ERASMUS UNIVERSITEIT ROTTERDAM

Erasmus School of Economics

ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

July 5, 2020

Data Visualization using t-SNE and Local Multidimensional Scaling

---

A Comparison and Application to Country Similarities

Bachelor Thesis: BSc2 Econometrics and Economics

Supervisor :

Prof. dr. P.J.F Groenen

Joppe de Bruin (450079)

Second Assessor :

A. Castelein

Abstract

This paper aims to compare two highly popular data visualization techniques for high dimensional data, t-SNE and MDS, since the seminal work on t-SNE (Van Der Maaten & Hinton, 2008) and later literature lack a comparison between t-SNE and the broader field of MDS methods. The focus on retaining the local structure of the high dimensional data by t-SNE is introduced in MDS by considering stress based MDS with weights (De Leeuw & Heiser, 1980). Moreover, Local MDS (Chen & Buja, 2009) is introduced to recreate the clustering characteristic of t-SNE. The two approaches are tested on three data sets: the MNIST data, a simulated data set and data on country characteristics. It is found that t-SNE overall is superior to the MDS implementations in retaining the local structure of the data. However, Local MDS is also able to show a clear cluster structure, which was not found in the t-SNE literature until now.

Contents

1 Introduction

In the scientific fields including but not limited to international-, policy- and development economics much policy advice is provided on a country specific level.This does not mean, however, that policies are not completely unsuited to be generalized and used in different situations. One of the main challenges in these fields is that policy areas cannot be considered to function in a economic vacuum. Therefore, the generalization of economic policies designed for a single country cannot be performed solely on the basis of a comparison between the countries on the specific area at which the policy was targeted. Many researchers (a.o. Saggi, Maskus, and Hoekman (2004)), aim to circumvent this problem by dividing countries in certain groups, based on for example their national income level. Even though Saggi et al. (2004) realize that this is not optimal, they use these categorizations to propose policy ideas to groups of countries. Due to the intertwined nature of the economy within countries, such one dimensional cross country comparisons are limited in their power. A similarity in one aspect of the economic system between two countries might be offset by a difference in another area, potentially yielding surprising outcomes. As a consequence of the vast amount of data that is collected by (economic) organizations around the world (e.g. IMF, World Bank, United Nations), data sets that consist of a broad spectrum of country characteristics can be constructed. Anderson and Hussey (2001), for example, compare the health system in OECD countries on the basis of a wide variety of metrics, ranging from immunization rates to the number of hospital beds per capita. These ubiquitous data sets theoretically allow researchers to compare countries on a high number of characteristics, ranging from economic indicators to freedom scores and quality of government to anthropological factors. However, the mere size and dimensionality of these data sets introduce a fundamental difficulty in gaining an intuitive understanding of the data (Van Der Maaten, Postma, & Van den Herik, 2009). In this type of situations, there is a strong desire to be able to obtain some intuition in the structure of the data. The fields of dimension reduction and data visualization aim to provide solutions to these problems.

Dimension reduction techniques aim to represent a high dimensional data set in a lower, easier to handle dimensionality. This field of research was partly developed to solve some inherent challenges in the world of statistics, data science and computer science, since many models experience trouble if the dimensionality of the data becomes too high (Bellman & Dreyfus, 1962). While this composes an interesting field of this research, this paper focuses on another important goal of dimension reduction: data visualization.

As was hinted upon earlier, one of the disadvantages of high dimensional data is the inherent difficulty to gain intuitive insights in the data structure. For a large part, this is caused by the limitations of humans to visualize data in maximally three dimensions. In image and textual analysis, for example, objects are often represented by thousands of dimensions (Van Der Maaten et al., 2009). Over the last decades, the field of data visualization has developed a broad scope of powerful methods and algorithms to map high dimensional data onto a 2D or 3D map that aids in gaining an understanding in the structure of the data.

In particular, this paper will focus on t-Distributed Stochastic Neighbour Embedding (t-SNE) (Van Der Maaten & Hinton, 2008), a highly popular method of visualizing high dimensional data. This method has gained enormous popularity mainly due to its superior ability to map data points in such a way that clear clusters in the data become apparent. Introduced as an improvement over the less successful Stochastic Neighbour Embedding (SNE) method (Hinton & Roweis, 2003), it frequently outperforms a wide array of other techniques, such as Principal Component Analysis, Sammon Mapping and Isomap (Van Der Maaten & Hinton, 2008). The main advantage is the ability of t-SNE to clearly separate clusters of points in the low dimensionality, whereas other methods have trouble letting clusters drift apart. Even though the seminal work by Van Der Maaten and Hinton (2008) compares the method to a wide variety of methods, its specific relationship with the more general defined methods of non-linear multidimensional scaling (MDS) is not investigated. As will be discussed in a later stage of this paper, t-SNE was designed to focus on modelling the local structure of data sets. Van Der Maaten and Hinton (2008) argue that the way in which this is approached, sets the method apart from other methods. Contrarily, Groenen and Van De Velden (2016) argue that by tweaking some weight parameters, MDS can be instructed also focus on the local structure. Moreover, Chen and Buja (2009) developed a variant of MDS (Local MDS), which shows similarities with t-SNE. This paper hopes to shed some light on this particular relationship between t-SNE and non-linear MDS, as this is still underdeveloped in the literature.

