from shortest-path distances between non-local points, using k-nearest-neighbor or epsilon-sized neighborhoods for acquiring the local shortest-distance information. PCA is then used to place the new linearly-constrained data points in the appropriate number of dimensions.

The Isomap algorithm, more formally, is as follows:

“1. Construct neighborhood graph: Define the graph $G$ over all data points by connecting points $i$ and $j$ if [as measured by $d_v(i,j)$] they are closer than $\varepsilon$ (\(\varepsilon\)-Isomap), or if $i$ is one of the $K$ nearest neighbors of $j$ (\(K\)-Isomap). Set edge lengths equal to $d_v(i,j)$.

2. Compute shortest paths: Initialize $d_g(i,j) = d_v(i,j)$ if $i, j$ are linked by an edge; $d_g(i,j) = \infty$ otherwise. Then for each value of $k = 1, 2, \ldots, N$ in turn, replace all entries $d_g(i,j)$ by $\min\{d_g(i,j), d_g(i,k) - d_g(k,j)\}$. The matrix of final values $D_g = \{d_g(i,j)\}$ will contain the shortest path distances between all pairs of points in $G$.

3. Construct $d$-dimensional embedding: Let $\lambda_p$ be the $p$-th eigenvalue (in decreasing order) of the matrix $\tau(D_g)$, and $v_p^i$ be the $i$-th component of the $p$-th eigenvector. Then set the $p$-th component of the $d$-dimensional coordinate vector $y_i$ equal to $\sqrt{\lambda_p}v_p^i$.” [Tenenbaum]

This global view of linear remapping allows for a retention of the full amount of data available during the reconstruction, however, this may cause problems not only with execution time for extremely large data sets, since shortest paths are computed exhaustively, but may restrict its flexibility in pathological
situation. As an example, the semi-myopic LLE algorithm can deal with conformal mapping manifold reconstructions to some degree without modification whereas Isomap (designed, obviously, to deal with isometry) requires a major algorithmic modification, C-Isomap, to deal with such situations. [de Silva] The only free parameter in the algorithm is the neighborhood size.

II.3.2. LLE

The LLE (Local Linear Embedding) uses a local theory, capitalising on the concept that a non-linear manifold is linear at infinitesimal distance measures (and thereby theoretically sufficiently near-linear for small distance measures). Each point’s local distance measures are obtained by constraining them to the desired dimensionality with a linear weighted reconstruction. An MDS-style method is then used to place the modified distances within a desired dimensional structure.

The following figures describe the LLE algorithm in an effective and concise manner:
LLE ALGORITHM

1. Compute the neighbors of each data point, $X_i$.

2. Compute the weights $W_{ij}$ that best reconstruct each data point $X_i$ from its neighbors, minimizing the cost in eq. (1) by constrained linear fits.

3. Compute the vectors $Y_i$ best reconstructed by the weights $W_{ij}$, minimizing the quadratic form in eq. (2) by its bottom nonzero eigenvectors.

---

from Saul and Roweis, “Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds”\textsuperscript{7,8}

Figure 1: LLE Description 1

![Diagram of LLE algorithm]

A data point $X_i$, its neighbors $X_j$, and its locally linear reconstruction $\sum_j W_{ij} X_j$. The reconstruction weights are constrained to satisfy $\sum_j W_{ij} = 1$.

from Saul and Roweis, “Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds”

Figure 2: LLE Description 2

This technique of local linear construction has great execution resource benefits since a sparse distance matrix can be used for calculations (only points within the neighborhood value have any bearing on any particular data point’s transformation) and is not troubled by odd global phenomenon, but some global data

\textsuperscript{7} \text{eq}(1): L(W) = \sum_i |X_i - \sum_j W_{ij} X_j|^2$. [Saul]

\textsuperscript{8} \text{eq}(2): \Phi(Y) = \sum_i |Y_i - \sum_j W_{ij} Y_j|^2$. [Saul]
may indeed be lost since the algorithm is dependant upon a memoryless and myopic structure view to piece together each local reconstruction. The algorithm, like Isomap, only has a single free parameter, the neighborhood size and “No learning rates or annealing schedules are required during the optimization, and no random initializations of local optima affect the final result.” [Saul]
Chapter III. Scenario Analysis

In this section, we investigate theoretical boundaries of applicability and potentially problematic situations for dimensionality reduction techniques. We will present these in a categorised and itemised list for later reference.

III.1. General Application Concerns

The following itemised concerns deal with the general overarching scenario for which one of our algorithms might be considered.

_a. Elements of Relations Unknown_

The low-dimensional (latent) manifold properties may well be unknown in an arbitrary applicative sense—whether it be the ordering, orientation, or even dimensionality of the latent manifold—though some tools (such as the PCA covariance measure for the intrinsic latent dimensionality concern) can give good practical approximations (within boundaries of applicable usefulness, relevant resolution, etc.) of this information. Inasmuch as these properties are specifically unknown for most applications, a result’s accuracy and correctness are difficult to measure as there is no available data for “correctness” checking, and even “sanity” checks, warnings of absurd results, may be unreliable given the possibility of some unexpected pathological situation on which the algorithm is to be run.
b. Uniformity of Latent Manifold Distribution

Of great importance when constructing an instantiation of the algorithms’
application is that uniformity distribution is inherent in the data passed to the
algorithm, i.e.: the input function data for the algorithm $A$, $A: g(f(x_m)) \rightarrow d(x_n)$,
where $f$ is the theoretical high-dimensional transformed manifold, $g$ is the sampling
function from the pure distribution function, and $d$ is the latent manifold
approximation. As such, an $f'$ with an uneven sampling uniformity is
indistinguishable from an $f''$ which is uniformly sampled but has a different center of
density or curve complexity— as always, an algorithm is only as useful as that data
input, but for theoretical correctness concerns, this potentially subtle consideration is
important to remember.

c. Manifold Transformations

The difference between isometric and conformal manifold mappings is also
important since the general assumptions governing each transformation are different.
As such, some algorithmic modifications may be necessary to deal appropriately
with the different types of manifold extraction (e.g.: Isomap vs. C-Isomap). Also,
non-Euclidean transformations may be inherent to the data and, indeed, it may not
even be clear what a correct answer for a specific transformation might be: should a
sphere be reduced conformally to a plane (perhaps in polar-style coordinates) and if
so, how should the unraveled seams of the conformal mapping, which will directly
and selectively disrupt the distance measure retention criterion of the problem
definition across its induced discontinuity, be decided; or should it be considered a
single-dimensional structure at its core since it is describable as a single variable in
an appropriate coordinate system?
d. Resource Consumption

A complexity analysis\textsuperscript{9} gives an approximate order of complexity of $O(CN^2)$ for the LLE algorithm and $O(CN^N)$ for the Isomap algorithm (where C is a constant functionally dependant upon the dimensionality of the sampled data, the dimensionality of the output data, and the neighborhood size that is chosen). This is reflected in the difference in the reconstruction perspectives of each algorithm (local versus global) and provides reason to believe that there will be a direct trade between global accuracy and execution time due to this perspectival discrepancy.

\textit{e. Neighborhood Size}

Each of our core non-linear algorithms requires, as its single free parameter, a neighborhood size upon which to localise preliminary distance calculations. The choice of this parameter may not only potentially cause the algorithm to fail (as in the case of a pathologically small or zero-size choice for the parameter), but the ideal choice for this parameter may not be obvious as execution of the algorithm commences.

