Stockholms universitet

Non-Linear Dimensionality Reduction by an Information Theoretic Optimal Manifold Approach

Ruben Ridderström

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2021:12
Matematisk statistik
Juni 2021

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Matematiska institutionen

Stockholm University

Mathematical Statistics
Stockholm University
Master Thesis **2021:12**
http://www.math.su.se

Non-Linear Dimensionality Reduction by an Information Theoretic Optimal Manifold Approach

Ruben Ridderström*

June 2021

Abstract

To visualize and understand high-dimensional data, performing some kind of dimensionality reduction is often required. Traditional methods such as principal-component analysis are not able to generalize well to non-linear data. Therefore non-linear methods are required. In this thesis we demonstrate such a method originally presented in the paper 'Optimal Manifold Representation of Data: An Information Theoretic Approach' by Chigirev and Bialek. This method uses information theoretic concepts to view dimensionality reduction as a compression problem to find the underlying manifold of the data. The method is nonparametric and has linear time complexity. There is one adjustable hyperparameter that allows us to look at the data structures at different spatial scales. In addition to demonstrating the method with intuitive test examples of clustering and manifold learning we also give a brief overview of fundamental information theoretic concepts and cover some basics of dimensionality estimation. We show that the method does manage to capture the underlying structure of the data both when performing clustering and in identifying underlying manifolds. And that it compares favorably to self-organizing maps which is a well-established method for performing dimensionality reduction.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: ruben.ridderstrom@gmail.com. Supervisor: Chun-Biu Li.

Acknowledgements

I would like to thank Prof. Chun-Biu Li, my supervisor at Stockholm University, for his guidance and support throughout this project.

Contents

1 Introduction

One of the main concerns when working with data is getting a good understanding of it. This can be done in a multitude of ways. We may want to visualize it which requires us to find a representation of the data in no more than three dimensions. Other ways towards understanding can be to find relevant statistics, extracting the most relevant features or removing the noise from the data. For all of these purposes it is often prudent to perform some kind of dimensionality reduction.

A common working assumption to perform dimensionality reduction is that the data lies on some unknown, underlying low dimensional manifold. This idea can be found in the paper by Hotellings [7] on Principal Component Analysis (PCA). Even though PCA can be shown to perform well when the underlying manifold is linear, linearity is clearly too strong of an assumption to make in the general cases. Instead we need to use non-linear methods that work well on non-linear data.

Information theory has historically been concerned with the theory of communication. In the founding paper of information theory Shannon introduced the rate-distortion theory [4]. Rate-distortion theory can be used to describe the trade-off between the necessary transmission rate of a channel when sending a message and the distortion introduced in the message on the receiving end. In this thesis we study and examine the method introduced in the paper 'Optimal Manifold Representation of Data: An Information Theoretic Approach' [5] that directly builds on and extends on rate-distortion theory. The idea of the method is to restrict the rate, also known as mutual information, between the data and the learned manifold. With the hope that by doing so extraneous noise can be removed and the underlying manifold will reveal itself.

The upcoming section cover basic information theoretic concepts which lay the foundation for the following sections. We then move on to describe how information theory can be used to perform data compression. This is followed by a thorough description of the optimal manifold method, how we solve for it and interpretations of the solution. The last two sections cover examples of the method applied to both clustering problems and to the problem of finding an underlying manifold. After this we briefly cover three mathematical concepts on how intrinsic dimension can be defined, as well as reproduce an example from the optimal manifold paper that demonstrate how to perform an estimation of intrinsic dimension. In the last section we discuss our findings and some possible extensions.

2 Base concepts of information theory

2.1 Shannon Entropy

The fundamental concept in information theory is that of Shannon entropy, information entropy or simply entropy. We denote a discrete random variable as X. We denote its alphabet, the potential values X can take, with $\mathcal{X}$. Furthermore the associated probability function is simply written $p(x)$. Here it is to be understood from the context that $p(x)$ is the probability function of X, and that x is an element in $\mathcal{X}$. We use $|\mathcal{X}|$ to denote the number of elements, also known as the cardinality, of $\mathcal{X}$.

The entropy of X is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{1}$$

We will use the convention that $0 \log 0 = 0$ which can be argued from the limit $\lim_{x \to 0} x \log x = 0$.

The definition leads to several properties of the entropy. The first is that the entropy is only a function of the probability function $p(x)$ of X. The actual values of the alphabet $\mathcal{X}$ are not of importance.

Rewriting the entropy by moving in the minus sign

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)} \tag{2}$$

we see that each term is the product of two non-negative numbers. Thus the lower bound of entropy is zero. Using Jensen's inequality it can further be shown that the entropy of X has an upper-bound of $\log |\mathcal{X}|$. Putting together the lower and upper bound we have

$$0 \leq H(X) \leq \log |\mathcal{X}| \tag{3}$$

The lower bound of 0 is achieved for a deterministic distribution which only takes on one value, that is $p(x) = 1$ for some element x. The upper bound is achieved for a uniform distribution in which case $p(x) = 1/|\mathcal{X}|$ for all x.

Entropy is naturally extended to multiple variables by the definition of *joint entropy*

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y) \tag{4}$$

and *conditional entropy*

$$H(Y|X) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y|x) \tag{5}$$

By splitting up the joint entropy, we arrive at the chain rule which naturally connects joint and conditional entropy.

$$\begin{aligned}
H(X,Y) &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(x,y) && (6)\\
&= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y|x)p(x) && (7)\\
&= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y|x) && (8)\\
&= -\sum_{x\in\mathcal{X}} p(x)\log p(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y|x) && (9)\\
&= H(X) + H(Y|X) && (10)
\end{aligned}$$

Thus the joint entropy of two random variables equals the entropy of one plus the conditional entropy of the other.

2.2 Dice example

To motivate the definition of entropy we will consider the example of a fair dice. That is a uniformly distributed random variable X with possible outcomes $\mathcal{X} = \{1, 2, \ldots, 6\}$ and the cardinality of $|\mathcal{X}|$ equal to six.

Information theory has historically primarily been concerned with communication and the limitations of transmitting information. Say we roll the dice a single time for an outcome x in $\mathcal{X}$. If we communicate this outcome to a receiving part, how much information has been transmitted? Maybe the most obvious measure is the cardinality of the alphabet $|\mathcal{X}|$ which in this case is 6. Other options would be monotonic functions of the cardinality.

In the founding paper of information theory, Shannon [4] argued for the choice of the logarithm of the cardinality, for reasons of intuition and mathematical suitability. The logarithm is closer to our intuitive feeling of a proper measure as we expect that sending two messages instead of one would double the amount of transmitted information. This would then correspond to increasing the logarithm from one to two. Mathematically it is also more suitable since the expressions we are concerned with are simple in terms of the logarithm, but would be awkward if expressed with respect to the cardinality.

For a uniform distribution, as in the case with a fair dice, we can evaluate the expression for entropy

$$H(X) = -\sum_{i=1}^{|\mathcal{X}|} p(x_i) \log p(x_i) \tag{11}$$

$$= -\sum_{i=1}^{|\mathcal{X}|} \frac{1}{|\mathcal{X}|} \log \frac{1}{|\mathcal{X}|} \tag{12}$$

$$= -\log \frac{1}{|\mathcal{X}|} \tag{13}$$

$$= \log |\mathcal{X}| \tag{14}$$

Thus a regular six sided fair dice has an entropy of $\log_e |\mathcal{X}| = \log_e 6 = 1.79$ nats and a three sided fair dice would have an entropy of $\log_e 3 = 1.01$ nats. So for a uniform distribution the amount of information contained in one outcome x of X decreases when the number of potential outcome decreases. If there is only one possible outcome the entropy is $\log 1 = 0$. So in this case we gain no information by getting to know the outcome of the dice roll. This is similar to the fact that when a random variable only has one outcome the variance of it is zero.

The base of the logarithm used only decides the unit of information and is therefore essentially up to choice. Using base 2 the unit is called bits while with base e the unit is called natural unit or nats. We can at any time perform conversion between the units by simply changing the base of the logarithm.

2.3 Bernoulli example

We know the uniform distribution maximizes the entropy of a random variable. The last section dealt with this case. Here we will exemplify the entropy between its lower-bound of zero and upper-bound of $\log |\mathcal{X}|$ when the outcomes are not necessarily uniformly distributed. In Fig. 1 we have plotted the entropy of a Bernoulli random variable X which has a probability p of being successful in which case it takes on the value 1, or it fails with probability $1-p$ in which case it takes on the value 0. The alphabet $\mathcal{X}$ thus contains the values 0 and 1.

In the case when $p = 0$ or $p = 1$, the outcome is given and the entropy is zero. The entropy is maximized for the uniform distribution where $p = 1 - p = 0.5$ in which case it is $\log_e |\mathcal{X}| = \log_e 2 = 0.693$ nats.

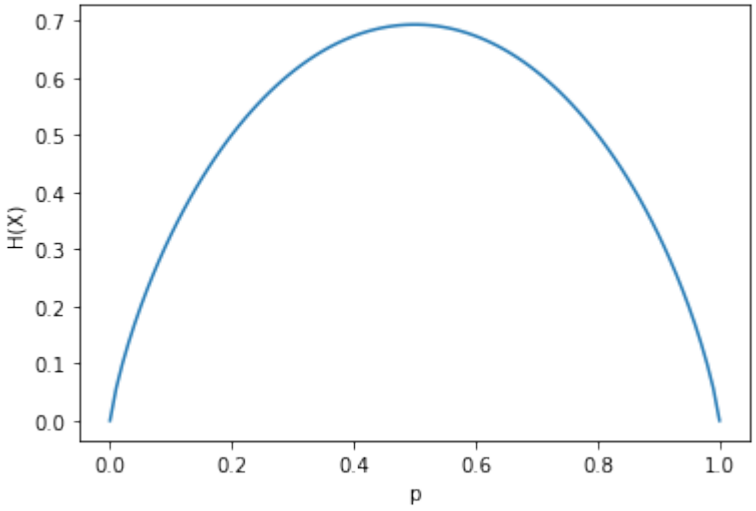

Figure 1: Plot of the entropy for a Bernoulli random variable over a range of probabilities of success p. The highest entropy is achieved for p=0.5 which is the same as a uniform distribution. We can notice that the entropy is a concave function with respect to p.

2.4 Relative entropy

Next we define the relative entropy also known as the Kullback-Leibler divergence. Let p and q be probability functions on $\mathcal{X}$. The relative entropy between p and q is defined as

$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \tag{15}$$

As is the case for entropy, the unit of relative entropy is given by the base of the logarithm. The lower bound can be shown to be zero and occurs if and only if the distributions p and q are equal. Relative entropy can be thought of as a distance measure between the distributions although it is neither symmetric nor obeys the triangle inequality.