Concretely, t-SNE has been applied to a broad scope of fields, ranging from genetic data (Li, Cerise, Yang, & Han, 2017) to computer security (Gashi, Stankovic, Leita, & Thonnard, 2009), image analysis (Gisbrecht, Schulz, & Hammer, 2015) and musical analysis (Hamel & Eck, 2010). Despite its universal applicability and strong performance, the above-mentioned list primarily focuses on natural and computer sciences. Social sciences, such as psychology and economics have not seen a prominent rise of t-SNE based visualizations of data sets. Reflecting on the first paragraph of this paper, much multilateral policy advise hinges on the comparison of countries and formulat-

ing similarities between them. t-SNE and MDS are therefore potentially powerful methods in an international policy adviser's toolbox.

As such, this paper aims to introduce t-SNE in this field and investigate the power and usefulness of the method in comparison to the widely used method of MDS based on stress functions. Consequently, the main research question this paper aims to address is: "How does the performance of t-SNE compare to MDS in the field of visualizing high dimensional country similarities?"

In particular, a high dimensional data set of country characteristics (e.g. GDP, level of corruption, economic freedom, etc.), obtained from the World Government Summit 2019, will be used to investigate this. For many researchers in the field of (international) policy economics, it is interesting to see whether t-SNE and MDS can be used as a powerful tool to visualize country similarities. Especially with regards to Van Der Maaten and Hinton's (2008) observation that other visualization techniques have trouble to let clusters of similar points drift apart, is of interest in this field. Clearly identifying clusters of countries is helpful in many policy applications. To gain a more complete comparison between the two methods, they will also be applied to the MNIST data set and a simulated data set containing a low dimensional cluster structure.

In summary, this paper aims to fill two distinguishable gaps in the current literature. Firstly, as a concrete and complete comparison between MDS and t-SNE has not been conducted before, it aims to shed light on their comparative performance. Secondly, by applying this comparison on a high dimensional data set of country characteristics, the performance of these visualization techniques will be investigated in a new context. The rest of this paper is structured as follows. Firstly, the two methods of interest, t-SNE and non-linear MDS will be discussed in detail. This review of literature will be focused on the main idea behind the methods. The paper then continues with a Algorithms section, which will elaborate upon the mechanics of the two methods in greater detail. Afterwards, the experiments corresponding to the three data sets are described and their results are presented. Finally, the results are used to answer the research question and the limitations of this paper are discussed.

2 Literature Review

Over the years many powerful visualization and dimension reduction techniques have been developed. In general, these methods aim to map a high dimensional data set $X = \{x_1, x_2, ...x_n\}$ into the two- or three dimensional set $Y = \{y_1, y_2, ..., y_n\}$. The main difficulty of these methods is

the preservation of as much of the information in the data as possible. As was mentioned in the Introduction, this paper focuses on two specific methods of dimension reduction and visualization: t-SNE and multidimensional scaling (MDS). The interested reader can refer to Van Der Maaten et al. (2009), Lee and Verleysen (2007) or Saul, Weinberger, Ham, Sha, and Lee (2006) for a general and complete overview of the field.

2.1 t-SNE

2.1.1 Stochastic Neighbour Embedding

Before the main method of this paper is introduced it is necessary to gain some insight in the ancestor/predecessor of the t-SNE method: Stochastic Neighbour Embedding. This method was introduced by Hinton and Roweis (2003) and forms the basis for the t-SNE method that uses the same framework with some profound deviations. The main idea of SNE is based on the conversion of the high dimensional data to pairwise similarities. Data points that are close in the high dimensional space should have high similarity scores, while points that are far away should have low scores. The SNE algorithm then aims to find a set of points in the low dimensional space and that generates pairwise similarity scores that are as close as possible to the high dimensional similarities.

There are multiple ways to compute the similarity between points, and SNE describes the similarity between two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as the conditional probability $p_{j|i}$, which models the probability that $\boldsymbol{x}_j$ is $\boldsymbol{x}_i$s neighbour. The probability distribution over all points is taken as a Gaussian that is centered at $\boldsymbol{x}_i$, taking the Euclidean distance between a pair of points as its argument. The $p_{j|i}$s are characterized by the following equation:

$$p_{j|i} = \frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\sigma_i)}{\sum_{k\neq i}\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\sigma_i)}, \tag{1}$$

where σ_i denotes the variance of the Gaussian centered at $\boldsymbol{x}_i$. In Section 3.1 the method of determining the value of σ_i is discussed. As was mentioned in the previous paragraph, the low dimensional points are chosen in such a way that their similarities are as close as possible to the high dimensional similarities $p_{j|i}$. The low dimensional similarities $q_{j|i}$ are defined as similarly:

$$q_{j|i} = \frac{\exp(-\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)}{\sum_{k\neq i}\exp(-\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)}, \tag{2}$$

where the variance is set to $1/\sqrt{2}$ without loss of generality and $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ refer to the low dimensional representations of points i and j respectively.