III.2. Input Concerns

The following category of concerns deal with potential issues with an instantiated input sample for our algorithms.

\textit{f. Sampling Error (Noise)}

Of particular concern is the ability of these algorithms to deal with varying amounts of noise (traditionally, sampling/instrumental error) and minimise the noise’s effect on the reconstruction (this concept is bound inextricably with the

\textsuperscript{9} Similar to and upheld by the one presented in [Huo].
potential problematic issue with PCA’s variance measures—a certain variance for a relevant dimension may be indistinguishable from error present).

g. Error Contaminating Curve Dynamics

Another issue with the presence of noise is the possibility that the amount of noise present will interfere with the distinguishability of the manifold curve dynamics.

h. Curve Sampling Density

Another relevant sampling issue is the choice of sampling density—a poor choice of which may limit the ability to describe a curve in a sufficiently locally linear fashion. Since our core algorithms deal specifically with extrapolating local linearities form neighboring points, the sampling density must be higher in sections with higher curvature in the transformed manifold in order to be able to extract local linearities with equivalent accuracy, especially under the presence of noise. [Bengio] This may seem at odds with the uniformity of distribution concern \((b)\), but as there is a difference in sampling and distribution uniformity, we can meet both criterion (if possible, which may be difficult given the unknown latent data concern, \((a)\)) as long as the distributions of both sample and distribution are tracked and considered appropriately (i.e.: concern \((h)\) gives reason for non-uniform sampling and concern \((b)\) imposes requires for a mapping of that sampling to a uniform distribution while restricting inappropriate sampling-space geometries and curvatures of the dimensional structure within the high-dimensional space).

Of potential concern is also how multiple samplings of each point may effect our algorithms, but this issue should not be problematic in either a data design situation (multiple instantiations can be averaged to create a mean point) or a direct execution situation (even in an errorless situation, distanceless differentiation between points should be retainable—as long as some record of equivalence between the meaning of these points’ reconstruction is kept, a necessary detail for most conceivable useful implementations, the reconstruction will be meaningful
whether for visualisation or if a mean location for the point is created after the algorithm’s execution.\textsuperscript{10}

\textit{i. Outliers and Edge Effects}

Outlying data points, whether error, sample, or transformation induced, may cause problems with a reconstruction, unfairly skewing the results of the algorithm or contaminating measurements based upon averages. This is particularly volatile when the neighborhood choice is static (outlying points may end up with invalid neighborhood values), but even when not critically problematic, the effects of seemingly innocuous outliers like sample domain edges are sure to affect the output of the algorithms since the (mean) relationship dynamic between neighboring points is specifically different for these edge-points or outliers (than for the remainder of the sampling distribution).

\textbf{III.3. Output Concerns}

The following category of concerns deal with potential issues with an instantiated output sample which is returned by our algorithms.

\textit{j. Remapability of Output}

Within the realms of concern regarding the output of these algorithms is included the mapability of the transformation and whether or not the algorithm reveals some concept of remapability (i.e.: some non-naive conceptual mapping relationship between the latent and high-dimensional manifold). For some applications (i.e.: visualisation), this is irrelevant, but for others (i.e.: data compaction), this is critical. This issue is of greater relevance specifically to the manifold learning definition that the more general dimensionality reduction problem.

\textsuperscript{10} Although this will affect the necessary or best choice of neighborhood size— reference concern (e).
k. Dimensional Orientation

Also of concern is the loss of dimensions’ relationships to each other (compared to the ordering and relationships of the dimensions in the input structure) in the low-dimensional result of the algorithms. Since our core algorithms use the linear techniques for their final embedding process, both of which necessarily discard dimensional structure to re-embed detected dimensions by relevance, the dimensional orientation of the input will be lost and recreating this relationship, if desirable may be difficult, if possible or relevant.

l. Transformed Output

The orientation issue is native to all of the core dimensionality reduction techniques due to the use of the linear techniques in the embedding phase: since coordinate data is generally discarded for distance data, no orientation information is generally retained in the output. Re-orienting this data, if desirable, may not be a straightforward procedure.

m. Outliers

Problematically-sampled points or those with an unusually large amount of sampling error may not fall within a seemingly appropriate range for the output coordinates, potentially skewing any error or scaling measure quite badly. It is possible to simply omit outliers according to some standard statistical procedure for doing such (via comparison to standard deviation of the remainder of the data, for example), but, except in straightforward visualisability applications, this may not be appropriate since the outlying point may need to be associated with some output coordinate. Simply omitting outliers may non-uniformly skew any attempted error measure as well.
III.4. Pathological Situations

There are certain potential sets of data which may be specifically problematic for our algorithms. We investigate a few major classes of these pathological situations here, even though these may disrupt the meaningfulness of certain aspects of either the dimensionality reduction or manifold learning problems’ definitions.\textsuperscript{11}

\emph{n. Curve Dynamic Nearness}

Parts of the high-dimensional manifold’s curves may be very close so as to cause interference with each other, even in the absence of contaminating noise, as in a knot or coil geometric structure.

\emph{o. Non-Differentiable Transformation Functions}

Piecewise or otherwise non-smooth/non-differentiable functions may well pose some problem for the algorithms (these classes of algorithms are often referred to as specifically designed for use with smooth— or a subclass thereof— transformation functions [Raykar, Weinberger]). To test the robustness of these algorithms under arbitrary transformation functions, we would like to investigate the disruption that such a differentiability dis-continuity might cause.

\emph{p. Non-Continuous Transformation Functions}

When considering arbitrary transformation functions, classes of pathologically disruptive transformations should be noted—we have little hope that the algorithms in the core of our investigation would be able to cope with a (or a series of) discontinuities (such as in a tangent function graph) since they are specifically grounded in the assumption that points on the latent manifold will be

\textsuperscript{11} It may not be fair to judge the algorithms we present for analysis in this dissertation against these general functional, rather than isometric or conformal, transformation since the algorithms were designed for these latter cases. However, as a potential raw transformation type and relevant to our guiding concern, we wish to at least mention their existence.
spaced in some directly relative fashion with the transformed manifold. However, since extraction of a manifold or mapping function from an arbitrary transformation is desirable from our (admittedly quite broad) investigatory perspective, we would like to note such here.

$q$. Non-Onto Functions

Similar to non-continuous transformation, we do not expect any of our core algorithms to be able to deal with a transformation where, for example, a single point or area in the sampled data is expected to lie on several different sections of the latent manifold (such as in a rose or other radial function), but, again, would like to note the transformation type for completeness’ sake.

Figure 3: Illustrations for Pathological Transformation Functions
Chapter IV. Gauging Performance

Though each algorithm has its own objective function by which it would be best served for evaluation, we wish an objective measure of performance which may be used to compare performance of different algorithms under different conditions. Presented here are a few straightforward error measures and their properties relative to the algorithmic question at hand.

It should be noted that these methods for evaluating error specifically require knowledge of the true original latent manifold, something generally not present in useful applications of the algorithms. However, we will create test cases for our experiments so as to control several factors along which to investigate the algorithms’ performance and we will have such comparative data in order to glean some error measures, not to measure a level of performance in a specific case, but rather to compare performance between cases and algorithms (and their modifications). These error metrics will not be concerned with the absolute values of their functions, but with comparative differences between case-based valuations.