Assume we have random variable X with an alphabet $\mathcal{X}$ and distribution p. We can make a connection between the entropy $H(X)$ and relative entropy $D(p||u)$

where u is the uniform distribution $u(x) = 1/|\mathcal{X}|$.

$$
\begin{aligned}
D(p||u) &= \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{u(x)} && (16)\\
&= \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{1/|\mathcal{X}|} && (17)\\
&= \sum_{x\in\mathcal{X}} p(x) \log p(x) - \sum_{x\in\mathcal{X}} p(x) \log \frac{1}{|\mathcal{X}|} && (18)\\
&= \sum_{x\in\mathcal{X}} p(x) \log |\mathcal{X}| - \left(- \sum_{x\in\mathcal{X}} p(x) \log p(x) \right) && (19)\\
&= \log |\mathcal{X}| \sum_{x\in\mathcal{X}} p(x) - H(X) && (20)\\
&= \log |\mathcal{X}| - H(X) && (21)\\
&\iff H(X) = \log |\mathcal{X}| - D(p||u) && (22)
\end{aligned}
$$

This gives another point of view on the lower- and upper-bound of entropy.

In the case when the distribution p equals the uniform distribution the relative entropy is zero, in which case the upper bound of the entropy is achieved. On the other end of the spectrum is the case when p assigns all probability to a single outcome. Then the relative entropy equals $\log |\mathcal{X}|$ and the entropy is zero.

2.5 Mutual information

Finally we define the mutual information. Let X and Y be random variables with joint distribution $p(x, y)$ and marginal distributions $p(x)$ and $p(y)$. The mutual information is the relative entropy between the joint distribution and the product of the marginal distributions, i.e., the independent distribution.

$$I(X;Y) = D(p(x,y)||p(x)q(y)) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \qquad (23)$$

As with relative entropy the lower bound of mutual information is zero and the unit is determined by the base of the logarithm used. In contrast to relative entropy the mutual information is symmetric $I(X;Y) = I(Y;X)$. We can rewrite mutual information in terms of entropy. This allows us to interpret it

as information gained of a random variable through the knowledge of another.

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{24}$$

$$= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x|y)}{p(x)} \tag{25}$$

$$= -\left(-\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(x|y)\right) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(x) \tag{26}$$

$$= -H(X|Y) - \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} p(x|y)p(y)\log p(x) \tag{27}$$

$$= -\sum_{x\in\mathcal{X}} p(x)\log p(x) - H(X|Y) \tag{28}$$

$$= H(X) - H(X|Y) \tag{29}$$

Say we are interested in knowing the outcome of X given the outcome of Y. How much does this increase our information of X? Or equivalently, how much does this reduce the entropy of X? This is what mutual information tells us. It can be compared to covariance between two variables. But unlike covariance, mutual information is not limited to describing linear dependencies.

Using the symmetry of mutual information it also follows that the relationship $I(X;Y) = H(Y) - H(Y|X)$ holds. These relationships can be illustrated as in the Venn diagram of Fig. 2.

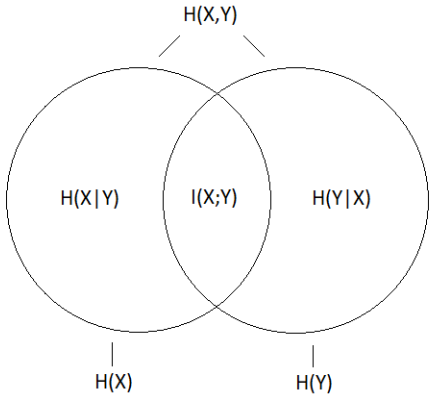

Figure 2: Graphical representation of relationship between mutual information, entropy and conditional entropy.

3 Using information theoretic concepts for compression

One of the main historical concerns of the field of information theory has been that of sending information over a channel [10][4]. And to what extent information can be compressed to allow for more efficient transmissions. To give some intuitions of the concepts introduced so far and connect them to the upcoming section we will here give a brief overview of the concept of compression.

3.1 Lossless compression of discrete data

Assume we want to transmit a message over a channel such that the receiver can reconstruct the original message exactly as it was sent. To make this concrete we assume the message can contain eight different symbols and the symbols can appear in the message with different frequencies. We represent this by a random variable X, where each symbol is an element x in $\mathcal{X}$ and the cardinality $|\mathcal{X}|$ is eight.

We will transmit the message over a digital channel which send bits that are interpreted as either 0 or 1. Therefore we use the base two in our computations. The goal is to transmit the message using as few bits as possible.

The original message we assume to be encoded using a simple fixed length encoding. In this encoding each symbol is assigned a unique sequence of 0's and 1's and each symbol have the same encoding length. Since we are using bits and want to encode eight symbols we need $\log_2(8) = 3$ bits to give each symbol a unique base two representation. The representation of the eight symbols are then

$$000, 001, 010, 011, 100, 101, 110, 111 \tag{30}$$

To compress the message we want to find an encoding which requires fewer bits and still allows us to reconstruct the original message. We can then compress the message, transmit it to the receiver and they can decode it.

The way to improve on the encoding is to give more frequently occurring symbols a shorter encoding length. While the less frequent symbols are given a longer encoding length. One way to do so would be to encode the two most common symbols simply as 0 and 1. The next four most common symbols could be encoded as 00, 01, 10 and 11. And the two least frequent occurring symbols could be encoded as 000 and 001.

Here a complication occurs in the fact that for a fixed length encoding the

receiver of the message knows when one symbol ends and the next one begins, simply by keeping track of how many bits have been transmitted. In a variable length encoding care must be taken in the construction of the encoding to make it uniquely decipherable. This can be done, but for simplicity we ignore this complication for the remainder of the discussion.

For a particular encoding C, denote the length of an encoded symbol x by $l(x)$. In the fixed length encoding, $l(x)$ would equal three bits for any symbol. For the variable length encoding $l(x)$ would equal one, two or three bits. We now define the expected length $L(C)$ of an encoding C.

$$L(C) = \sum_{x \in \mathcal{X}} p(x) l(x) \tag{31}$$

This is simply the expected length of a symbol of the encoded message. It can be shown that the lowest achievable expected length is closely related to the entropy of the message [11] and can be bounded in the following way

$$H(X) \leq L < H(X) + 1 \text{ bits} \tag{32}$$

Thus the amount of compression that can be achieved is closely related to the entropy of the message.

Going back to the message with eight different symbols, using a fixed length encoding each symbol has an encoding length of three bits. So the expected length is also three bits independent of the distribution of the symbols.

We know from before that the entropy of a random variable is maximized with a uniform distribution. In this case the entropy of the message is $\log_2(|\mathcal{X}|) = \log_2(8) = 3$ bits. From Eq. 32 we also know this is the lower limit of the achievable expected length. So for a uniformly distributed message there does not exist an encoding with a shorter expected length than the simple fixed length encoding.

To give another example, assume that six of the symbols of the message only occurs 1% of the time. And that the other two symbols both have an equal 47% probability of occurring.

In this case the entropy of the message is

$$\begin{aligned} H(X) &= -\sum_{x \in \mathcal{X}} p(x) \log p(x) && (33) \\ &= -(6 \cdot 0.01 \log_2 0.01 + 2 \cdot 0.47 \log_2 0.47) && (34) \\ &= 1.42 \text{ bits} && (35) \end{aligned}$$

If we assume the best encoding achieves the upper bound of expected length of

$H(X) + 1$ bits, we then get an expected length of

$$L(C) = H(X) + 1 = 1.42 + 1 = 2.42 \text{ bits} \tag{36}$$

This is lower than the expected length of three bits for the fixed length encoding. So in this case, with symbols being more unevenly distributed, we can compress the message by encoding the data before transmitting it.

3.2 Lossy compression and the Rate-Distortion Function

In the previous section we dealt with encoding discrete symbols in a way that allowed the receiving part to reconstruct the message exactly as it was sent. In the more general case we are interested with encoding and decoding of both discrete and continuous data.

Performing an encoding-decoding process of continuous data that allows for perfectly reconstructing the sent data would require an infinite number of bits. Therefore we will in this section generalize the encoding-decoding process further by letting go of the restraint that the message has to be able to be decoded exactly as it was sent. This in turn requires us to define a distortion measure.

To formalize the more general case we first define the encoder and decoder functions. In the previous section we encoded individual symbols. It can be shown that by instead encoding sequences we can achieve a more efficient encoding which allows for better compression. We therefore define the encoding and decoding functions on sequences.

Assume we want to encode a sequence of n independent and identically distributed random variables $X^n = (X_1, X_2, \ldots, X_n)$. We call X^n the source. We then define a $(2^{nR}, n)$-rate distortion code as an encoding function

$$f_n : \mathcal{X}^n \to \{1, 2, \ldots, 2^{nR}\} \tag{37}$$

together with a decoding function

$$g_n : \{1, 2, \ldots, 2^{nR}\} \to \hat{\mathcal{X}}^n \tag{38}$$

We start the encoding decoding process with a sequence $x^n \in X^n$, and end up with a new sequence $\hat{x}^n \in \hat{X}^n$. We call the possible outputs $\hat{\mathcal{X}}^n$ of the decoding function the reproduction alphabet. The reproduction alphabet is usually the same as the source alphabet. The rate R of the code is the number of bits used per element of the sequence.

To define distortion requires us to define a distortion measure. Many options could be considered, but for discrete data a common choice is to simply count

the number of incorrect symbols

$$d(x, \tilde{x}) = \begin{cases} 0 & \text{if} \quad x = \tilde{x} \\ 1 & \text{if} \quad x \neq \tilde{x} \end{cases} \tag{39}$$

This is known as the Hamming distortion. For continuous data the squared error is often used

$$d(x, \hat{x}) = ||x - \hat{x}||^2 \tag{40}$$

The distortion between the sequences is defined as the average of the per symbol distortion

$$d(x^n, \hat{x}^n) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i) \tag{41}$$

Then for a $(2^{nR}, n)$ code the distortion D is defined as

$$D = E[d(X^n, g_n(f_n(X^n)))] \tag{42}$$

$$= \sum_{x^n \in X^n} p(x^n) d(x^n, g_n(f_n(x^n))) \tag{43}$$

Here E stands for expectation, where the expectation is taken with respect to X^n. It can be shown that rate R is closely connected to how small we can make the distortion D associated with the code. To formalize this we make three definitions [11]:

A rate distortion pair (R, D) is said to be *achievable* if there exists a sequence of $(2^{nR}, n)$-rate distortion codes (f_n, g_n) with $\lim_{n\to\infty} E[d(X^n, g_n(f_n(X^n)))] \leq D$.

The *rate distortion region* for a source is the closure of the set of achievable rate distortion pairs (R, D).

The *rate distortion function* $R(D)$ is the infimum of rates R such that (R, D) is in the rate distortion region of the source for a given distortion D.

For the simple case with a Bernoulli source, and using the Hamming distortion, the rate distortion function can be calculated analytically. Let X be a Bernoulli random variable with probability of success p and denote the entropy as $H_p(X)$. The rate distortion function has the form [11]

$$R(D) = \begin{cases} H_p(X) - H_D(X), & 0 \leq D \leq \min\{p, 1-p\} \\ 0, & D > \min\{p, 1-p\} \end{cases} \tag{44}$$

Using this we plot the rate distortion function for a Bernoulli variable with $p = 0.5$ shown in Fig. 3.