Hinton and Roweis (2003) argue that when the points in the low dimensional space faithfully model the similarity between the points in the high dimensional space, the conditional probabilities $p_{j|i}$ and $q_{j|i}$ should be equal. Therefore they introduce the Kullback-Leibler divergence (Kullback & Leibler, 1951), which is a measure of equality between two probability distributions. The SNE algorithm aims to find the points $\boldsymbol{y}_i$ (with corresponding $q_{j|i}$) that minimize the Kullback-Leibler divergences over all datapoints. The cost function C is defined as follows

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}. \tag{3}$$

Due to the structure of this cost function, the authors argue that SNE constitutes an improvement over previous methods. Analyzing the sum of Kullback-Leibler divergences, it can be shown that modelling a high $p_{j|i}$ with a small $q_{j|i}$ (i.e. modeling two points $\boldsymbol{x}_j$ and $\boldsymbol{x}_i$ that are far apart in a high dimension with two points $\boldsymbol{y}_j$ and $\boldsymbol{y}_i$ that are close in the low dimension) will result in a large loss, while a small $p_{j|i}$ that is modelled by a large $q_{j|i}$ will result in a smaller loss (Van Der Maaten & Hinton, 2008). Consequently, SNE puts more emphasis on the local structure of the data.

In isolation, the cost function is decreasing in q_{ij}. However, the restriction that q_{ij} represents probabilities ($\sum_i \sum_j q_{ij} = 1$) introduces the asymmetry in the cost function. When a small p_{ij} is modeled by a large q_{ij}, some of the density is wasted which means that the other p_{ij} values must be modeled with lower values, introducing a cost. When this happens, the 'wasted' density can be shared among all other values of p_{ij}. Since the log of the ratio between p_{ij} and q_{ij} is weighted by p_{ij}, the cost in this situation is decreased in its importance. When the opposite happens, the cost will be higher due to the increased importance induced by the weight.

This can be illustrated in a simplified case by considering the following ternary plots, which show all possible combinations of probabilities with the restriction that they sum up to one. The simplification stems from the fact that a probability of only three discrete values is considered. The Kullback-Leibler loss function is illustrated by the colour and contour lines of the plot. Each plot corresponds to a unique combination of three values for p_{ij}, whereas the plot itself shows the Kullback-Leibler value for each combination of q_{ij}s. Note that the restriction $p_1 > p_2 = p_3$ is made such that the plots can be compared. Without this restriction, the plot is free to be rotated, without changing the fundamentals. When comparing the two plots, two points (indicated by blue and red dots) are of special interest. On the left graph, the red plot denotes a combination of q_{ij} values that model the highest p_{ij} value much too low. The corresponding value for the Kullback-Leibler function is 0.1838. On the right plot the the red dot denotes a combination of q_{ij} values that model

the highest p_{ij} value much too high. The corresponding value for the Kullback-Leibler function is 0.1920, signifying the asymmetry as it is higher than the previous value.

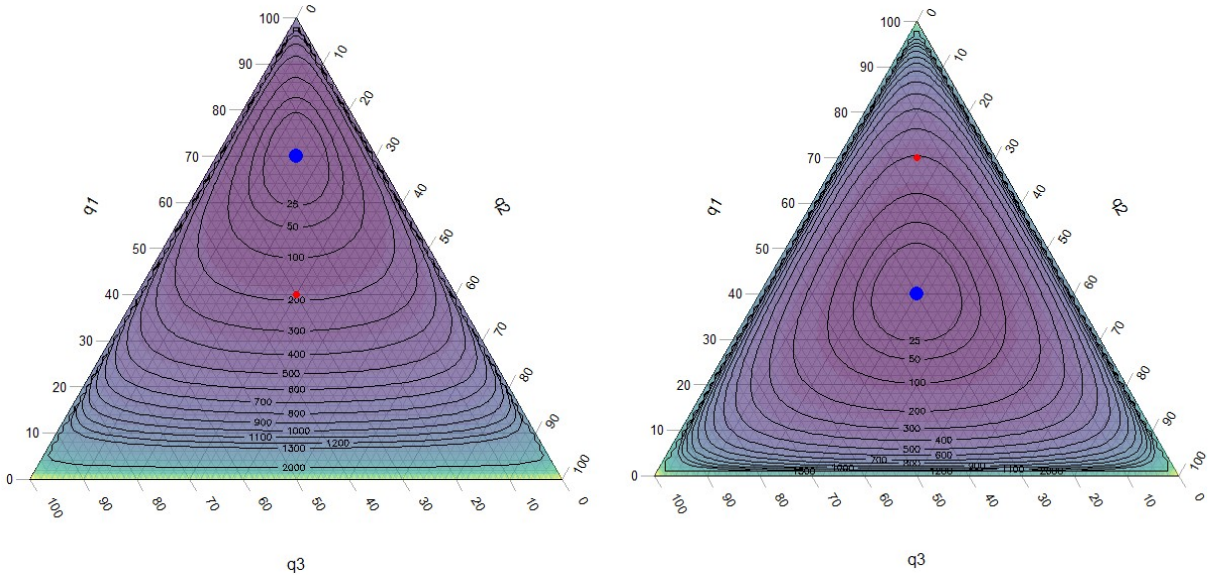

Figure 1: Contour plot of Kullback-Leibler divergence in ternary coordinate system. Left: $p_1 = 0.7, p_2 = 0.15, p_3 = 0.15$. Right: $p_1 = 0.4, p_2 = 0.3, p_3 = 0.3$. The blue dots represent the values of q_i that correspond to the true values (p_i). The red dots model the values of p_i incorrectly (Left: p_1 is modeled too low. Right: p_1 is modeled too large.)

The specific algorithm for finding the points $\boldsymbol{y}_i$ is provided in the methodology section of this paper.