IV.1. Mean Squared Error (MSE)

A per-point distance error is a directly simple and obvious option for our error measure. This measure may be problematic due to the algorithms’ general lack of orientation preservability, however, this aspect may be useful if we wish the error value to preserve this mis-orientation issue. Inherent to the output of our algorithms is not only a rotational and dimensional orientation issue, but a scaling issue of the re-embedded data layout. We therefore have several options with the MSE measure even only as far as dimensional orientation and scaling— we can use unscaled,
naively scaled (applying a mean of scaling error compared against the original location of points), and proportionately scaled (in order to minimize the magnification of effect scaling weights from points equally displaced objectively—points very close to the distribution’s center of mass, which will be centered to the origin by the linear embedding subalgorithms, might have an undesirably magnified effect on points further away from the output distribution’s center of mass when applying the contribution of the scaling factor directly). We can also chose whether or not to permute the dimensional orientation or use some method inherent to our selection of data distribution(s) to attempt to force an alignment of the extracted dimensions as we see fit. Each of these MSE methods may have uses depending upon the underlying assumptions of the application at hand as well as the data being used—the choice will depend largely on which aspects of the reconstruction wish to be judged.

IV.2. Normalised MSE (nMSE)

In order to deal with the problems of orientation and scaling, we can use a normalised version of the MSE measure to judge closeness of reconstruction invariant to error levels which are likely to scale upwards as points are further away from the reconstruction’s center of mass (due to the larger relative distances a scaling or rotation effect will cumulate when operating upon larger numeric coordinate values).

nMSE measures scaling error against the coordinate value of a point do not make much sense here as our coordinate system does not reflect a comparative valuation of any sort, merely an orientation. However, using a nMSE measure of the MSE ratio to the points’ variance across the range of trials may help dampen the effect of continuous error inflation due to points further away from the origin being rotated or rescaled upon reconstruction to a larger degree on the coordinate scale.
IV.3. Auto-Transformed MSE

In an alternate attempt to improve upon the MSE method, we would like to attempt to realign the configuration of points in the output re-embedding as close as possible with the original alignment of the latent manifold. In order to do this, we attempt a radial transformation of the output data as a whole over a mean level of difference between each point’s re-embedded and original orientation compared to the center of mass of the manifold’s data distribution. In order to avoid problems that may arise in this method from symmetries or other pathological situations, we can try several methods of applying this auto-transformation: to each of the variants of non-rotated MSE mentioned above (section IV.1), for example.

Like the nMSE method, this auto-transformation attempts to remove inflated error measures due to scaling and rotational transformations of the theoretical latent manifold which are native to the algorithms we shall use, realigning the output with the known orientation. Unlike nMSE, the reduction in error depends upon the level of success of the auto-transformation procedure. There are, of course, alternate methods for reconstructing the appropriate alignment configuration of the output data points than the mean angle difference method we put forth above, but many of these (such as those presented in the field of Shape Analysis) deal with more complex levels of transformations than we require here. We may, however, select a different transformation procedure taking cues from such fields (for example, using landmarks to test the best transformation as in a landmark-style Procrustes Matching algorithm, randomly sampled to avoid structure-generated symmetries which may befuddle the rotational procedure). Alternatively, though we do not do so here, and merely emulate such with a (minimum) best-figure selection from a range of different styles of these scaling and rotation methods, glean a best-fit figure for the scaling and rotation multipliers through a complex and time-consuming, but useful for minimising the error figures, linear programming script.

These best-fit (or emulated best-fit) transformations of the output are good for attempting to describe an error level consistent with categorisation or visualisation, where orientation issues are much less relevant and are therefore minimised as much as possible.
IV.4. Data Coherence Measure (DCM)

As an alternate version of an orientationless metric, we use the visualisation-motivated, local relation coherence measure which measures the relative distances within a neighborhood against the true latent distances in that neighborhood. This method bypasses the rotational issue altogether, although scaling may be an issue unless a further variant on the data coherence such as comparing ratios of neighborhood distances instead of their absolute values to the true latent distance data. We consider it wise to take a larger neighborhood sample for the data coherence error measure than is taken for the algorithm’s neighborhood size in order to glean information about how the data coheres on a larger scale than the local reconstruction’s objective functions would strictly dictate, but only slightly larger so as to keep the measure constrained to local behaviour.
Chapter V. Core Experiments

For our core experiments, we used Matlab (v.6) to run the minimally modified algorithms presented and offered by the core algorithms’ designers.

V.1. Experiment Outline

In the following set of experiments, designed to test both the metrics we have outlined and the robustness of the different core algorithms under different noise conditions, we use a latent 2-dimensional manifold sampled evenly over the range [-1,1] for both x and y coordinate axes. To strike a balance between avoiding low numbers of data points (usefulness) and keeping the experiments streamlined so that we may run a sufficient number of trials for each experimental variable (practicality), we use approximately 1000 data points for each manifold test.

For our transformed (high-dimensional) manifolds, we use the following functions: a straight-forward polynomial transformation of the x-axis into the new z-axis \( z = x^2 \), an absolute value transformation of the x-axis into the z-axis \( z = |x| \) to test mild discontinuities in differentiability and provide a modified foil to the polynomial transformation, and the “swiss roll” example used in many of the seminal papers introducing the core non-linear algorithms (we will use \( x = r \sin \theta \); \( z \)

\[ 12 \text{ We elected to use a rectangular sample in stead of a circular sample of the theoretical latent and transformed manifolds— in spite of the possibility of conflating our error measurement due to the edge effects of the irregular geometry— to avoid confusingly visualisable results and to include the effects of those same edge-effects as potentially important aspects of a sample which might not be so conveniently resampled. Since the error measures are to be considered relatively, this should not introduce any problems with extrapolable experimental results. As a useful side-effect of the combination of our method of latent manifold transformation and choice of rectangular sample, the expected variance measure for each dimension of the algorithm output is predictable (except in high sampling error situations) and usable for re-orientation and error metric computation.} 

\[ 13 \text{ 1089 data points (a 33x33 grid), to be exact.} \]
= r\cos\theta \mid r = \theta(c). \) In order to compare the algorithmic results to the evenly spaced latent manifold grid, we will need to maintain equivalent distances between points rather than a direct transformation of the coordinates in the latent manifold grid—we will use the integral calculus equations for arclength\(^{14}\) to acquire equidistant sampling across the transformed manifold.

To test robustness under error conditions, we will use errorless control examples as well as increasing Gaussian noise distributed randomly over all dimensions of the transformed manifold. In order to test a maximal amount of noise with a minimal amount of interference to the grid organisation, we will test an amount of noise based on the standard Gaussian distribution where 99% of the distorted points will lie within half of the distance between the closest neighboring point(s).\(^{15}\) We will use this error level as a basis (error level 0) and create a set of test error levels with exponentially increasing and decreasing standard deviation values from this basis value.\(^{16}\) We extend the error levels as integers from -2 to 5, selecting this range for a minimal but noticeable error on the low boundary and for a point at which the structure of the original noiseless data becomes almost inextricable (visually) from the error-introduced data on the high boundary.

---

\(^{14}\) \(s = \int_a^b \sqrt{1 + |f'(x)|^2} \, dx.\)

\(^{15}\) Standard deviation, \(\sigma = .01213121118\) for our experiment’s data grid.

\(^{16}\) \(\sigma = .01213121118 \cdot 2^\alpha\), where \(\alpha\) is the named error level integer.
For all data points in our experimental results, we use 100 trials and take the mean performance over those trials in order to acquire an idea of expected performance under the given conditions.