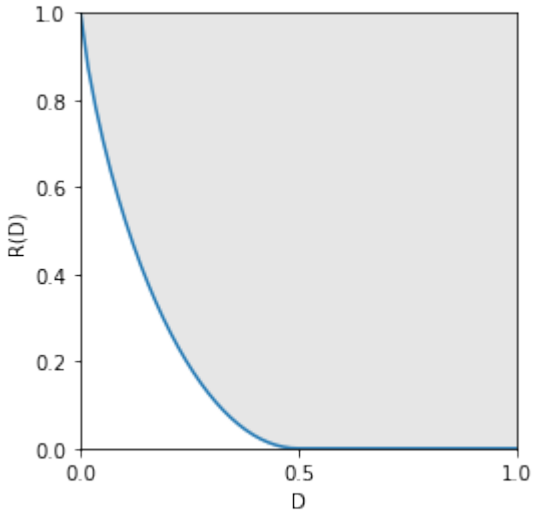

Figure 3: Rate-distortion function for a Bernoulli source with $p = 0.5$. In this case the analytical solution can be calculated. We can see that as the allowed distortion goes up, the necessary rate decreases. This is a general property of the rate distortion function which is monotonically decreasing and convex.

The graph shows the lowest attainable rate for a given distortion. The white area under the graph is thus unattainable. We can see that as we let the distortion increase, the required rate decreases. And in this case we can actually achieve zero distortion by having a rate of one.

Next we define the information rate-distortion function using the mutual information between the input sequences x^n and the decoded output sequences $\hat{x}^n$

$$R^{(I)}(D) = \min_{p(\tilde{x}|x):\sum_{x,\tilde{x}} p(x)p(\tilde{x}|x)d(x,\tilde{x})\leq D} I(X;\tilde{X}) \tag{45}$$

The information rate-distortion function is the smallest attainable mutual information between the source X and an encoding alphabet $\tilde{X}$ for a given distortion. Here the minimization is over the conditional distribution $p(\tilde{x}|x)$.

Surprisingly, it can be shown that the rate-distortion function $R(D)$ equals the information rate-distortion function $R^{(I)}(D)$ [11]. And thus we can find the minimum achievable rate by calculating the information rate-distortion function. We will therefore from now on concern ourself with the mutual information rather than the rate. We also denote the information rate-distortion as simply the rate-distortion.

This gives us an alternative interpretation of Fig. 3 in terms of mutual infor-

mation. We can view the y-axis as the mutual information instead of the rate. And the gray area above the graph is what we are optimizing $p(\tilde{x}|x)$ over to achieve the lowest possible mutual information for a given distortion.

4 The optimal manifold method

In the example with Bernoulli distributed data, the analytical solution to the rate-distortion function can be calculated but in general an analytical solution is not available. Instead we have to resort to numerical methods both for calculating the mutual information $I(X;\tilde{X})$ and the assignment $p(\tilde{x}|x)$.

In this section we will work out the details in doing so. We interpret the symbols of the reproduction alphabet $\hat{X}$ as cluster centers or manifold points depending on the context. The method in explicitly calculating these cluster centers is referred to in the paper 'Optimal Manifold Representation of Data - An Information Theoretic Approach' [5]. We will use the method of this paper to find the underlying manifold.

Often when dealing with data there is the assumption that the data lies on a unknown, low dimensional manifold. The simplest way to extract such a manifold is to use Principal Component Analysis [7], PCA, to tackle the problem. The benefit of PCA is its simplicity. But it assumes that the underlying manifold is linear. For non-linear manifolds we need non-linear methods.

We will instead use the optimal manifold method. To do this we add the assumption that the underlying low dimensional manifold $\mathcal{M}$, can be parametrized by a parameter t, and described by the function $\gamma(t)$. Here $t \in \mathbb{R}^d$ is assumed to be of dimension $d < D$, where D is the dimension of the original high dimensional feature space. Then $\gamma(t)$ is a mapping from the low dimensional parameter space, to the manifold $\mathcal{M}$ in the high dimensional feature space. This means that we are now looking for a conditional probability $p(t|x)$. In addition we want to find the associated manifold points $\gamma(t)$.

As we will be working with continuous data we need to choose an appropriate distortion function. One could imagine choosing a distortion function based on a wide set of criteria. In practice a common choice is the squared error distortion function

$$d(x, \gamma(t)) = ||x - \gamma(t)||^2 \tag{46}$$

which is what we will be using.

4.1 Solving the rate-distortion function

We are looking for an encoding of the data, in the form of the conditional distribution $p(t|x)$, that for a given distortion minimizes the mutual information between the data and the manifold with respect to $p(t|x)$ and $\gamma(t)$. That is, for

the rate-distortion function

$$R(D) = \min_{p(t|x):\sum_{x,t} p(x)p(t|x)d(x,\gamma(t))\leq D} I(X;\mathcal{M}) \tag{47}$$

we want to minimize $R(D)$ with respect to $p(t|x)$ subject to the constraints

- $p(t|x) \geq 0$
- $\sum_t p(t|x) = 1$
- $\sum_{x,t} p(t|x)p(x)d(x,t) \leq C$

Using a Lagrange multiplier we can formulate this as a minimization of the functional

$$\mathcal{F}(p(t|x)) = D + \lambda I + \iint d^D x d^d t \lambda_0(x) p(t|x) \tag{48}$$

where D is the distortion

$$D = \iint d^D x d^d t \, p(x)p(t|x)||x - \gamma(t)||^2 \tag{49}$$

and I is the mutual information

$$I = \iint d^D x d^d t \, p(x)p(t|x) \log \frac{p(t|x)}{p(t)} \tag{50}$$

To minimize $\mathcal{F}$ we want to set the functional derivatives to zero

$$\frac{\partial \mathcal{F}}{\partial \gamma(t)} = 0 \tag{51}$$

$$\frac{\partial \mathcal{F}}{\partial p(t|x)} = 0 \tag{52}$$

Solving Eq. (51) and (52) gives rise to the following equations (For details see Appendix)

$$p(t) = \int d^D x \, p(x)p(t|x) \tag{53}$$

$$\gamma(t) = \frac{1}{p(t)} \int d^D x \, x p(x)p(t|x) \tag{54}$$

$$p(t|x) = \frac{p(t)e^{-\frac{1}{\lambda}||x-\gamma(t)||^2}}{\int d^d t' \, p(t')e^{-\frac{1}{\lambda}||x-\gamma(t')||^2}} \tag{55}$$

Eq. (53) and (55) are similar to the formal solution obtained in rate-distortion theory [11]. In contrast Eq. (54) comes from solving Eq. (51) and gives us the explicit cluster coordinates which we interpret as manifold points.

As we do not have access to the marginal distribution $p(x)$ but only a discrete number of samples we approximate it as $p(x) = 1/N$, where N is the number of samples. This means that the data points are weighted equally in determining the optimal manifold.

4.2 Interpretation of the solution

By looking closer at the equations of last section we can better understand the optimal manifold method. We can see that Eq. (53) is simply the marginalization of t. On the other hand, Eq. (54) can be rewritten as

$$\gamma(t) = \frac{1}{p(t)} \int d^D x \, x p(x) p(t|x) \tag{56}$$

$$= \int d^D x \, x \frac{p(x,t)}{p(t)} \tag{57}$$

$$= \int d^D x \, x p(x|t) \tag{58}$$

Which makes it clear that it is the conditional expectation of x given t.

Finally in Eq. (55) the denominator is a normalizing factor. The numerator consists of the marginal distribution of $p(t)$ followed by a factor very similar to the normal distribution where $\lambda = 2\sigma^2$ and $\gamma(t) = \mu$. Looking back at the functional (48) and its derivatives (51) and (52) we can see that the exponential term follows from the fact that the mutual information consists of the natural logarithm. The form of the exponent is the result from the choice of using the Euclidean distance. So even though we made no explicit distributional assumptions, we see how the forms of distortion function and mutual information lead to the normal distribution.

The formulation of the functional (48) and its derivative (52) with respect to $p(t|x)$ opens up for two perspectives of the λ parameter. Starting out from the perspective of Eq. (48) we can see that if we let $\lambda \to \infty$ the information term will completely dominate the distortion term. In this case minimizing $\mathcal{F}$ is equivalent to minimizing the mutual information. Conversely, if we let $\lambda \to 0$, minimizing $\mathcal{F}$ will mean minimizing the distortion. One way to interpret λ is as the tradeoff parameter between compression (quantified by the mutual information) and the distortion.

From Eq. (55) we can get another viewpoint of λ. Here the λ term appears in the exponents. Increasing λ, in the limit letting λ approach infinity, means the exponents goes towards zero. Thus essentially all points are equally weighted in the computation of cluster centers. In contrast by letting λ go towards zero, closer points are given a larger importance.

The second interpretation of λ is thus that it controls the scales through which we view the data. A small λ means we view the data in small scale, while a large λ means a bigger or potentially even global scale. When compared with the normal distribution we can solve for σ

$$\lambda = 2\sigma^2 \iff \sigma = \sqrt{\frac{\lambda}{2}} \tag{59}$$

Thus for example if we let λ be eight then σ, or the standard deviation, is equal to two. And so the majority of the weight would be placed on data points within a radius of two units from the cluster centers.

4.3 The modified Blahut-Arimoto algorithm

The numerical solution to the rate-distortion function can be calculated using the modified Blahut-Arimoto algorithm. Since we are not only solving the rate-distortion function but also calculating the explicit locations of the cluster centers $\gamma(t)$, we need to add an additional step (61). We can perform one iteration of the algorithm by computing the following steps in order

$$p^{(n)}(t_k) = \sum_{i=1}^{N} p(x_i) p^{(n)}(t_k|x_i) \tag{60}$$

$$\gamma_k^{(n)} = \frac{1}{p^{(n)}(t_k)} \sum_{i=1}^{N} x_i p(x_i) p^{(n)}(t_k|x_i) \tag{61}$$

$$p^{(n+1)}(t_k|x_i) = \frac{p^{(n)}(t_k) e^{-\frac{1}{\lambda}||x_i - \gamma^{(n)}(t_k)||^2}}{\sum_{k=1}^{K} p^{(n)}(t_k) e^{-\frac{1}{\lambda}||x_i - \gamma^{(n)}(t_k)||^2}} \tag{62}$$

Here N is number of data points, K the number of cluster centers while (n) and $(n+1)$ are used to denote the iteration steps. To get started the cluster centers are randomly initialized from the data points and the marginal distribution $p(x)$ is set to be uniform.

The algorithm alternates between estimating the marginal distribution $p(t)$ and conditional distribution $p(t|x)$. The first step (60) chooses $p(t)$ to minimize the mutual information for the current $p(t|x)$. The second step (62) equivalently chooses the $p(t|x)$ that minimizes the mutual information given the current $p(t)$. Between these two steps the location of the cluster centers are calculated (61).

The iteration is repeated until convergence, which is defined as

$$\max_k |\gamma_k^{(n)} - \gamma_k^{(n-1)}| < \epsilon \tag{63}$$

where epsilon is chosen according to the desired precision.

During the formulation we have only used information theoretic concepts and made no distributional assumptions. The algorithm has a linear time complexity with the size of the input N, and all the examples of this thesis had a running time of under a second.