2.1.2 t-Distributed Stochastic Neighbour Embedding

Even though SNE provided somewhat better results than previous methods with respect to the visualization of high dimensional data, Van Der Maaten and Hinton (2008) identified two points of improvement. Firstly, the optimization over the Kullback-Leibler divergences is time consuming and difficult. Secondly, the "crowding problem is introduced as another problem using SNE.

Firstly they aim to simplify the optimization of the method by introducing the concept of symmetric SNE. Instead of optimizing the sum of Kullback-Leibler divergences between all conditional distributions, the idea is implemented to mimimize a single Kullback-Leibler divergence between two joint probability distributions P and Q:

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{4}$$

In this symmetric SNE, the joint probabilities in the low dimensional space (q_{ij}) are given by:

$$q_{ij} = \frac{\exp(-\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)}{\sum_{k \neq l} \exp(-\|\boldsymbol{y}_k - \boldsymbol{y}_l\|^2)}. \tag{5}$$

Instead of defining p_{ij} in a similar fashion, the high dimensional similarities are constructed as a symmetrized version of the conditional probabilities in equation (1):

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}. \tag{6}$$

The main reason for defining p_{ij} in this fashion is explained by Van Der Maaten and Hinton (2008) due to the fact that outliers would have extremely low values of p_{ij} if they would be defined similarly to equation (5). The effect of this would be that the loss associated with this observation would be negligible, regardless of its position in the low dimensional map.

Next to this symmetric version of SNE, Van Der Maaten and Hinton (2008) discuss and improve upon the weakness of SNE to create a low dimensional clutter of points. In visualizing the high dimensional data, one of the desires is to be able to distinguish different data structures, which is hard if all points tend form a clutter in the middle of the map. The authors refer to this as the "crowding problem" and argue that this is an inherent problem of reducing the dimensionality of a data set. The main element of this phenomenon is that it is impossible to embed all pairwise differences perfectly in a lower dimension (i.e. lower than the intrinsic dimensionality of the data). The consequence of this is that there is insufficient space in the lower dimension to model moderate distances, which results in them being modeled too far apart. As we have seen in previous paragraphs, the purpose of SNE is to match the similarities (i.e. p_{ij} and q_{ij}) in the high- and low dimensional space as good as possible. If the distances between moderately distant data points in the high dimension are modelled too far apart in the low dimension, these points will be pulled towards each other. As this happens between many points, the system is unable to let the points in the map drift apart.

To solve this cluttering of points in the center of the low dimensional map, Van Der Maaten and Hinton (2008) consider a different distribution than the previously proposed Gaussian in the low dimensional space: a Student t-distribution. The well known characteristic of its fat tails (relative to a Gaussian) raises a natural way to allow the larger pairwise distances for moderately distant data points. This is illustrated in a simple case in Figure 2, where the shaded areas represent a symmetric interval around the mean of 50%. The figure shows that the same probability can be attained with a greater distance from the mean under the Student-t distribution. This solves the

crowding problem as now the moderate distances in the high dimensional space can be reliably modeled by larger distances in the low dimensional map. When using the Student-t distribution, the joint probabilities in the low dimensional space then become

$$q_{ij} = \frac{(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|\boldsymbol{y}_k - \boldsymbol{y}_l\|^2)^{-1}}. \tag{7}$$

Next to the circumvention of the crowding problem, the authors argue that the introduction of the Student-t distribution increases the efficiency of the method somewhat, as the values for q_{ij} can be computed faster than for regular SNE. In conclusion, t-SNE aims to match the similarities between points in the high dimensional space, given by p_{ij} (Equation 6) and low dimensional space, given by q_{ij} (Equation 5), by minimizing the Kullback-Leibler divergence (Equation 4) over $\boldsymbol{y}_i$.

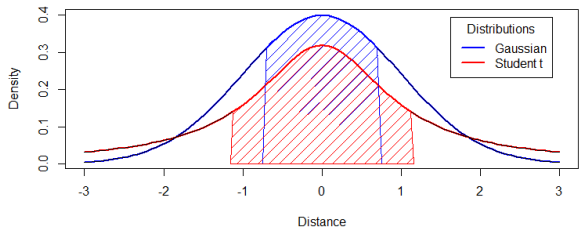

Figure 2: Comparison Gaussian and Student-t Distributions. Shaded areas cover equivalent probabilities

2.2 Multidimensional Scaling

Even though Sammon mapping can be considered a special case of non-linear multidimensional scaling (MDS), the comparison between t-SNE and MDS in Van Der Maaten and Hinton (2008) is limited. Moreover, later literature mainly deals with the application or improvement of t-SNE, making a broader comparison of the method with non-linear MDS underdeveloped. As such, this subsection will concisely introduce MDS and compare it to t-SNE. In Section 4, experiments will be described to compare the two highly popular methods for visualizing high dimensional data.

Multidimensional scaling is based on the premise that a similarity or dissimilarity score can be determined between all pairs of points in a data set. Using these scores, MDS aims to find a map of points in a low dimension, where the pairwise distance between points represents the original dissimilarity between the high dimensional data points as well as possible (Cox & Cox, 2008). This paper considers the non-metric subclass of MDS methods that are based on the formulation of stress functions. Generally, these are based on the minimization of the squared difference between the original dissimilarity and distance of the low dimensional representation of the original data.