To acquire figures for error measurements, we use the following base transformational functions, combined and permuted where possible as described in Chapter IV:
Scaling—applied by dimension to each point\textsuperscript{17}
\[
\begin{align*}
c_x &= \frac{\sum (v_i/v'_i)}{n} \\
c_y &= \frac{\sum (v_j/v'_j)}{n}
\end{align*}
\]

Proportional Scaling—applied by dimension to each point
\[
\begin{align*}
c_x &= \frac{\left(\sum [(v_i/v'_i)\cdot|v_i|] / n\right)}{\left(\sum |v_i| / n\right)} \\
c_y &= \frac{\left(\sum [(v_j/v'_j)\cdot|v_j|] / n\right)}{\left(\sum |v_j| / n\right)}
\end{align*}
\]

Rotational Adjustment—applied to each point\textsuperscript{18}
\[
c_\theta = \frac{\sum (v_0-v'_0)}{n}
\]

Figure 5: Auto-Transformation Functions Used in Core Experiments

We used a neighborhood of 24, selected as a doubling in neighborhood radius for the k-neighborhood value, used (as opposed to a doubling of the number of neighbors) in order to maintain symmetry for the grid-type structure, to calculate values for our data coherence error measurement. For each trial, all error calculations were averaged over the error calculation for each point to give an error figure for that error metric over that trial.

V.2. Neighborhood Size

In order to minimise the experimental variables for our trial sets, we would like to use an appropriate constant neighborhood value for the algorithms. In order to do this, we take a cue from the seminal paper on the charting non-linear dimensionality reduction algorithm, assuming that our best choice of neighborhood will be at the point where the neighborhood is most linear. [Brand] At low

\textsuperscript{17}c_x,c_y as scaling factors for x and y axes respectively, n as number of data points, subscripts i and j used to respectively denote x and y coordinates for the referenced point, and v and v' used to respectively denote the point enumerated by the summation function for the original latent and output re-embedded version of the data set.

\textsuperscript{18}c_\theta as converted polar coordinate rotation angle for data set, subscript 0 used to denote converted polar coordinate angle of the referenced point.
neighborhood sizes, error will cause the neighborhood of points to seem non-linear, and at high neighborhood sizes, the curve of the manifold will again make the set seem non-linear. The charting algorithm paper suggests that, as neighborhood size increases, there should be a size at which the number of points in the neighborhood grows most slowly, this taken to be the most-linear neighborhood descriptor of the point in question. The charting algorithm, however, uses this technique on a per-point basis, and the algorithms we are presently testing use a global valuation for neighborhood size. We compare neighborhood sizes across different error-levels here to see if the charting paper theory will hold with our knowledge of the linearity of the manifolds we have created and it does seem that a neighborhood value of \( k = 8 \) (\( \varepsilon = 0.12 \) in our 33x33 grid) is a valid choice with enough points to overcome all but pathological amounts of noise while remaining locally relatively linear across all of our error levels.

![Figure 6: Relation between k and \( \varepsilon \) for Increasing Error Levels](image)

The plateau at \( k = 8 \) for the near-errorless case is, from the graph relationships shown here, the largest span of increasing neighborhood radius with minimal activity that retains a low rate of increase for higher error levels as well.

---

19 \( k \) being the value for \( k \) nearest neighbors and \( \varepsilon \) being the radius value for an epsilon-ball, the nearest neighbors being all those points which fall within the boundaries of this structure.

20 The error levels increase polynomially (factor of 2) as the curves grow more smooth and shallow. The most distinct (top-most) curve is a near-errorless control trial. All curves are mean values for the rate of increase over 100 trials.
Since these are averages of per-point values and sets of trials, edge effect as well as the smoothness of the mean curve detract from the usefulness of this value—the charting algorithm’s conjecture is based on the idea that neighborhood sizes will be decided for each locality. [Brand] Even so, the value we derive is in line with our instincts (the grid’s most linear $k$ size would be 4, but a larger gap lies outside the $k$ value of 8 before more points are added to the neighborhood and by choosing an only slightly larger neighborhood size than the theoretically most linear, we mollify increasing error effects while retaining a maximal linearity) and with best-result precedent offered in other published experiments. [Saul] All of the tested transformations gave extremely similar results on the low-order neighborhood radii shown here—though we will not be able to declare a perfect neighborhood size for the experiments we are running without algorithmic modification, we can use our findings here as a best general approximation of an optimal selection for the variable.

It might be argued that since the appropriate neighborhood radius size is dependant upon a ratio of noise to density, a change in sampling density would render our results void. Though this is definitely true, the change would be scaled to the inversely dependant density rating and since we are using constant density throughout the experiment set in order to facilitate performance metrics, we can ignore the issue for the value-scaling purposes as long as we are consistent and could emulate any difference due to a change in the function dynamics with a differently sloped manifold transformation.

It should also be noted at this point that a uniform choice of epsilon for the epsilon-neighborhood value, if consistent, will decrease the actual number of neighbors as the error level increases. This may (and does) cause problems on the higher error levels, but we would rather use the constant value even if it does corrupt data for some trials in order to maintain consistency with the established algorithms’ framework and avoid introducing other axes of variation where possible so as to avoid corrupting the overall comparative data. It is conceivable that some scaled epsilon value could be used that would maintain a certain neighborhood size, but this is overlapped by the purpose of the $k$-sized neighborhood option and would be a useless distinction here.
V.3. Expected Error Values

In order to see to what degree each algorithm adds error to the reconstruction as opposed to an idealised reconstruction, we will wish to compare the error values of the algorithm’s reconstruction to the error in a perfect reconstruction (possible only when the latent manifold is known). This is meaningless, of course, in an errorless case, but gives a good measure of how much faster error in the reconstruction grows than error in the sampling procedure would dictate (in an ideal reconstruction).

When comparing error level trials against idealised performance, the same noise distribution is used upon the latent manifold as if a naive linear dimensionality reduction was used, ignoring any extension of the error into a z-axis and measuring this error as per the method for which comparison is requested. For simplicity’s sake and to avoid corrupting the data in pathological cases or corrupting our expected error data (and since the introduction of error is relatively small as is the potential benefit from modifying this modified latent manifold), we avoid any scaling or rotational transformation of the data distribution for the expected error valuation, assuming that, on average, a normal-style distribution should retain the same general volume and orientation.

V.4. Core Experiment Results

We present the following results of our experiments:

We begin with a close look at our different selections of error measures for algorithm performance— using the k-LLE algorithm as an arbitrary basis scenario, we obtain the following MSE data when the 2-dimensional algorithmic output is measured directly and when it is scaled to fit the appropriate volume of the original (pre-transformation) latent manifold:
Figure 7: Unscaled and Scaled MSE Measures for k-LLE over $z = x^2$ Transformation
Of important note here is that even though the scaling helps a great deal at low error levels, the scaling transformation magnified the error measure after a certain point (error level 0, in our graphs). After investigating singular trials, it seems that the scaling algorithm that we used does not work well when either the data points are rotated sufficiently from the original aligning axes or when the dimensional orientation of the output is reversed from what is expected.

From this point a quick investigation of the normalised MSE measure reveals the following data:

![Graph showing Unscaled Normalised MSE (mean per point) against Error Level Exponent (nil = errorless trial)]
Figure 8: nMSE Measures for Unscaled, Scaled, and Proportionately Scaled Data for k-LLE over $z = x^3$ Transformation
From this data, rendered practically unusable by the massive difference in variance between error levels, we decided to continue to pursue adjusted MSE calculations instead of nMSE figures.