The only parameter which has to be provided to the algorithm besides the number of cluster centers is λ. As discussed, λ can be understood as the scale through which we view the data or the weight which we place on the mutual information. As a starting point for choosing λ, we can generate a range of λ values. For each λ value we use the modified Blahut-Arimoto algorithm to calculate the mutual information and distortion. This produces the rate-distortion curve, which plots the mutual information against the distortion to produce an approximation of the rate-distortion function. We can use this curve to pick out λ values for which the cluster centers neither collapses to a single point, nor falls on the original data points. Examples of these behaviours will be seen in the upcoming sections.

5 Clustering using the optimal manifold method

Even though the optimal manifold method is intended to be used for finding underlying manifolds of the data it can also be used to perform clustering. In this section we will show examples which also serves to clarify the behaviour of the method.

5.1 A non-separable case

The first example uses the data shown in Fig. 4. From the viewpoint of clustering we are interested in classifying the data as belonging to a set of clusters. Finding the appropriate number of clusters can be seen as part of the problem statement. Furthermore we could be interested in hard clustering where each point is assigned to only one specific cluster, or soft clustering where we assign each data point a probability of belonging to each cluster. Using the optimal manifold method will generally give us a soft clustering.

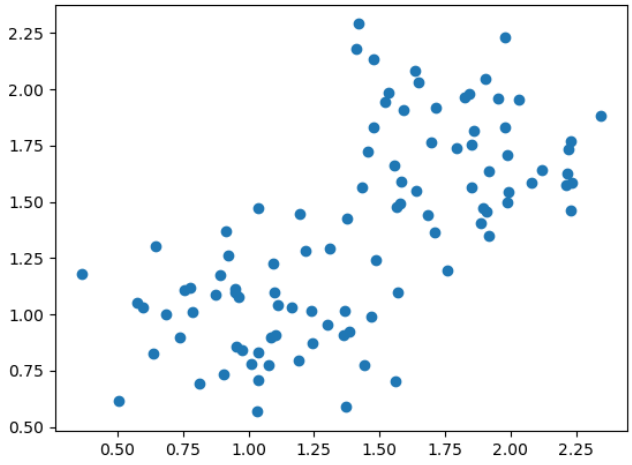

Figure 4: Generated non-separable two-dimensional data. The data has no clear separation boundaries but it is still possible to discern the outline of two separate clusters.

In addition to the input data we also have to specify the number of clusters to

use. For this example we use two clusters. Running the algorithm for a range of λ values sweeps out the rate-distortion curve plotted in Fig. 5.

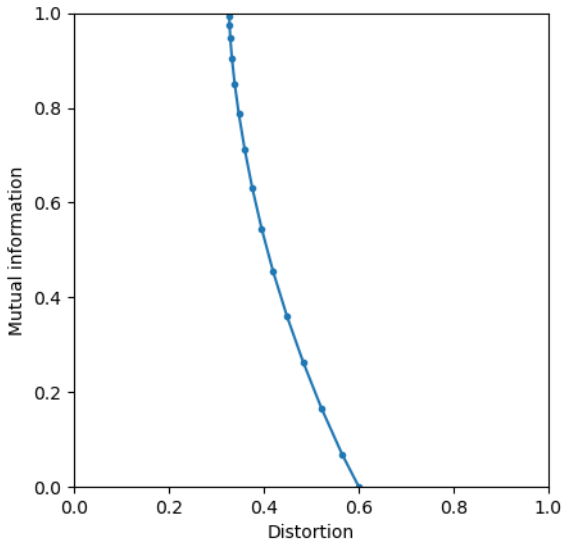

Figure 5: Rate-distortion curve for the non-separable data which shows the trade-off between mutual information and distortion. Each of the points on the curve corresponds to a fixed value of λ. The value of λ varies from 0.01 to 0.7.

We can recognize that the curve is convex and monotonically decreasing as was the case with the analytically calculated rate-distortion function for a Bernoulli source earlier shown in Fig. 3. A clear difference is that the distortion never gets close to zero, no matter how high the mutual information gets. This is as expected, since no matter how the two cluster centers are placed, there will always be many points which lie at a distance to the cluster centers.

In addition we can calculate the cluster centers and plot them together with the data. The result can be seen in Fig. 6.

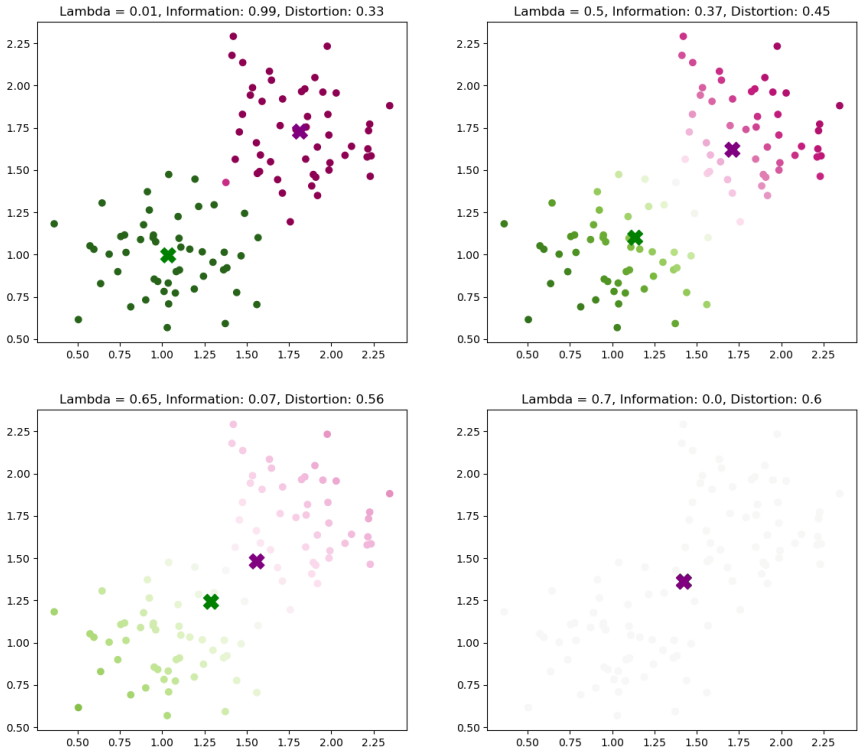

Figure 6: Clustering on the non-separable data using two clusters for four different values of λ. The top left plot corresponds to the point of minimum distortion in Fig. 5. The two cluster centers are depicted as "X" and their locations do not have to lie on any data point. Notice that in the bottom right plot both clusters centers are located on top of each other even though only the purple one is visible in the plot.

Since the method gives a soft clustering, each of the data points have a probability $p(t|x)$ to belong to either of the clusters. The colors of the data points are set according to which cluster they are the most likely to belong to, with the intensity of the color being proportional to the probability. Thus a data point which is green has a higher probability of belonging to the green cluster center and vice versa. In the top-left plot in Fig. 6 the points have a probability near one and so they are all either bright green or bright purple. While in the bottom-right plot all points are almost equally probably to belong to either cluster and thus their color intensities are low.

In every run the algorithm attempts to minimize both the information as well as the distortion but weight their importance differently depending on the value of λ. In the top-left plot λ is very close to zero and so the algorithm places

the cluster centers in a way that minimizes the distortion. This results in a mutual information near 1 and a distortion of 0.33. By gradually increasing the importance of information, we end up with the bottom-right plot where the mutual information is 0 and distortion 0.6.

The alternative perspective is viewing the λ parameter as controlling the scale through which the data is seen. The top-left plot with a low λ is then viewing the data in a small scale, and the clustering is done more heavily weighting nearby data points. This leads to data points belonging to the nearest cluster with a probability of close to one.

On the opposite end in the bottom-right plot, a large value in λ can be interpreted as viewing the data in a global scale. Thus, each data point has almost equal probability to belong to either cluster center. This leads to the cluster centers collapsing and forming one cluster center. Even though the plotting only makes one cluster center visible, both centers are actually located in the middle of the plot on top of each other.

5.2 Self-organizing maps

To see how the optimal manifold method perform compared with other methods, we also apply the self-organizing map [8] to the examples of the upcoming sections. Self-Organizing Maps (SOM) is a well established method to perform dimensionality reduction and visualize data.

The method uses a set of interconnected neurons for which we must specify the shape. If we want to be able to easily visualize the neurons a one- or two-dimensional grid shape is appropriate. Each of the neurons have an associated weight vector which is in the same space as the input data. When fitting the algorithm, the neurons of the grid remain in place. What is updated are the position of the weight vectors.

Before running the fitting procedure, the positions of the weight vectors are initialized. One way to do this is by setting the weight vectors positions to small random values. We then iterate over the input data, or input points. For each point the closest weight vector is found, where the distance is measured using Euclidean distance. The neuron associated with the closest weight vector is called the best matching unit (BMU). The weight vector of the BMU is moved closer to the input point. In addition, all neurons which are located close to the BMU have their weight vectors moved towards the input point.

The updating of the position of weight vector W_v associated with neuron v is

performed according to the formula

$$W_v^{(s+1)} = W_v^{(s)} + \theta(u, v, s) \cdot \alpha(s) \cdot (D(t) - W_v^{(s)}) \tag{64}$$

Here s is the current iteration step, u is the index of the BMU and t is an index to the current input point.

The α function gives the learning rate which decays over iterations. The distance function θ calculates a distance between the neurons of the grid. Common choices for θ is to simply return one for neurons close to the BMU and zero for the remaining ones. Another alternative is to use a Gaussian distance function. For all our examples a Gaussian distance function was used.

To obtain the self-organizing map the entire training data is often iterated over multiple times. After training the weights of the neurons have moved closer to the input points. The result can be visualised using a U-matrix. The U-matrix displays the grid of the neurons. The neurons are colored on a grayscale where neurons that lie close to their neighbours are darker. And neurons which are more distant to their neighbours have a lighter color.

It is also possible to directly plot the weight vectors in the input space. In our examples we have plotted both the U-matrix and the weight vectors in the input space. The actual calculations were done using the established implementation MiniSom [6].

We use the non-separable data from last section and apply the self-organizing map using a 5x5 neuron grid to give an example of the method. The result can be see in Fig. 7.