The method can be applied with a wide variety of distance measures (for an overview, see Cox and Cox (2008)), but as the final aim of this paper is to compare MDS to t-SNE, this paper considers Euclidean distances. As the method optimizes a potentially non-linear stress function, the optimization is not trivial.

Under the name of least-squares MDS, Groenen and Van De Velden (2016) review a variety of stress based MDS approaches and their optimization and implementation using the **SMACOF** package in R. The focus is on the raw Stress function as described by De Leeuw and Heiser (1980):

$$\sigma^2_{\text{raw}}(\mathbf{X}, \delta_{ij}) = \sum_{i<j} w_{ij}(\delta_{ij} - d_{ij}(\mathbf{X}))^2, \tag{8}$$

where δ_{ij} is the measured dissimilarity between points i and j, $d_{ij}(\mathbf{X})$ is the distance in low dimensional space between points i and j and w_{ij} is a weight determining the importance of a certain pair of points in the determination of the stress function. The main innovation of this particular formulation of the stress function is the introduction of the weights (w_{ij}). As Groenen and Van De Velden (2016) describe, these weights greatly improve the applicability of MDS and facilitate a wider variety of problems that can be tackled with MDS. This increased versatility of MDS implementations is not considered by Van Der Maaten and Hinton (2008), who only compare t-SNE to relatively simple variant of MDS.

The introduction of these weights, however, increase the difficulty of finding the low dimensional embedding that minimizes the stress function. In finding the set of points in the low dimension that matches the similarities in the high dimension as close as possible, this stress function is minimized using the SMACOF algorithm. This technique was developed by De Leeuw and Heiser (1980) and uses the technique of majorization to find the optimal solution to the objective function.

One of the critiques expressed by Van Der Maaten and Hinton (2008) about MDS is its poor performance to focus on the local structure of the data. This argument is substantiated by the comparison of t-SNE and Sammon mapping (Sammon, 1969), which can be considered a variant of this MDS formulation with $w_{ij} = \delta_{ij}^{-1}$. We have seen, now, that MDS can be defined in a broader way than that. When the MDS method is formulated as is done in this paper, these weights can be used to discover and explore to what extent t-SNE might be able to capture the local structure of the data in a superior fashion.

As Groenen and Van De Velden (2016) discuss, the focus on retaining the local or global structure in this model formulation can be altered by taking advantage of the phenomenon of power weights. Using power weights, the weigths w_{ij} are formulated as a power of the dissimilarity measures δ_{ij}:

$w_{ij} = \delta_{ij}^q$. As the value of q varies, the emphasis on local or global structure is emphasized. Values below zero put more emphasis on the small dissimilarities (local structure) and values above zero emphasize large dissimilarities (global structure). This observation can be used in the comparison to t-SNE and a suitable value for q should be determined.

In addition to power weights, a nearest neighbour weighting function can be employed to enforce the focus on the local structure of the data. This method takes a pre-specified number of neighbours and assigns a value $w_{ij} = 1$ if point j is one of the nearest neighbours of point i. The advantage of this method is that the MDS algorithm focuses on retaining the relationship between the nearest neighbours of all points.

Lastly, an exponential kernel type function can be used to differentiate the importance of the local and global structure in the data. Similarly to power weights, points that are further away are given less importance in their contribution to the stress function. In general, this method defines the weights as follows: $w_{ij} = e^{-\delta_{ij}^2}$. The main difference with power weights is the shape of the weights as a function of the distance (See Appendix A). This weighting works best when the distances are not too far from zero and hence, it is often combined with standardization of the data.

The weights that are described above all focus on increasing the importance of the small dissimilarities. We have seen, however, that even though t-SNE focuses on retaining small distances, it also aims to let clusters of locally similar points drift apart. This 'outside' force has been researched in the context of MDS as well. Chen and Buja (2009) extend the stress based MDS definition by altering some of the dissimilarities and weights. In essence, they propose an extension of the above-mentioned k-nearest neighbour approach. Their idea is to introduce a penalty for large dissimilarities, essentially making them larger. To compensate for the larger dissimilarity and to prevent them from dominating the stress function, these penalties are accompanies by a reduction in the weight of these points. The implementation is introduced as follows: for each point i, a local neighbourhood ($\mathcal{N}_i$) is constructed with a pre specified number of neighbours. For the points that are not in this neighbourhood, δ_{ij} is changed to a fixed large value, D_∞, and the corresponding w_{ij} is set very low. For the points that are in the local neighbourhood, nothing is changed, and their weight is set equal to one. As such, the stress formulation in Equation 8 is changed to:

$$\sigma^2_{\text{raw}}(\mathbf{X}, \delta_{ij}) = \sum_{(i,j)\in\mathcal{N}} w_{ij}(\delta_{ij} - d_{ij}(\mathbf{X}))^2 + \sum_{(i,j)\notin\mathcal{N}} w_{ij}(D_\infty - d_{ij}(\mathbf{X}))^2, \tag{9}$$

where D_∞ is set very high and w_{ij} is set equal to one for the nearest neighbours and set in such a way that it off-sets the penalty in the stress function (D_∞) for the points that are not among the