In an attempt to fix the scaling problems with the previous MSE data, we apply the MSE calculations to automatically rotated data (in order to fix alignment problems), use a alternate proportionately-based scaling algorithm, as well as attempt flipping the dimensional axes in order to deal with noise on occasion disrupting the variance-based ordering of dimensions that we expect from the algorithm’s output. We also apply our “data coherence measure” to obtain the following graphs:
[figure continued overleaf]
Figure 9: Alternate MSE Measures and Data Coherence Measure Graphs for k-LLE over $z = x^2$ Transformation
None of the different MSE measures is uniformly most fit at all error levels—each method seems to have difficulties with certain problematic situations, mainly (as case-investigation shows) due to orientation of outliers.

Figure 10: MSE Methods for k-LLE over $z = x^2$ Transformation

Taking the minimum value from the comparison of these graphs, we attempt to obtain an emulated best-fit of the algorithmic output to the latent manifold and we obtain two useful comparative data spectra, shown here in comparison to expected error values for each error level trial:
Figure 11: Emulated Best-Fit MSE for k-LLE over $z = x^2$ Transformation

Figure 12: Data Coherence Measure for k-LLE over $z = x^2$ Transformation
Using these two error measures, we then compare each algorithm’s performance on each test manifold transformation as well as investigating the comparative performance between algorithms on each of these transformation’s data sets:
\[ z = x^2 \]
Figure 13: Comparative Algorithmic Performance on Different Data Set Transformations
The graphs depict the emulated best-fit MSE (mean per point) and data coherence measure (mean per point) against the error level exponent. The graphs show different lines representing various metrics such as $\chi^2$, $|x|$, and expected error, as well as a swiss roll. The x-axis represents the error level exponent with 'nil = errorless trial' and the y-axis represents the emulated best-fit MSE or data coherence measure. The graphs highlight the performance of $e$-Isomap in relation to error level and coherence.
Figure 14: Comparisons of Algorithmic Performance by Algorithm
V.5. Extrapolations from Core Experiments

The metrics, each with their own niche and purpose, display relatively expected levels and increase in error, though the artifacts produced by orientation problems easily befuddle the use of more naive metrics as error levels increase and expectations of dimensional orientation are found to be invalid. Our local coherence measure, however, does an excellent job of measuring expected behaviour of the algorithms and is in line with visualisability results in many of the papers previously written on the subject.

As expected, the LLE algorithm did not perform quite as well overall as the global-view Isomap algorithm, but the savings in execution time was extremely noticeable (2, approaching 3, orders of magnitude for our test suite—within the realm expected by our complexity analysis and the data set size). Both algorithms’ epsilon-neighborhood versions seem to give a minor improvement on low levels of error, but have problems with error levels beyond a certain amount due to the tendency for outlying points to have an insufficient number of neighboring points within the proscribed distance to allow a valid or useful embedding (this is shown by the wildly conflated error levels for the e-Isomap algorithms and the complete failure of the e-LLE algorithm at those error levels)\textsuperscript{21}. This is unfortunate since we would expect gains from the epsilon-neighborhood variant of the algorithms to be greatest when high levels of error introduced outliers and other distortions for which a static neighborhood size by membership numbers would admit potentially distortion-inducing comparisons between “neighbors” which are of greater coordinate difference than would be desirable for the local linear calculations.

The different transformations that we used also provide some general uniform information about the performance of the algorithms— for all of the algorithms, the non-differentiable absolute value function did not cause any greater

\textsuperscript{21} The Isomap algorithm as provided by its designers includes a culling procedure for outliers to prevent the error conflation we see here, but given our driving perspective behind this investigation, we felt that an imbedding was necessary for each data point and that error levels should reflect this fact. Implementations of these algorithms specifically for visualisation purposes would not be negatively affected by this culling whereas a categorisational implementation would find this culling problematic.
disruption to the algorithms as the smooth polynomial curve. For all algorithms as well, the “swiss roll” transformation seemed to increase in error more quickly as error levels rose, due, apparently, to the difference in proximities for different localities of the latent manifold.

A comparison of the local coherence error graphs to the ideal algorithmic response curves does a good job of showing at what point error causes the results of the algorithm to be no longer visualisable—the highest error level we use borders on irrevocably damaged by noise and when the error value of the local coherence measure rises above this point, we can assume that visualisability/recognisability is not attainable (on average), giving us a theoretical maximal error level for the algorithm. This algorithm-dependant ratio of data density to noise (assuming our experiments’ uniform data distribution over both the transformed and comparator latent manifold) allows, in general, an almost maximally disruptive amount of noise for the Isomap algorithms (data coherence measures are just above the levels of expected error) and the LLE algorithms perform reach this maximal level of error functionality at a error magnitude’s standard deviation value of approximately $\frac{1}{4}$ that of the maximally disruptive amount of noise (for a modified standard Gaussian noise distribution).\footnote{We use the term “maximally disruptive amount of noise” here to refer to the point at which the level of error introduction completely obfuscates the underlying pattern of the data as far as could be discerned by a human subject. It would be more rigorous, perhaps, to base this term on a relationship between statistical noise variance across the dimensions and the indistinguishability of a noise variance axis from the data variance axis, but not only is it not clear to what resolution this comparison should be made, but we would like to avoid defining such boundary terms codified in the same manner as the experimental dynamics. We think this use of the term is sufficient for our purposes here and the experiments allow for a distinctive pattern in the data which can be extended without difficulty should the wish to do so arise for a modified consideration for what is a pathologically destructive amount of noise for a system.}
Chapter VI. Algorithmic Modifications

Several modifications to these core algorithms have been conjectured in different papers and are more than noteworthy. The use of landmark points for either core algorithm to minimise execution time and complexity (at the hopefully minimal cost of accuracy) or to allow for algorithmic result re-orientation, is a useful option for large data sets. [de Silva] The idea already introduced to us by the charting algorithm [Brand] of non-global neighborhood size selection would help avoid outliers in the data set as well as maximising the effectiveness at each point of an appropriate neighborhood size choice (at the obvious increase in cost of execution resources) and removing the single free parameter from the two algorithms. Alternate methods of removing outliers are also viable, and under certain noise conditions, could be very useful, but this could also be destructive if not used with caution and could disrupt the relationships between points as well as potentially destroying valid data or failing to plot appropriately the latent spectrum of data points.23 The use of conformal algorithms such as the C-Isomap variant [de Silva] may also be useful, but use of such must be implemented with caution since it may not be immediately clear in which way a conformal mapping should be “unfolded.”24, 25

Here, we would like to revisit our concerns outlined in Chapter III and address them, where appropriate, in an attempt to postulate extensions of the core algorithms which may increase their robustness in these situations of note:

---

23 See footnote 21.
24 See sub-section III.1.(c).
25 For this reason and the implications it has on obfuscating what might be a good manner to extrapolate a useful, generic, and sensible performance metric, we shy away from conformal transformations in our presented experiments.
a. Elements of Relations Unknown; b. Uniformity of Latent Manifold Distribution; c. Manifold Transformations

Here we have the classical implementation conundrum of judging the success of a process without any information to make such a comparison. In a supervised capacity, this is not particularly problematic— a selection by the supervisor can be made from several disparate implementations or a single implementation’s output can be denied as erroneous. For an unsupervised process, however, the unsureness may range from which algorithm is appropriate for the data or desired result to unsureness for how to judge different varied responses against each other. Template methods can be used, as well as assumptions such as uniform data distributions, to guide a success metric of some sort, but these will necessarily be more vague than the extremely relative error metrics we posit in this dissertation. There is no obvious universal solution to this conundrum, as any solution will be implementation-specific, required to use implicit data from the experiment design, but assumptions like a uniform data distribution may be useful in many cases as a skeletal framework for unsupervised success metrics which could, if necessary, be applied in a permuted fashion against the responses from several different sub-types of algorithm which other case-specific data is unknown (such as comparing results of Isomap and C-Isomap algorithms when the manifold transformation type is unknown).