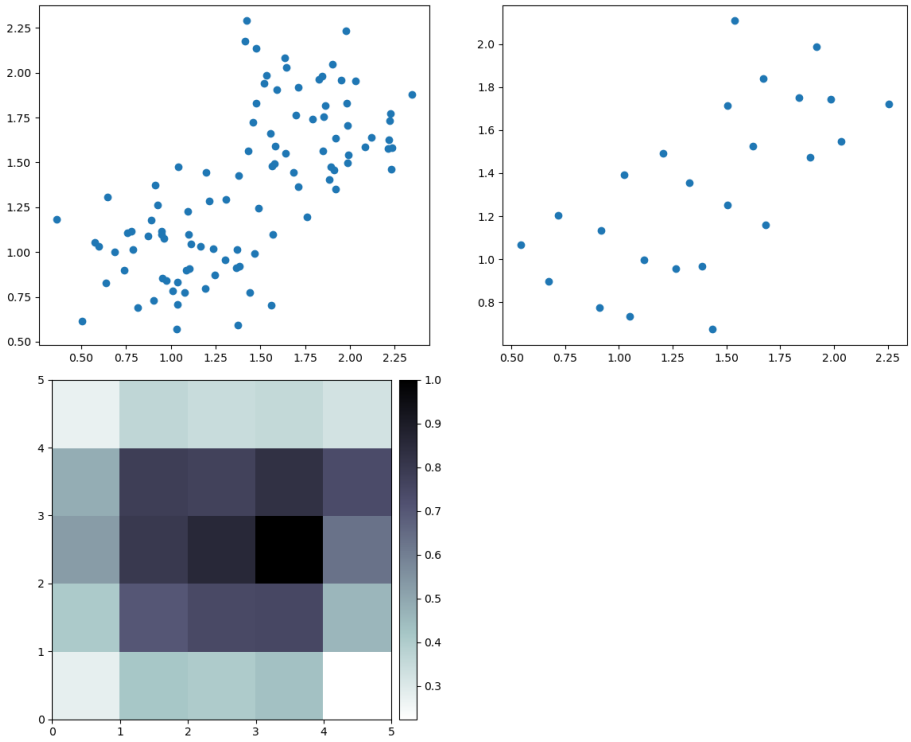

Figure 7: Example of applying a self-organized map to a dataset. To the top left is the generated non-separable data from the last section shown again for reference. The upper right plot shows the weight vectors of the obtained self-organized map in the input space. On the bottom left we plotted the U-matrix of the self-organized map. In the U-matrix each neuron is represented by a square that is colored according to the average distance to its neighbours.

Here we see that the weight vectors of the self-organized map have spread out evenly over the input data. In the U-matrix we can see that the neurons at the center of the grid are located closer to their neighbours than the neurons at the edges.

A potential drawback of using a self-organized map is that it requires the specification of multiple parameters. The grid layout and size, learning rate function α and distance function θ must be specified. This can lead to results that are harder to objectively evaluate.

5.3 Multiple spread out clusters

The next dataset [1] shown in Fig. 8 consists of seven clusters. Four of the clusters are connected and three are completely separated from the others. The optimal manifold method was performed using 200 cluster centers and with a λ value of five.

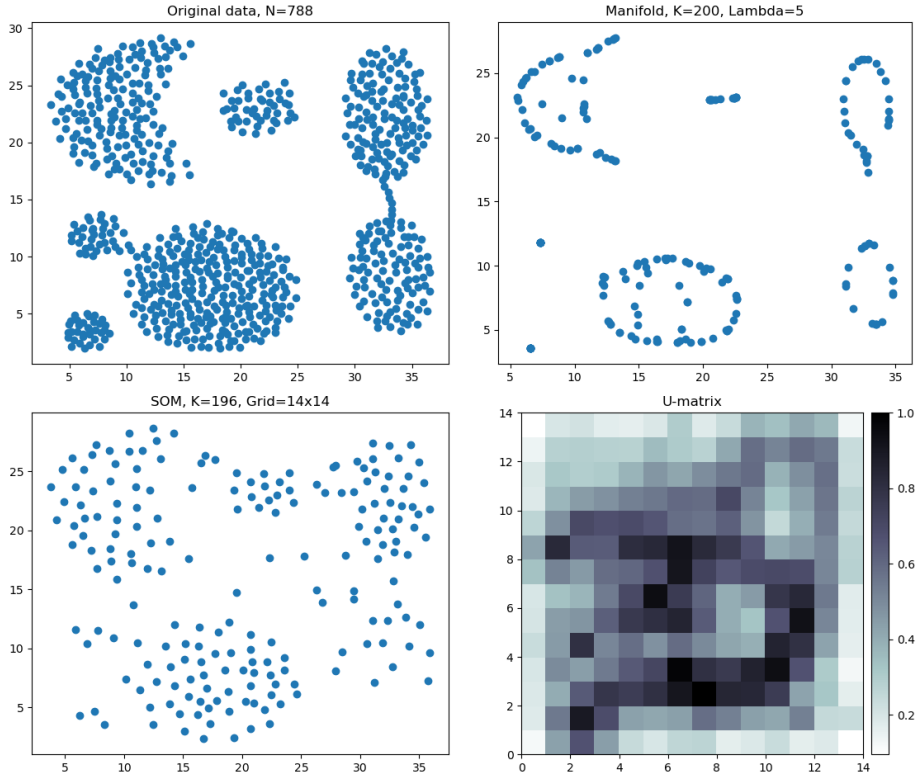

Figure 8: Comparison of the optimal manifold method and a self-organized map for clustering. To the top left the 788 points of the input data are plotted. The input data consists of seven clusters of which four are connected. The result of running the optimal manifold method can be seen to the top right. To the bottom left is the result of applying a self-organized map and to the bottom right is the associated U-matrix.

From the cluster centers it is no longer possible to see that four of the clusters were connected. The two smallest clusters of the input data has been reduced to what seems to be only a single cluster center which is actually multiple cluster centers that have collapsed on top of each other. Once again we can expect this to be a result of the chosen λ value. Choosing a smaller λ value might

make the cluster centers stay also separated for the smaller clusters. For the four larger clusters the method does a better job of capturing the underlying structure. Here the cluster centers are placed in a shape indicating the shapes of the clusters.

On the bottom row is the result of training a self-organizing map using a grid of size $14 \times 14 = 196$ which gives approximately the same number of neurons as the 200 cluster centers of the optimal manifold. The weight vectors appear in a structure similar to the input data but cluster structures are less clearly defined. It does not appear that the method has captured the underlying structure. From the U-matrix we can potentially see a couple of regions of more closely connected neurons. Still it is not obvious from only the U-matrix how many clusters exists in the input data or how they were connected.

6 Manifold learning with the optimal manifold method

The main motivation of the optimal manifold method was to perform dimensionality reduction of the data and capture the structure of the underlying manifold. To evaluate the methods performance we begin by recreating an example from the original paper [5].

6.1 Two-dimensional data

We create a dataset consisting of 3150 points uniformly scattered around a semi-circle of radius 20 and with a Gaussian noise of variance $\sigma^2 = 1$ added. Using this data we run the algorithm with λ set to 8 and using 100 cluster centers, or in the viewpoint that we are looking for an underlying manifold, 100 manifold points. Setting λ to 8 and using the result from Eq. (59), we see that this results in the majority of the weight being placed on data points within a radius of two units from the cluster centers.

From Fig. 9, we see that the manifold points successfully capture the underlying one dimensional manifold where the data points are generated from.

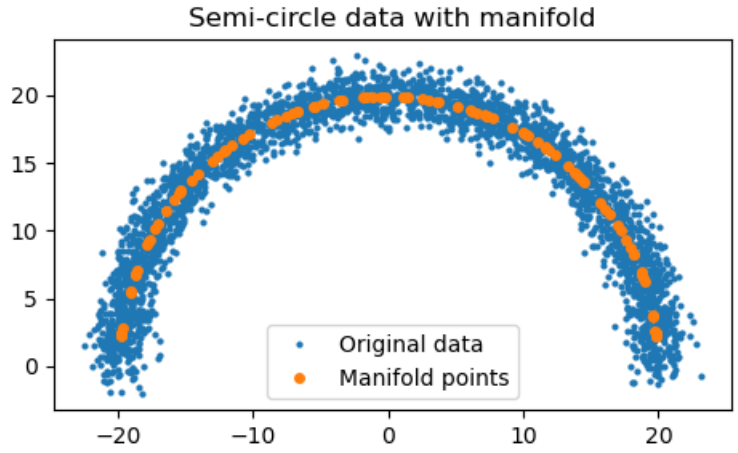

Figure 9: Example of using the optimal manifold method on the semi-circle data. The plot shows both the input data and the manifold points from running the optimal manifold method. The 3150 points of the input data are displayed as the smaller blue points spread out in a semi-circle shape. The 100 manifold points are drawn on top as larger orange points.

6.2 Three-dimensional swiss roll

To see how the method performs on a three-dimensional dataset, we apply the optimal manifold method and the self-organized map to a swiss roll dataset as can be seen in Fig. 10. The swiss roll is a common example used to evaluate the performance of dimensionality reduction methods where the purpose is to see how well different methods can unroll the data. But here we are mainly interested in seeing how well the methods can identify the underlying manifold.

The optimal manifold method was run using 500 manifold points and with a λ value of one. This is roughly equivalent to a standard deviation around the cluster centers of $\sqrt{1/2} = 0.7$. From Fig. 10, method manages to clearly separate the swirls of the swiss roll and place the manifold points on the two-dimensional underlying manifold.

We can compare with the result from the self-organized map shown in Fig. 10. The outline of the manifold can clearly be seen but the self-organized map does not manage to separate the swirls from each other. Instead there are weight vectors spread out between the swirls of the swiss roll. We also notice that there are more weight vectors in the places where the swirls are closer together. Turning our attention to the U-matrix, we can see that there are two darker spots of possible clusters but otherwise there is no clear pattern or obvious interpretation from the U-matrix.

In general, it seemed to be the case that the optimal manifold method gave consistent looking results when run repeatedly, using different seed values for the random number generator used for initialization. In contrast both the weight vectors and the U-matrix of the self-organized map had a more erratic behaviour between runs with consistent problems of separating the data appropriately.

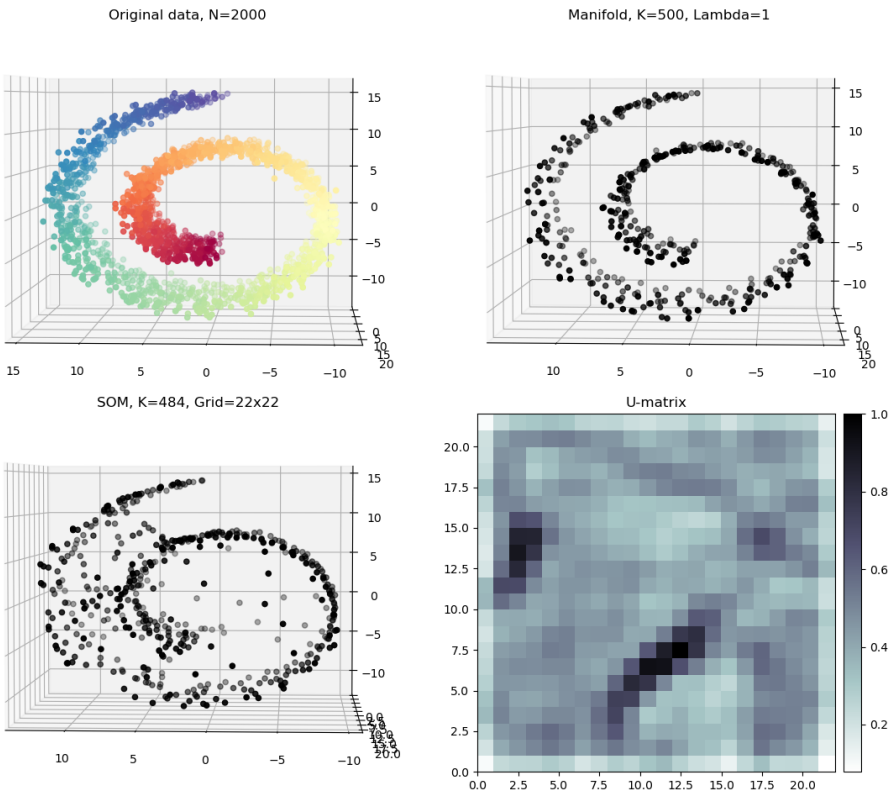

Figure 10: Comparison of using the optimal manifold method and a self-organized map on a three-dimensional swiss roll dataset. All the plots except the U-matrix are three-dimensional with the swiss roll shown from the side. The input data to the top left was generated using 2000 points with added normal noise. The input data points are plotted using a color scale for better visualization. The other plots are in grayscale. The optimal manifold method was done using 500 manifold points with a λ value of one and can be seen to the top right. The self-organized map is in the bottom left. It was obtained using a grid size of 22×22 neurons to give a number of neurons, 484, roughly equivalent to the number of manifold points. To the bottom right is the U-matrix of the self-organized map.