nearest neighbours. In that way, $\mathbf{X}$ is chosen in such a way that points i and j are far away from each other, as the contribution of the distance between the pair of points to the cost function is minimal when $d_{ij}(\mathbf{X}) = D_\infty$. In this way, Chen and Buja (2009) try to induce a force that pushes points away from each other when they are not similar. In general, they find that this Local MDS approach is better able to retain the local structure of the data than other MDS implementations. In this paper, the values for w_{ij} and D_∞ are constructed from a penalty coefficient c: $w_{ij} = c$ if point j is not in the local neighbourhood of point i and D_∞ is differentiated for each pair of points as $D_{ij} = \delta_{ij}/c$. When c is chosen low (such as 0.01 or 0.001), this creates the desired outside force between points that are not in each other's neighbourhood. This deviates slightly from the original approach that is taken by Chen and Buja (2009), who work out Equation 9 and combine the w_{ij} and D_∞ parameters by a single parameter t. Doing this, however, changes the structure of the stress function, making it unsuitable for the **SMACOF** package to find the solution. The implementation with the penalty parameter c, however, captures the same idea and is implemented in other literature as well (Groenen & Van De Velden, 2016).

We have seen that both t-SNE and MDS aim to preserve the similarity between pairs of data points in the high dimensional space, by mapping them in a lower dimension. The methods, however, take different approaches. Van Der Maaten and Hinton (2008) argue that MDS is not capable of retaining the local structure well, because because extremely small distances in the high dimensional space excessively contribute to the cost function, compared to the 'regularly' small distances. As the local structure of the data consists of more than only the extremely small distances, MDS, is not well able to capture this. This shortcoming, however, is not entirely generalizable to all MDS models based on stress equation (8), when other weighting functions are considered. This paper will investigate whether this finding is still valid when different weight functions are considered as their effect on the degree of locality in MDS is different.

3 Algorithms

We now turn to the mechanics of both methods and consider how the optimal solution can be found for both t-SNE, using the algorithm that Van Der Maaten and Hinton (2008) introduced, and MDS based on the SMACOF algorithm (Groenen & Van De Velden, 2016; De Leeuw & Mair, 2009).

3.1 Implementing t-SNE

To find the set of points $\boldsymbol{y}_i$ that represent the high dimensional data points $\boldsymbol{x}_i$ as well as possible, t-SNE optimizes the Kullback-Leibler divergence that is given in Equation 4 over all datapoints $\boldsymbol{y}_i$ (the points in the low dimensional map). A gradient descent approach is used for this purpose. Before we consider the mechanics of this method, the method to determine the σ_i parameter in Equation 6 is described. This parameter is a measure of the variance of the Gaussian that is centered around a point $\boldsymbol{x}_i$ in the high dimension space. Hinton and Roweis (2003) argue in their initial introduction of SNE that using a fixed variance for all points is unlikely to be optimal due to varying density of points in the high dimensional space. To account for this, a binary search is conducted to find the value of σ_i that results in a pre-specified perplexity.

$$PP(P_i) = 2^{H(P_i)}, \tag{10}$$

where $H(P_i)$ is defined as the Shannon entropy: $H(P_i) = -\sum_j p_{ij} \log_2 p_{ij}$. Intuitively, the authors argue that the perplexity can be seen as the effective number of neighbours of a point $\boldsymbol{x}_i$, employed in k nearest neighbours clustering methods.

After the definition of the final parameter σ_i, the Kullback-Leibler divergence can be optimized over the points $\boldsymbol{y_i}$ in the low dimension using a gradient descent. Van Der Maaten and Hinton (2008) show that the gradient of Equation 4 has a surprisingly simple form:

$$\frac{\partial C}{\partial \boldsymbol{y}_i} = 4 \sum_j (p_{ij} - q_{ij})(\boldsymbol{y}_i - \boldsymbol{y}_j)(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1}, \tag{11}$$

The gradient descent is initialized by taking a random sample of a small variance Gaussian, centered around the origin. As the optimization is not a convex problem, a momentum term is added in the update of the gradient to avoid the algorithm to get stuck in poor local minima. The update formula is then defined as follows:

$$\mathbf{Y}^{(t)} = \mathbf{Y}^{(t-1)} + \eta \frac{\partial C}{\partial \mathbf{Y}} + \alpha(t)(\mathbf{Y}^{(t-1)} - \mathbf{Y}^{(t-2)}), \tag{12}$$

where $\mathbf{Y}^{(t)}$ is the matrix containing solutions $\boldsymbol{y}_i$ at iteration t of the algorithm, η is the learning rate, and $\alpha(t)$ denotes the momentum at iteration t. The learning rate is added to speed up the process of the optimization. In accordance with Jacobs (1988), an adaptive learning rate is implemented which increases the importance of the direction of the gradient that are stable. As a final trick to find better visualizations, t-SNE is implemented with an 'early exaggeration'. When the number of iterations is still low in the initial stage of the optimization, the values of p_{ij} are multiplied by

a constant (usually 4). Due to this exaggeration of the p_{ij} values, the algorithm is forced to model relatively large corresponding q_{ij} values, which generally results in tight and widely separated clusters in the low dimensional map. Algorithm 1 provides the pseudo-code as it is described in Van Der Maaten and Hinton (2008).