d. Resource Consumption

Allowing a range across the scale of execution time/resource consumption vs. accuracy scale is very useful (for example, a quick-response analysis which may drive the justification for a more in-depth and resource-consuming analysis during system idle time)— between the core algorithms (Isomap vs. LLE) and their variants (such as the use of a random sampling of landmark points to approximate results with reduced execution-time demands) a sufficiently populated spectrum is available to serve this purpose.
e. Neighborhood Size

We posit that the epsilon-neighborhood is a better neighborhood selection method than the k-neighborhood method from our experimental data and theoretical investigation of its ability to deal with outliers or density differences, even if it can be more problematic, as our data also shows. When execution resources are available to do so, however, we suggest the effectiveness of the epsilon-neighborhood variant of dynamic neighborhood-size selection. [Brand]

f. Sampling Error (Noise)

Our error metrics and experiments find that these core algorithms are quite robust under relatively high levels of noise, especially the more involved Isomap algorithms. Different experimental methods could be used to mollify any problematic effects of different types of error contamination, but that is beyond our scope here.

g. Error Contaminating Curve Dynamics; n. Curve Dynamic Nearness

As can be seen from a comparison of the success levels between the “swiss roll” and other transformations in our experiments, these concerns are very real issues for the success expectations of the core algorithms. In section 1 of this chapter, following, we introduce a potential algorithmic modification for dealing with these issues more robustly.

h. Curve Sampling Density

Without presupposed knowledge of the transformation function (which we have here, but only wish to use for error metrics so as not to inappropriately simplify the algorithms), we will need to take some measure of estimated curvature via comparative density [Brand], tangent estimation [Bengio], or some similar method.
To investigate and confirm the usefulness of sampling densities based upon curve dynamics, we compared our uniform-density data used in the core experiments with an equivalently sized and scaled data sample that was sampled non-uniformly. To create this comparator data set, we used a sampling function over the $x$-axis (used to project to the $z$-axis for the transformed manifold) for which sampling density was directly related to the change in tangent slope between sampled points. When comparing this alternate sampling density (with an appropriately modified latent manifold for error calculations, scaled so as to cover the same coordinate volume as the core experimental setup) with the original experimental data we found that the overall metrics did not evaluate the density-based distribution as obviously more accurate as can be shown by the following graphs:

---

$^{26}$ Density-based point distribution was based on the arc-length of the transformed manifold and chosen so that the difference between the values for slope (value of the derivative of the transformation function at that point) for adjacent sampling points along the transformation axis remained constant.
Figure 15: MSE and DCM Values for Uniform and Variable Sampling Density Data Sets using k-LLE Algorithm over $z = x^2$ Transformation
In an attempt to understand this somewhat unexpected result, we compared the graphs of per-point error for an errorless version of the data between the two sampling-density methods:

![Figure 16: DCM Values Per Point for Uniform and Variable Sampling Density Data Sets using k-LLE Algorithm over Noiseless $z = x^2$ Transformation](image)

And to look more closely at the effect of the sampling density upon error levels by constraining the view to the points near the $x$-axis, minimising the edge effects\(^{27}\) of the data set and the distortion inherent to the algorithm’s reconstruction:

\(^{27}\) See the following sub-section, VI.(i), for more on edge effects.
Figure 17: DCM Values Per Points Constrained by y-axis Range of [-2,2] for Uniform and Variable Sampling Density Data Sets using k-LLE Algorithm over Noiseless \( z = x^2 \) Transformation

These graphs help illustrate that whereas increased sampling density does indeed seem to increase the accuracy of the reconstruction, the decreased sampling density in other sections of the transformed manifold may not be worth the cost (in overall accuracy) if, as in our experiments, there are overarching data-set size-constraints. The concerns and directive related to sampling density and its relation to curve dynamics may be best deployed when a uniform sampling has been effected to sufficient resolution and the system in which the algorithm is to be executed allows for an increase in total data set size, allowing a most effective targeted use of these extra data-points. This experimental illustration also reminds us of the paradigm in which our posited metrics are most useful— they are chosen to be maximally generic, but such a choice can require, in instances such as these, that the experimenter avoid allowing the comparative data results to be to far generalised. Though it may not be the most manipulable data structure, this is also a reminder
that retention of the per-point error structure can be considerably more informative at times than a single error value for an entire reconstruction.

\[ \text{i. Outliers and Edge Effects; m. Outliers} \]

Again, the epsilon-neighborhood selection method is a manner of dealing with outlier difficulties, as well as is simply culling outliers with statistical methods from input or output (again, given that this does not disrupt the assumptions or goal of the algorithm’s implementation). For dealing with the stretching/skewing effect of edge-points of a manifold sample (see Figure ?? for an example of the error-exacerbation of the edge-effects for even a noiseless sample)\(^\text{28}\), it is possible to modify the algorithm to use edge points estimated as likely to have a non-uniform (non-centered or otherwise abnormal) neighborhood distribution as neighbors only and removing them from the reconstructed embedding (giving structure information to their neighbors, but having no neighbors or output embedding of their own). This may be acceptable for arbitrary manifold samples where “extra” points can be sampled for this purpose, but may not be appropriate for static data sets in which all data points must be mapped for categorisation or other purposes.

\(^{28}\) Epsilon neighborhood versions of the result graphs are almost identical except the error values are slightly lower, scaled proportionately.
Figure 18: Edge Effect Illustration—MSE and DCM Values Per Point for LLE and Isomap over Noiseless $z = x^2$ Transformation
j. Remapability of Output

Though the process by which our core algorithms map the transformed manifold to the reconstructed low-dimensional version of the data distribution does not create a conception of a specific mapping between the input and output data (and there are algorithms which specifically do this, like the charting algorithm [Brand]), the ordering of data points is retained, so some rough mapping between the coordinates of the input and output for each point can be pursued in a manner similar to curve-fit algorithms or by using a neural-network, for example, to learn the transformation type of each axis.29 It would likely be preferable, for accuracy and computational efficiency, if this mapping were discovered as part of the standard algorithmic process, but it is not clear how one might augment one of the algorithms in our core experiments to accommodate this wish.

k. Dimensional Orientation

This data may or may not be relevant, dependant upon application, but since the relationships are destroyed by the general process of restructuring the dimensional data, the only real option seems to permute possible orientations according to an expected form or template as might be necessary.

l. Transformed Output

Transformations as discussed in Chapter IV can be applied if some orientational form or template is known (such as a basic grid structure which can be realigned with the coordinate axes), or data can be transformed as such in order to coincide with some displayability criterion for visualisation (longer/denser dimensional spread rotated to the $x$-axis for left-to-right readability), but without

---

29 These methods, of course, will have their classical issues, such as in the potential indifferentiability between noise-skewed approximations of a low-dimensional polynomial and a fair approximation of a more complex descriptive function, requiring templates, guidelines, or reliance upon error-fit guesses as to the form of the function best describing the data.
some case-specific knowledge or criterion, output transformation is generally neither relevant nor pursuable.