6.3 Three-dimensional s-shape

The original paper [5] had one example of a three-dimensional dataset which we have here chosen to recreate mainly for completeness. In this case only the optimal manifold method was run but not the self-organized map, as the underlying challenges and analysis of the dataset are very similar to that of the

swiss roll dataset.

The dataset uses 2000 data points generated on a three-dimensional s-shape with added Gaussian noise. On the top of Fig. 11 we see the original data points and the optimal manifold in three-dimensions and on the bottom their two-dimensional projections. We note that the manifold points seem to fall on the underlying two-dimensional manifold as was the case with the swiss roll.

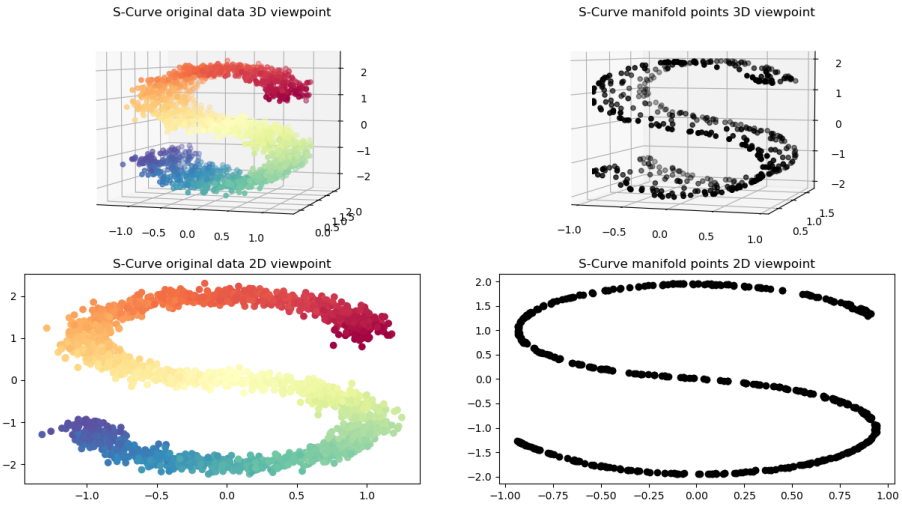

Figure 11: Three-dimensional s-shaped dataset and the result of using the optimal manifold method on it. The input data is shown to the top left in color for better visualization. The manifold points from the optimal manifold method are shown to the top right. The bottom row shows the two-dimensional projections.

6.4 One-dimensional manifolds of different width

The next dataset consists [3] of two different, one-dimensional manifolds, that have different widths and can be seen in Fig. 12. In addition to having different widths the bottom-right manifold consists of more data points which are more densely packed.

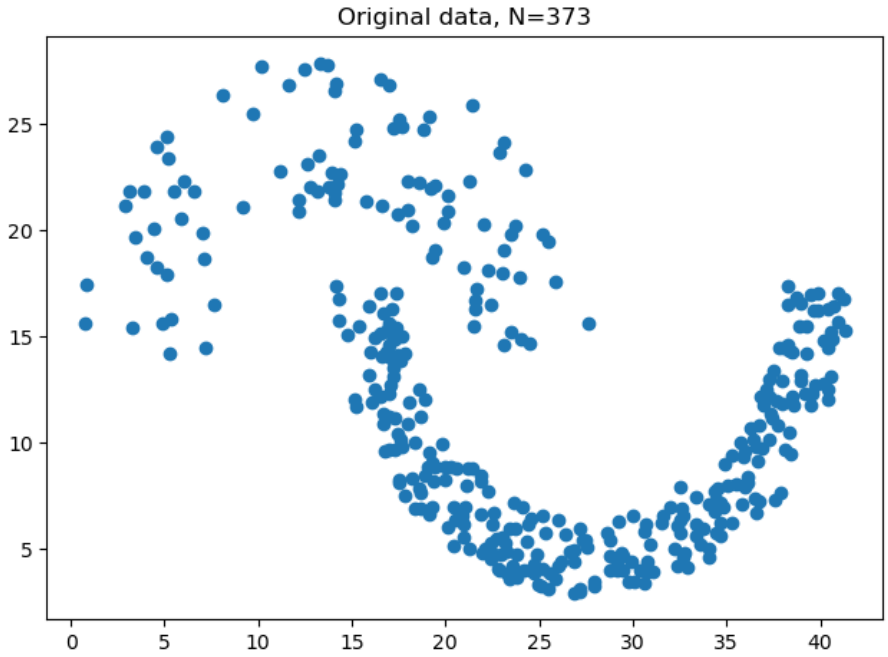

Figure 12: Two manifolds of different width, consisting of 373 points.

Running the optimal manifold method using 100 manifold points and setting λ to three and 10 give the result of Fig. 13. With a λ value of three the structure of the thinner manifold is captured well by the manifold points. In comparison the thicker manifold still has the manifold points spread out in the two-dimensional space. Having λ set to three therefore set an appropriate scale to capture the thinner manifold but not the thicker one.

When increasing λ to ten, the thinner manifold keeps its overall structure even though many manifold points now collapse into each other and the manifold points thus appear to be further apart. The same phenomenon can be seen for the thicker manifold which now has less of a two-dimensional structure. Even so the underlying structure is not obvious from the manifold points of the thicker manifold. The problem to find a smooth set of manifold points for the thicker manifold could be due to the small number of data points belonging to it.

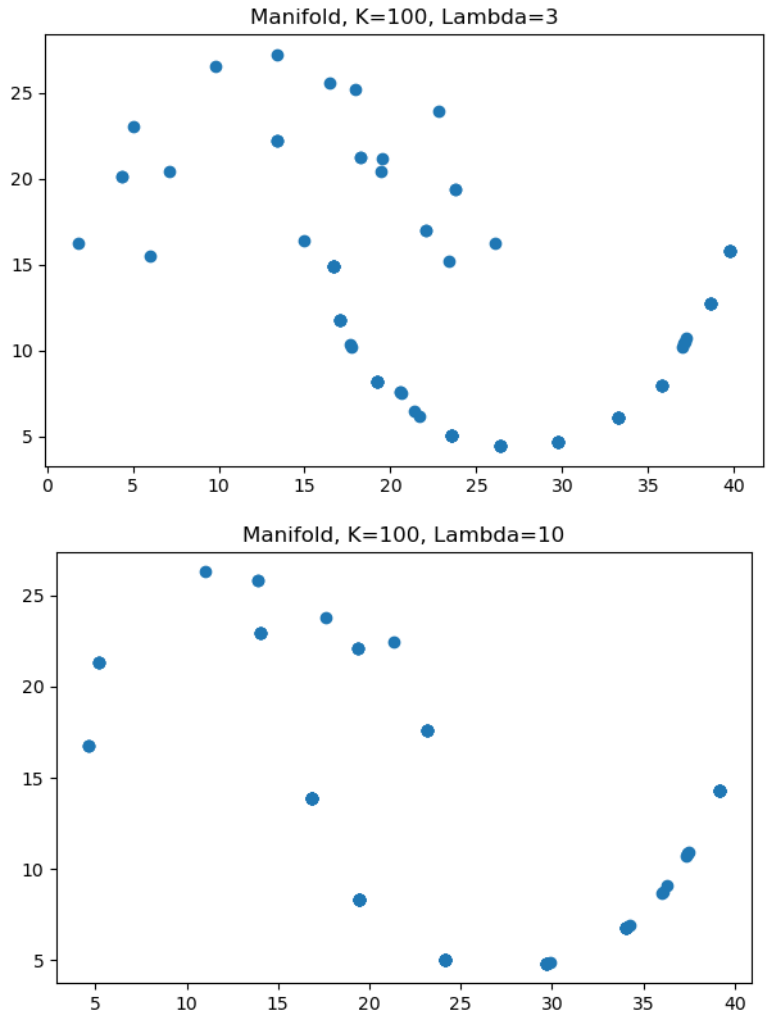

Figure 13: Result of using the optimal manifold method. In both cases 100 manifold points were used. In the top plot a λ value of three was used and in the bottom plot a λ value of ten. For both λ values the manifold points seem to capture the structure of the thicker manifold but struggle on the thinner manifold.

In Fig. 14 we plot the result of applying a self-organized map. The outcome is a self-organized map where the weight vectors are evenly spread out on the input points. We can see that there are more weight vectors in the more densely populated manifold. But the self-organized map once again has a problem with properly separating the clusters with weight-vectors being spread out in between

the clusters. Nevertheless, the two separate clusters can be detected by using the visualization of the U-matrix.

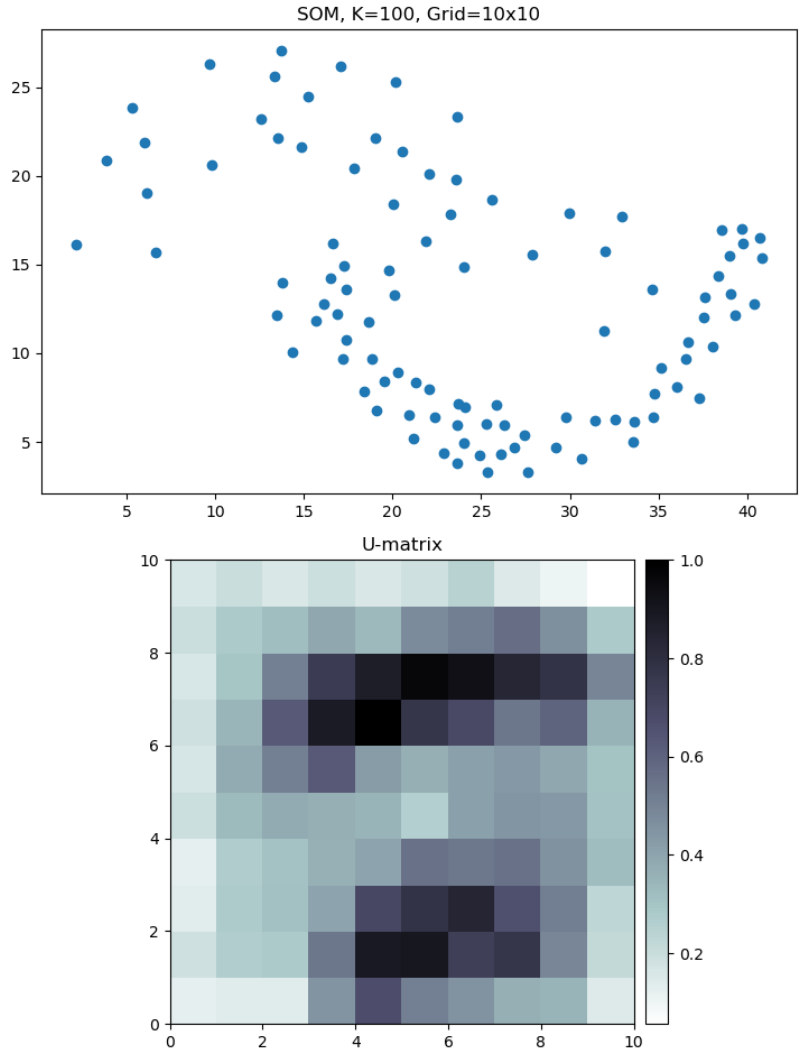

Figure 14: The result of applying a self-organized map to two manifolds of different widths. The self-organized map was obtained using a grid size of 10×10, for a total of 100 neurons.