Algorithm 1: Simplified description t-SNE (Van Der Maaten & Hinton, 2008)

Result: Low dimensional embedding of high dimensional data $\mathbf{Y}^{(T)}$

Input: high dimensional data set $\mathbf{X}^{(0)}$; perplexity; optimization parameters: number iterations T, learning rate η , momentum $\alpha(t)$, early exaggeration parameter;

begin

normalize or standardize data and perform initial PCA compute $p_{j|i}$ with fixed perplexity (using binary search and Equation 1);

compute $p_{ij} = \frac{p_{j_i} + p_{i|j}}{2n}$;

initialize low dimensional embedding: $\mathbf{Y}^{(0)} \sim \mathcal{N}(0, 10^{-4}\mathbf{I}_n)$;

for $t = 1 \rightarrow T$ **do**

compute q_{ij} (Equation 5);

compute gradient: $\frac{\partial C}{\partial \mathbf{Y}^{(t-1)}}$ (Equation 11);

update solution: $\mathbf{Y}^{(t)} = \mathbf{Y}^{(t-1)} + \eta \frac{\partial C}{\partial \mathbf{Y}^{(t-1)}} + \alpha(t)(\mathbf{Y}^{(t-1)} - \mathbf{Y}^{(t-2)})$

end

end

3.2 Implementing MDS

The MDS approach that is taken in this paper is based on the optimization of the stress function defined in Equation 8. There are multiple ways to minimize the stress function, which is not trivial to optimize (Groenen & Van De Velden, 2016). This paper considers the highly effective SMACOF algorithm with implementation in R, as introduced by De Leeuw and Mair (2009). This method is based on an optimization technique named majorization, which results in an algorithm that guarantees the descent of the objective function. More information on the technical details can be found in De Leeuw and Mair (2009) and Groenen and Van De Velden (2016).

4 Experiments and Results

This section will present a variety of implementations of the t-SNE and MDS methods that were discussed throughout this paper. Firstly, the replication of the t-SNE algorithm in R will be

discussed and verified. Then, some experiments, based on three different data sets will be performed and analysed. Firstly a comparison between the MDS implementations and t-SNE will be made using the MNIST data set. This data set is analyzed first, since Van Der Maaten and Hinton (2008) describe t-SNE's good performance especially in the context of these data. However, such a real world data sets lacks a certain ground truth, making the comparison limited in its power. Consequently, simulated data is analyzed using both implementations to gain more insight in the comparison of the method in an controlled environment. Lastly the performance of the methods is introduced in the field of visualizing the similarities between countries around the world.

4.1 Replication t-SNE

Since the original software for the t-SNE was developed for MATLAB implementations [1], we first focus on testing this paper's implementation of t-SNE in R. For this purpose, Fisher's well-known iris data set (with standard implementation in MATLAB and R) will be considered. The data set consists of 150 entries, corresponding to a particular iris flower and member of one of three categories: Setosa, Virginica or Versicolor. Each flower has four corresponding variables that describe the sepal width, sepal length, petal length and petal width. To verify the R implementation of the t-SNE algorithm, this four dimensional data set is reduced to two dimensions and visualized accordingly.

To verify the correctness of the R implementation and maximize the comparability between the two implementations, we initialize both implementations with the same low dimensional embedding (see Appendix B). In the standard settings for the MATLAB software, the values for the hyper parameters are defined as: $\eta = 500$, $\alpha(t) = 0.5$ for $t < 250$ and $\alpha(t) = 0.8$ for $t \geq 250$, perplexity is set equal to 30, p_{ij} is blown up by a factor 4 in the first 100 iterations, 1000 iterations are done. These values are provided for the application of the method on the MNIST dataset. Performing the algorithm on the iris data set with these parameters does not give optimal results. Especially the learning rate is too high and harms the convergence of the algorithm. This can be seen in Appendix B, where two graphs show a cost function that shows undesirable jumps, making the algorithm unable to converge. This also emphasises that t-SNE is an algorithm for which the optimization is far from trivial. Accordingly, for the iris data set, the learning rate is changed to 10 based on the analysis of the cost function, as this value yields the lowest Kullback-Leibler divergence for these data.

Figure 3 shows the output of the two implementations (left: R, right: MATLAB). On a first

[1] https://lvdmaaten.github.io/tsne/

glance the two figures look nearly identical, indicating the correctness of the R code. Extending the analysis and considering the left plot in Figure 4, the two methods indeed produce the same outcome, which can be seen by distances between all points lying on the 45° line for both implementations. Finally, we investigate the value of the Kullback-Leibler divergence over the iterations of the algorithm. In Figure 4, a clear decreasing function can be seen that converges to a minimum value. The jump around iteration 100 corresponds to the values for p_{ij} that are changed to their true value again (i.e. the end of the early exaggeration stage). Thus, we can conclude that the R implementation works correctly and can be used for further analysis of the data sets under scrutiny in this paper.

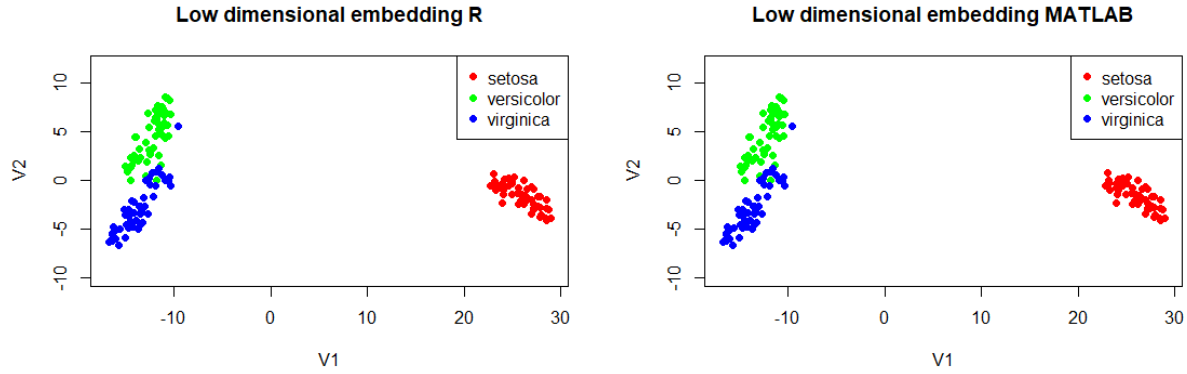

Figure 3: Low dimensional embeddings of Iris data set (learning rate = 10)