\textit{o. Non-Differentiable Transformation Functions; p. Non-Continuous Transformation Functions; q. Non-Onto Functions}

Discontinuities in differentiability (as long as the transformed manifold remains continuous) do not pose much of a problem for re-embedding, as shown by our core experiments’ results with the absolute value function, as long as the piecewise function could be emulatable to some useful resolution by some continuous function. However, accurate re-mapping of these transformation functions may be difficult since an arbitrary-resolution sample of a piecewise function could possibly be emulated by a similar, if much more complex, differentiable function.\textsuperscript{30} Non-continuous and non-onto transformation functions are not applicable for consideration for our core algorithms since there are underlying assumptions to these algorithms and the transformations for which they are applicable (specifically the isometry/conformal mapping and the smoothness assumptions [Raykar]) that disregard such degenerative possibilities. Some concern is presented in the literature [Bengio] for the problematic existence of multiple disparate manifolds in a sample space, but this should be either emulatable by a combination of non-continuous and piecewise functional descriptions (and so covered by our previously mentioned concerns) or reshaped via problem definition through scope or sampling assumptions.

\textsuperscript{30} See footnote \textsuperscript{29}. Also, case-specific hints as to a disruption of continuous behaviour in sections of the mapping might be usable, but these may fail easily, possibly detecting either false positives or false negatives.
VI.1. A Conjectured Algorithmic Modification for Proximity Issues

An extension suggested to us by the results and reading of this set of experiments is that, given an assumption of a continuous and smoothly transformed manifold, the difference in direction of a distance vector between neighboring points could be used to modify that neighboring point’s subsequent relationships in inverse proportion to the difference in orientation (angles) between distance vectors of the neighboring point in order to constrain output manifold behaviour towards a more smooth and more near-locally linear behaviour. This introduces an extendable concept of momentum or an attempt to constrain noise with a physics-style field-type representation and reconstruction of a manifold since the effect of a distance vector could be pseudo-gravitationally extended through several points. This momentum would assist with problems of interference between nearby non-local sections of the manifold and constrain error.31 It would also combine a more global view of the manifold reconstruction like Isomap’s with the relative speed of the local view used by the LLE algorithm if applied to the latter.

31 Labeled concerns (g) and (n).
The reconstruction process’s weightings are modified inversely proportionally to the difference between a neighbor’s direction and its own neighbors direction—in this example, \( \cos(\theta_{ij}) \) would be applied to the weighting of \( x_j \) for \( y \)’s reconstruction in addition to a similar weighting modifier from all other points which, like \( x_i \), had \( y \) as a neighbor.

To test the usefulness of the algorithm, we ran a single-step version and a three-step (each step being the number of subsequent neighbors through which the momentum influence was applied) version of the method applied to the k-LLE algorithm, diluting the effect of steps beyond the first by half each time over examples from our core experiments as well as over an auxiliary coil structure (as in Figure ?!), using a cosine comparison of the angles to acquire the angle-momentum multiplier (which also would maintain similar reverse angles and avoid diluting the original neighbor’s relationship). Of course, the grid from our experiments should cancel out the effect of the momentum weighting for the errorless cases, the symmetries creating equivalent weighting deterioration, but we expect better noise reduction as the effect of points with a large magnitude for the third-dimensional component of the noise addition on the reconstruction is diminished.

Unfortunately, the coil function was still unfoldable\(^{32}\) by our algorithm when the coils of the transformation’s loops were sufficiently close together to interfere with the neighborhood generation process and, as shown by the Figure ?!, performance tended to be, on average, similar to the performance of the unmodified

\(^{32}\) Desired results: uniformly-spaced points on a single-dimension with point-ordering retained from the latent manifold (line).
algorithm for the data coherence measure and though performance for higher levels
of error returned an improved value for the emulated best-fit MSE measure when
using the 3-step version of the modification, when compared to the lack of
difference in the alternate measure, may have been due to the algorithm’s output’s
tendency for better rotational adjustment and scaling rather than better actual
performance.
Figure 20: Error Results for Momentum-Variant of k-LLE Algorithm for “Swiss Roll” Transformation
An alternate implementation of this momentum-style theory centers around a method for constraining the neighborhood selection rather than weighting the reconstruction of a point’s embedding. In an attempt to emulate the idea behind the momentum-based algorithm, that information about potential neighbors’ neighborhoods can be used to present a more educated selection (or weighting) of the original point’s neighborhood, we implement two alternate methods of single-step momentum neighborhood selection. In both, we normalise the distance from a point to all other points in the data set to \([0,1]\). In the first method, we then use the normalised differences of each other point to our original point as the original point’s distance data for neighborhood selection, discarding the original point’s normalised distance data. In the second method, we use these normalised differences of the other points to scale (by direct per-figure multiplication) the original point’s normalised distance data (instead of discarding it). Comparing these two alternate neighborhood selection methods in the k-LLE algorithm for the “swiss roll” transformation, we acquired the following data:
Figure 21: Error Results for Alternate-Neighborhood-Selection-Variants of k-LLE Algorithm for “Swiss Roll” Transformation
Again, though the “normalised neighbor’s neighborhood” method seems to be something of an improvement over the standard k-LLE algorithm, according to the MSE graph, this may be an artifact of the rotation and scaling process, since the DCM figures show no noticeable difference in results for the alternate methods. The alternate neighborhood selection methods also failed to unravel the coil function that we used previously as a test for the momentum-based algorithm— of all of the algorithms that we use throughout our experiments in this dissertation, the only ones that can unravel this function successfully are the epsilon versions of the core algorithms, and only when the neighborhood radius is taken to be just less than that separating the sections of the coil.

Another theory which might lead to an alternate algorithmic extension would be to attempt to create a skeleton framework from a sine-wave (Fourier) decomposition of the transformed and noise-perturbed manifold or some other either generalist, generative, or even case-specific framework, if possible. Constraining, to a variable degree, the noise to that skeleton while allowing components (with a certain level of contribution to the landscape) to change the structure of the manifold skeleton offers what is basically an alternate extension of the PCA algorithm that is used in both of our core algorithms.\textsuperscript{33} We have not endeavored to modify our algorithms in this presentation to take advantage of structural knowledge in the data (and have actually attempted specifically to avoid approaching any such modifications) even though, for example, our momentum algorithms might well perform noticeably better if the momentum scaling factor were constrained to an orientation reinforcement that took advantage of the grid-like structure of the latent manifold (using a transformation of the cosine angle comparator to do so, as a possible option). However, since some structural data can be extracted from procedures like a variance-based latent dimensionality conjecture\textsuperscript{34}, perhaps some structural knowledge can be extracted online to modify the manner in which the algorithm executes a discriminatory function (like the angular comparison in our example) as long as attention is granted to the possibility of pathological degenerations when using such automated algorithmic directives.

\textsuperscript{33} In a similar but inversely directed manner of the SDE algorithm presented by [Weinberger].

\textsuperscript{34} See [Camastra] and [Pettis] for examples.
Chapter VII. Algorithmic Extensions

In this chapter we would like to discuss potential applicative algorithmic extensions of both our core algorithms and the general field of dimensionality reduction algorithms.

VII.1. Visualisation

Visualisation seems, from the attributed of the algorithms we have outlined in this paper (and the burden of proof from extensive real-world usage), one of the applications best-suited for the core algorithms we used in our considerations and experiments. Allowing for outlier culling, arbitrary transformational adjustment according to whichever criterion might seem most appropriate for the application, and other algorithmic shortcuts, these algorithms also give the most regular performance when judged by an evaluative metric which is motivated by this applicative perspective (our data coherence measure).