7 Dimensionality estimation and fractal dimensions

Often it is of interest to know the underlying dimensionality of the data we are working with. This can be of direct interest to better understand the data. In addition, it is often a required parameter for dimensionality reduction methods. For this purpose we need a mathematical definition of dimension. A natural viewpoint of dimension is that of degrees of freedom. By this viewpoint the dimension of a surface is the number of variables needed to specify the location of a point on the surface. This works for simple objects like a point, line, area or volume. Unfortunately the existence of objects like the Peano curve which is a line that can fill up a two-dimensional space makes this definition intractable.

Instead the notion of fractal dimension can be used which is more well defined. Suppose that we are interested in knowing the length of the coastline of England. If we take a map showing the coastline, one could imagine simply straightening out the coastline of the map and measuring it. Taking this measurement and accounting for the scale of the map would get us an approximation of the length.

Of course the map just gives a representation of the coastline. If we instead went out and measured the coastline with a ruler, we would get a much finer measurement which would turn out to be longer. Fractals are objects like coastlines for which there exists a finer level of detail the closer we zoom in. Furthermore it is not possible to stretch out fractals in such a way that we can measure the arc length of them.

The dimension of a fractal is measured with its fractal dimension. While our informal notion of degrees of freedom only makes sense with dimensions being integers, fractal dimensions are real non-negative numbers. Our normal notion of a line has a fractal dimension of one. An object like the Peano curve that fills up a two-dimensional space has a fractal dimension of two. But one can also define fractal curves which has a dimension between one and two.

The fractal dimension can be understood as how much the detail of a shape changes with the length by which it is measured. If we measure a line of length one, with a measuring stick also of length one, this requires one measurement. If we measure the same line with a measuring stick that is of length one-third, we need three measurements. We can capture this relationship by the following equation

$$N = \epsilon^{-D} \tag{65}$$

Here N is the number of measurements, ϵ the length of the measuring stick and D the dimension of the shape.

To continue the example in two dimensions we can imagine measuring a unit

square. If we do this using another unit square, it would require one measurement. But if we instead made the measurements using a square with a side length of one-third we would need $(1/3)^2 = 9$ measurements. These relationships do not necessarily hold for fractals. For a fractal using a measuring stick which is a third of the length may give a measurement which is four times as long.

There are multiple definitions of fractal dimension. They all have in common that they measure how much the details of the data changes when the scale at which we look at the data varys. We will briefly describe the capacity-, information- and correlation-dimension before demonstrating how the correlation dimension can be used to estimate dimension.

7.1 Capacity dimension

The capacity dimension, also known as box-counting dimension, can be understood as the change in the number of boxes needed to cover the data when we decrease the side length of the boxes. An illustration of this can be seen in Fig. 15. In the figure boxes have been drawn to cover the entire coastline. As the size of the boxes decreases, the number of boxes necessary to cover the coastline increases.

We denote the number of boxes needed to cover the object, using boxes with side length ϵ, as $N(\epsilon)$. Solving Eq. (65) for D we get

$$N(\epsilon) = \epsilon^{-D} \iff \log N(\epsilon) = -D \log \epsilon \tag{66}$$

$$\iff D = -\frac{\log N(\epsilon)}{\log \epsilon} \tag{67}$$

The capacity dimension is defined as the limit we get when we let the side length ϵ approach zero

$$D_{cap} = \lim_{\epsilon \to 0} -\frac{\log N(\epsilon)}{\log \epsilon} \tag{68}$$

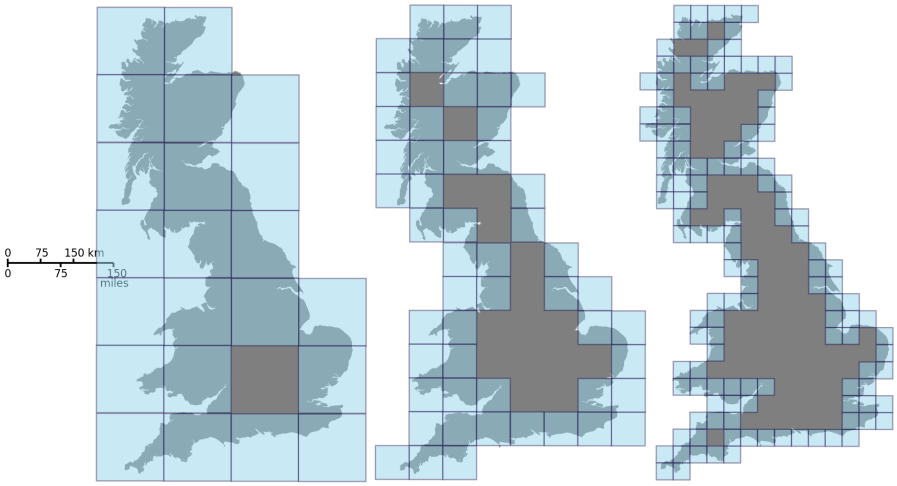

Figure 15: Illustration of how the capacity dimension can be calculated on the coastline of England [9]. The left map covers the coastline using the largest boxes. The coastline of the middle and the right map are covered using smaller boxes, which results in more boxes having to be used.

7.2 Information dimension

The information dimension is the fractal dimension of a probability distribution. Let $N(\epsilon)$ be the number of boxes needed to cover the support of the distribution. And let p_i denote the probability that box i is occupied. We then define the information dimension as

$$D_{inf} = \lim_{\epsilon \to 0} \frac{\sum_{i=1}^{N(\epsilon)} p_i \log p_i}{\log \epsilon} \tag{69}$$

The name information dimension comes from the similarity between the numerator and the definition of entropy.

In practice the information dimension is often impossible to calculate with finite sample sizes since it is difficult to estimate the probabilities of the boxes being occupied. Under the simplified assumption that the boxes have equal probability of being occupied, the information dimension can be shown to equal the capacity dimension [11].

7.3 Correlation dimension

The correlation dimension is one of the fractal dimensions discussed and it is often the easiest one to estimate in practice. The idea is that for a sphere placed on a manifold of dimension d, as we let the sphere radius ϵ increase, the number of points within the sphere will increase proportionally to ϵ^d. Thus the number of points will grow quicker for manifolds of higher dimensions. To calculate the correlation dimension we first define the discretized correlation integral

$$C(\epsilon) = \lim_{N\to\infty} \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j<i} H(\epsilon - ||x_i - x_j||) \tag{70}$$

Here H is a step function such that $H(x) = 1$ for $x \geq 0$, and $H(x) = 0$ when $x < 0$. The norm used is the Euclidean norm. The result is to estimate the probability that any two points lie within a distance of ϵ. The correlation dimension is defined as

$$D_{corr} = \lim_{\epsilon\to 0} \frac{\log C(\epsilon)}{\log \epsilon} \tag{71}$$

This can be visualized by calculating the correlation sum and plotting it against the distance ϵ on a log-log plot. The slope of the plot then coincides with the correlation dimension. The correlation dimension is often used because it can easily be estimated.

7.4 Dimensionality estimation using the correlation dimension

In the optimal manifold paper [5], the correlation dimension was used to estimate the dimension of the semi-circle data from section 6.1. First the optimal manifold method was run on the semi-circle data to generate a set of manifold points. Then the correlation sum was calculated for a range of ϵ values, both on the original data points and on the manifold points. This was then plotted on a log-log scale. For reference the plots of the original data and the manifold points are shown again in Fig. 16.

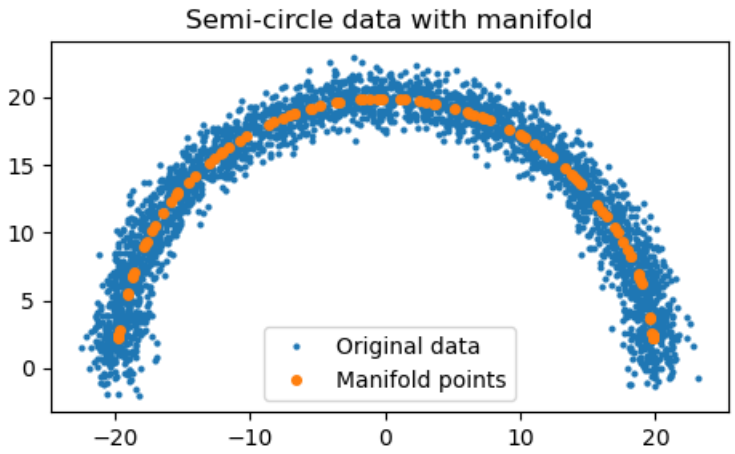

Figure 16: Semi-circle data with manifold points plotted on top. Shown here again for reference.

The log-log plot of the correlation sum plotted against ϵ can be seen in Fig. 17. Notice that from around $\log(\epsilon) = 1$ the points are placed on top of each other.

The manifold points have a constant slope of 0.88 and the correlation dimension remains fixed. For the original data points the slope decreases with increasing radius. As the radius of the theoretical sphere increases the correlation dimension of the original data approaches that of the manifold data and the slopes start to coincide.

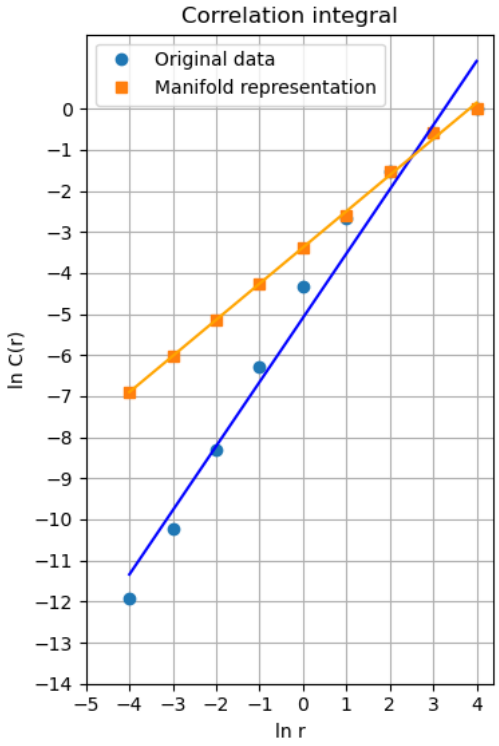

Figure 17: Correlation sum $C(\epsilon)$ plotted against radius ϵ on a log-log scale for the semi-circle data. The blue points show the result for the original data and the orange points the result for the manifold data. The drawn lines show the average slopes which are 1.56 for the original data and 0.88 for the manifold data.

8 Discussion

We have demonstrated the performance of the optimal manifold method on a variety of datasets both for clustering and to find an underlying manifold. The main purpose of the method as introduced is to identify underlying manifolds. In the examples given, the method is indeed able to do so. The produced manifold points fall on the underlying manifold, with the noise of the data being reduced.

To produce these results, the only parameter that had to be tuned was the λ parameter which can be interpreted as the scale of the lens through which we view the data. As previously discussed, a good starting points for plausible λ values can be found by running the method over a range of λ values to produce a rate-distortion curve. From this we can pick λ values where the manifold seems well behaved. For all the examples shown the data was either two- or three-dimensional which made understanding and verifying the results straightforward. In general with data of higher dimensions, tuning the λ parameter and interpreting the results could be harder. Understanding these difficulties would require further studies.

The method as presented does not produce a direct mapping to a lower dimensional space. To get a mapping that reduces the dimensionality of the data requires further steps to be performed on the output of the method. Many alternatives for doing this exists. One alternative would be to apply a self-organized map on the manifold points produced from the optimal manifold method. This would then produce an output which has the same dimension as the grid of the self-organized map. Exploring the possible ways in which the optimal manifold method could be combined with other methods to perform dimensionality reduction is another potential area of study.

Finally it is worth mentioning that there exists other information theoretic approaches for performing dimensionality reduction. The optimal manifold method is based on rate-distortion theory and uses a form of clustering to find the underlying manifold. Other proposed methods [2] instead take a more direct approach and attempts to arrive at an explicit functional mapping between the input space of the data and a lower dimensional space.

Appendix

Functional derivatives

In order to arrive at the self-consistent Eq. (54) and Eq. (55) which are used to calculate the rate-distortion function we need to take the derivatives of the functional with respect to $\gamma(t)$ and $p(t|x)$.

The full functional has the form

$$\mathcal{F} = D + \lambda I + \iint dx'dt'\lambda_0(x')p(t'|x')$$

where

$$D = \iint dx'dt'p(x')p(t'|x')||x' - \gamma(t')||^2$$
$$I = \iint dx'dt'p(x')p(t'|x')\log\frac{p(t'|x')}{p(t')}$$

and the last term of the functional is the Lagrange multiplier.

Derivative with respect to gamma

We begin by taking the component-wise derivative of the functional with respect to $\gamma(t)$. In doing so we will use the following properties of the Dirac delta function δ

$$\frac{\partial f(t')}{\partial f(t)} = \delta(t' - t)$$

$$\int dx'\delta(x - x')f(x') = f(x)$$

And we denote the Kronecker delta by δ_{ij}

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & i \neq j \end{cases}$$

We proceed with taking the derivative.

$$\begin{aligned}
\frac{\partial \mathcal{F}}{\partial \gamma_i(t)} &= \frac{\partial}{\partial \gamma_i(t)} \left[D + \lambda I + \iint dx' dt' \lambda_0(x') p(t'|x') \right] \\
&= \frac{\partial}{\partial \gamma_i(t)} D \\
&= \frac{\partial}{\partial \gamma_i(t)} \iint dx' dt' p(x') p(t'|x') ||x' - \gamma(t')||^2 \\
&= \iint dx' dt' p(x') p(t'|x') \frac{\partial}{\partial \gamma_i(t)} \sum_j (x'_j - \gamma_j(t'))^2 \\
&= \iint dx' dt' p(x') p(t'|x') \sum_j 2(x'_j - \gamma_j(t')) \delta_{ij} \frac{\partial \gamma_j(t')}{\partial \gamma_i(t)} \\
&= 2 \iint dx' dt' p(x') p(t'|x') (x'_i - \gamma_i(t')) \frac{\partial \gamma_i(t')}{\partial \gamma_i(t)} \\
&= 2 \iint dx' dt' p(x') p(t'|x') (x'_i - \gamma_i(t')) \delta(t' - t) \\
&= 2 \int dx' p(x') p(t|x') (x'_i - \gamma_i(t))
\end{aligned}$$

Setting the derivative equal to zero and solving for $\gamma_i(t)$ gives

$$\begin{aligned}
0 &= 2 \int dx' p(x') p(t|x') (x'_i - \gamma_i(t)) \\
\gamma_i(t) &= \frac{\int dx' p(x') p(t|x') x'_i}{\int dx' p(x') p(t|x')} \\
\gamma_i(t) &= \frac{1}{p(t)} \int dx' p(x') p(t|x') x'_i
\end{aligned}$$

which gives the components of the sought form. From the derivative of the components we arrive at the full derivative

$$\gamma(t) = \frac{1}{p(t)} \int dx' p(x') p(t|x') x'$$

which corresponds to Eq. 54.

Derivative with respect to the conditional probability

Next we will take the derivative of $\mathcal{F}$ with respect to $p(t|x)$.

$$\frac{\partial \mathcal{F}}{\partial p(t|x)} = \frac{\partial}{\partial p(t|x)} \left[D + \lambda I + \iint dx' dt' \lambda_0(x') p(t'|x') \right]$$

The derivative of the first and third term follows immediately

$$\begin{aligned}\frac{\partial}{\partial p(t|x)}D &= \frac{\partial}{\partial p(t|x)}\iint dx'dt'p(x')p(t'|x')||x'-\gamma(t')||^2 \\ &= \iint dx'dt'p(x')\frac{\partial p(t'|x')}{\partial p(t|x)}||x'-\gamma(t')||^2 \\ &= \iint dx'dt'p(x')\delta(t'-t)\delta(x'-x)||x'-\gamma(t')||^2 \\ &= p(x)||x-\gamma(t)||^2\end{aligned}$$

and similarly

$$\frac{\partial}{\partial p(t|x)}\iint dx'dt'\lambda_0(x')p(t'|x') = \lambda_0(x)$$

What remains is the derivative with respect to the information term

$$\begin{aligned}&\frac{\partial}{\partial p(t|x)}I \\ &= \frac{\partial}{\partial p(t|x)}\iint dx'dt'p(x')p(t'|x')\log\frac{p(t'|x')}{p(t')} \\ &= \iint dx'dt'p(x')\left[\delta(t'-t)\delta(x'-x)\log\frac{p(t'|x')}{p(t')} + p(t'|x')\frac{p(t')}{p(t'|x')}\frac{\partial}{\partial p(t|x)}\frac{p(t'|x')}{p(t')}\right] \\ &= p(x)\log\frac{p(t|x)}{p(t)} + \iint dx'dt'p(x')p(t')\left[\frac{1}{p(t')}\frac{\partial p(t'|x')}{\partial p(t|x)} + p(t'|x')\frac{\partial p(t')^{-1}}{\partial p(t|x)}\right] \\ &= p(x)\log\frac{p(t|x)}{p(t)} + p(x) + \iint dx'dt'p(x')p(t')p(t'|x')(-1)(p(t'))^{-2}\frac{\partial p(t')}{\partial p(t|x)} \\ &= p(x)\log\frac{p(t|x)}{p(t)} + p(x) - \iint dx'dt'\frac{p(x')}{p(t')}p(t'|x')\frac{\partial\int dx''p(t'|x'')p(x'')}{\partial p(t|x)} \\ &= p(x)\log\frac{p(t|x)}{p(t)} + p(x) - \iint dx'dt'\left[\frac{p(x')}{p(t')}p(t'|x')\int dx''\delta(t'-t)\delta(x''-x)p(x'')\right] \\ &= p(x)\log\frac{p(t|x)}{p(t)} + p(x) - \iint dx'dt'\frac{p(x')}{p(t')}p(t'|x')\delta(t'-t)p(x) \\ &= p(x)\log\frac{p(t|x)}{p(t)} + p(x) - \frac{p(x)}{p(t)}\int dx'p(t|x')p(x') \\ &= p(x)\log\frac{p(t|x)}{p(t)} + p(x) - \frac{p(x)}{p(t)}p(t) \\ &= p(x)\log\frac{p(t|x)}{p(t)}\end{aligned}$$

We proceed to set the derivative of the entire functional to zero and solve for $p(t|x)$

$$p(x)||x-\gamma(y)||^2 + \lambda p(x)\log\frac{p(t|x)}{p(t)} + \lambda_0(x) = 0$$
$$\iff$$
$$\lambda p(x)\log p(t|x) - \lambda p(x)\log p(t) = -\lambda_0(x) - p(x)||x-\gamma(t)||^2$$
$$\iff$$
$$\lambda p(x)\log p(t|x) = \lambda p(x)\log p(t) - \lambda_0(x) - p(x)||x-\gamma(t)||^2$$
$$\iff$$
$$\log p(t|x) = \log p(t) - \frac{\lambda_0(x)}{\lambda p(x)} - \frac{1}{\lambda}||x-\gamma(t)||^2$$
$$\iff$$
$$p(t|x) = \frac{p(t)}{e^{\frac{\lambda_0(x)}{\lambda p(x)}}} e^{-\frac{1}{\lambda}||x-\gamma(t)||^2}$$

Which for an appropriately chosen $\lambda_0(x)$ equals Eq. 55.

References

[1] Gionis A, Mannila H, and Tsaparas P. Clustering aggregation. *Acm transactions on knowledge discovery from data*, 1(1), 2007.

[2] Globerson A and Tishby N. Sufficient dimensionality reduction. *Journal of Machine Learning Research*, 3(Mar):1307–1331, 2003.

[3] Jain A and Law M. Data clustering: A user's dilemma. *Lecture Notes in Computer Science*, pages 1–10, 2005.

[4] Shannon C E. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[5] Chigirev D and Bialek W. Optimal manifold representation of data: an information theoretic approach. *Advances in Neural Information Processing Systems*, 16:161–168, 2004.

[6] Vettigli G. Minisom: minimalistic and numpy-based implementation of the self organizing map, 2018.

[7] Hotelling H. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[8] Lee J A and Verleysen M. *Nonlinear Dimensionality Reduction*, chapter 5.2, pages 135–143. Springer, 2007.

[9] Prokofiev. Estimating the box-counting dimension of the coast of great britain, distributed under cc by-sa 3.0 license, 2021.

[10] Hartley R VL. Transmission of information 1. *Bell System technical journal*, 7(3):535–563, 1928.

[11] Cover T M and Thomas J A. *Elements of Information Theory*. John Wiley and Sons, Inc, 2 edition, 2006.