VII.2. Natural Categorisation

The other application which seems best suited to our core algorithms is the natural non-reinforced categorisation along feature based-axes extracted from the high-dimensional input data. With the inclusion of a simple output-data density-based grouping pass, the natural dimensionality-detection features inherent to the linear (specifically PCA) dimensionality methods allow for a categorisational engine which could be useful on its own or as part of some larger construct.
One important consideration for such categorisational implementations, however, is that natural human categorisation may not be in line with the results of a mechanical statistical categorisation as it may often depend on attending to features which may not necessarily be the most obvious is a historyless or non-connotative sense. To cite the literature’s real-world data handwriting sample example [Tenenbaum], it may not be obvious upon what axes the handwriting samples are organised—some rough guess can be made that approximates the algorithm results, such as the tendency for organisation according to slant and loops for handwriting samples, but these axis-namings are often imperfect and require human intervention to apply proposed usefulness to the algorithmic output. This is not a crippling drawback, but worthy of note as this applicative sense may be most useful when used as an internal mechanism of a larger process or as a suggestion or guide for categorisations rather than a replacement for natural human categorisation methods.

VII.3. Incremental Versions

The core algorithms in our considerations are not configured for simple application in an incremental sense, as they require distance matrices to be recomputed on the addition of data— the Isomap algorithm, in particular, with its shortest path calculations and global view of the algorithmic situation may require considerable recomputation when adding data points to the input set. However, these algorithms can be modified in implementation to allow for incremental data addition when necessary.

Both algorithms can use a maintained data structure for distance computations and neighborhood constructions indexed by positional coordinates in order to allow the algorithm immediate access to the area which is affect by the addition of the data point in question, allowing the LLE algorithm to compute new neighborhoods and adjust their embeddings only where necessary and allowing the Isomap algorithm to recompute shortest paths and, if a further matrix of paths is kept, allowing for a change in value by difference rather than a full recomputation of these shortest paths. A recomputation of the linear method for embedding would
still be necessary, however, and due to the expanding space requirements of these matrices, however, it may be more practical to expend the time required to recalculate the entire standard algorithm’s paths, neighborhoods, and outputs.

An alternative method would be to use the guiding theories behind the algorithms to create new methodologies established purposefully for incrementality. A neural network implementation\(^{35}\) might be well suited for an incremental version of such an algorithm.

### VII.4. Neural Network Implementations

The core algorithms presented here do not lend themselves to an obvious and straightforward implementation via neural network, but again, perhaps their intimations could be used to design such a system. There are examples of alternate algorithmic methods to achieve the goals of dimensionality reduction using neural networks which may be used as a reference and guiding point for such an implementation. [Jones]

One possible way to implement the ideas behind the algorithms we have studied here within a neural network architecture would be to seed the neural network with high-verity data from the algorithms’ assumptions— the local-neighborhood distance values, for example. Allowing the network to learn a coordinate mapping (and/or global distance matrix information) in which this high-verity data is kept as constant as possible should allow for a functional implementation and the use of a learning objective function that is more global in scale and scope, directing the behaviour of the system in a global rather than local manner. This style of implementation may also be used with a temperature-based verity-rating on older annealed data (concretising its values as the algorithm continues to run) in order to allow for a potentially useful and time-saving incremental implementation of the dimensionality reduction techniques.

\(^{35}\) See the following section, VII.4.
VII.5. Data Compaction

Ultimately, the set of algorithms in the field of dimensionality reduction (and its subfield of manifold learning) perform some of the functions necessary for an extremely robust and generic data compaction and/or function learning method, and may perform sufficiently well as such in certain scenarios, but do not lend themselves to such due to some of their underlying assumptions (as referenced in our algorithmic concerns) and the conflict of these assumptions with the necessary breadth of scope for such a method.
Chapter VIII. Conclusion

Our experiments and investigations have offered considerable insight into not only the Isomap and LLE algorithms and their direct variants, but their algorithmic relatives, and perhaps most importantly, the dynamics of the scenarios for which these algorithms are used.

We have offered a experimental template for comparative analysis of manifold learning and dimensionality reduction algorithms and offered a number of performance metrics which may be used, each of which with their own set of evaluative criterion, allowing a metric to be chosen custom to particular situation’s needs. The most generically useful of these we find to be a version of the standard MSE measure (when the algorithmic output is permuted by dimensional orientation of resultant axes, rotated, and scaled according to a best-fit linear programming script which minimises the MSE value) and a data coherence measure\textsuperscript{36} which allows an almost complete avoidance of the inflated-error performance evaluation aspect contributed by the arbitrary transformations within the output created by these algorithms.

We have offered a comparative analysis of the functionality of the Isomap and LLE algorithms, discovering their comparative accuracy under increasing amounts of additive noise, their approximate functionality limitations under such additive noise, and posited the use of dynamic neighborhood declaration to remove the algorithms’ shared single free parameter as well as maximise accuracy. We have also offered up scenario boundaries and concerns for the function of these algorithms, through which we have offered guidelines for algorithmic design, improvement, and further analysis. There is no obvious and universally more desirable algorithm here (the main tradeoff between the two core algorithms, and

\textsuperscript{36} The results of which can also benefit from a ratio-based comparison to remove the output’s scaling transformation rather than a direct value comparison (which we used to acquire our presented experimental results).
many of the modifications thereof, being the classic paradigm trading resources for accuracy), nor did we expect to find one, but the guidelines created by a combination of our experimental comparisons and scenario concerns also helps allow for more precise construction of specifically custom-modified algorithms to suit an implementer’s instantiational purpose.

Though we have offered some ideas for pseudo-hybridisations of the two algorithms which we studied most closely, none of the designs that we tested here performed convincingly better than the original algorithms, but the momentum-based and other motivating ideas may yet yield some useful mechanism under different implementation.

Finally, though we concluded that the set of algorithms detailed herein are not suitable for the purposes which originally sparked interest in this investigation, through our analysis we have posited a set of applications for which these algorithms seem most practical. The fields we have studied here, like many of the subfields related to artificial intelligence, are at a conjunction of many different, and seemingly disparate, areas of study—we feel that all of the fields concerned can only benefit from continued investigation of these particularly interesting areas of conjunction.

### VIII.1. Further Work

The fields of dimensionality reduction and manifold learning would quite possibly benefit from certain continuations of the work and ideas presented here:

- New algorithmic design focusing upon the limitations of the current generation of algorithms as presented in our analysis.
- Verification of comparative analysis of the variable-neighborhood selection method applied to the core algorithms.
- Applications of the theories presented here to non-isometric transformations.
- Implementations of purely conjectural improvements offered here.
- Relative comparisons of other current dimensionality reduction techniques.
• Dynamic case-detector sub-algorithms that allow for unsupervised case-specific algorithmic improvements.

• Extensions of the test suite to cover other dynamic situational spectra—e.g.: cost benefit analysis of changes in global data density; direct benefit of using different sampling geometries within the raw data field.

• Extensions of analyses to extend to related fields—e.g.: pure statistical analysis; data compaction.

• Dynamical systems analysis of algorithms to gain further insight into behaviours.
Bibliography


Huo, Xiaoming and Chen, Jihong, Local Linear Projection (LLP), School of Industrial & System Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205.


Riden, James, *Unsupervised Learning on Galaxy Spectra*, Division of Informatics, University of Edinburgh, 2002.