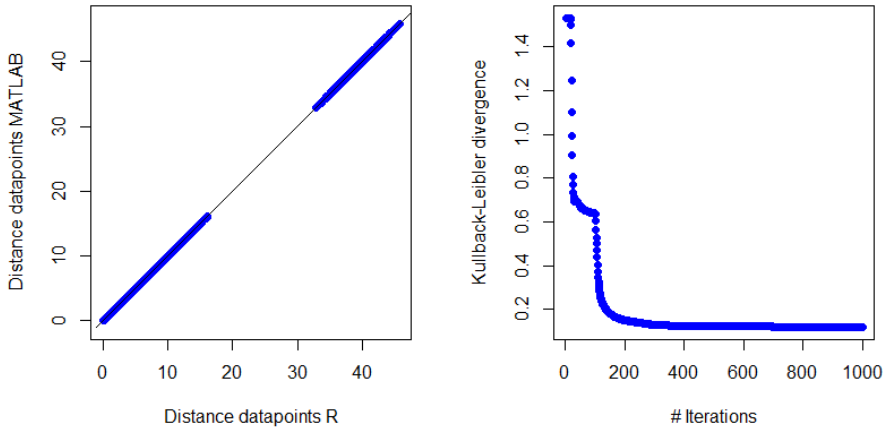

Figure 4: Results comparison R and MATLAB. Left: comparison distances MATLAB and R. Right: Kullback-Leibler divergence for R implementation

4.2 Comparison t-SNE and MDS

One of the main aims of this paper is to compare the performance to visualize a high dimensional data set using t-SNE and MDS. Two approaches are taken to investigate this. Firstly both MDS and t-SNE are applied to a data set that has a particularly good performance for the t-SNE method (Van Der Maaten & Hinton, 2008): the MNIST data set. Afterwards both methods are applied to a simulated high dimensional data set for which the low dimensional structure is known.

Before the specific experiments are described and their corresponding results discussed, we introduce a method to compare the visualizations that result from t-SNE and MDS. As t-SNE and MDS both rely on different cost functions and algorithm design, a new evaluation metric is introduced, based on the work of Chen and Buja (2009). The metric is based on the idea of neighbourhood preservation by the methods. As the aim of t-SNE is to retain the local structure of the data, and we also try to accomplish this with the MDS implementations, this is an interesting characteristic to assess. A high degree of overlap between the local neighbourhoods of a data point in the high dimension and low dimension indicates that the embedding is a good representation of the high dimensional data structure. Therefore, the metric for the pointwise overlap is defined as:

$$N_K(i) = |N_K^Y(i) \cap N_K^X(i)|, \tag{13}$$

where $N_K^Y(i)$ is defined as the set of K-nearest neighbours of point i in the low dimensional embedding, and $N_K^X(i)$ is the high dimensional equivalent. A global measure for the overlap is simply obtained by averaging over all individual overlap coefficients. To normalize this global overlap coefficient, we divide by K:

$$M_K = \frac{1}{K} \sum_{i=1}^{n} N_K(i). \tag{14}$$

In this paper a value of $K = 10$ will be used to construct the nearest neighbourhoods. In the rest of this paper, the embeddings that are visualized in the main text have the characteristic that the specific parameter values of the method yield the highest average overlap coefficient, unless stated otherwise.

4.2.1 MNIST Data Set

The MNIST data set is constructed from 60,000 images of handwritten digits (0 to 9). These images each consist of 784 (=28x28) pixels, corresponding to the dimensionality of the data. For each pixel, a gray scale value ranging from 0 to 1 is provided. For the comparison a subset of 6000

data points of this data set is used that was collected by Van Der Maaten and Hinton (2008) [2], of which a random sample of 1000 digits is taken. The t-SNE algorithm will be implemented in close correspondance to the work of Van Der Maaten and Hinton (2008) and equivalent parameters will be used the implementation.

As has been described before, this paper attempts to shed more light on the relationship between t-SNE and MDS. To focus on the local structure of the data (as is the goal of t-SNE), MDS will be implemented with the following weights as introduced in Section 2.2: 1) power, 2) k-nearest neighbour and 3) kernel weights. In addition, Local MDS will be implemented to investigate whether the introduction of an outside force can lead to similar visualization as t-SNE. These implementations will initially be performed in the traditional way, where the weights are based on the Euclidean distances. Next to this, the $\mathbf{P}$ matrix that is constructed in t-SNE will be used to create the weights and as input to create a dissimilarity matrix as alternative for the Euclidean distance. This idea comes from the Bibliometry literature. In that field, a MDS-like method was introduced (VOS Viewer) that uses similarity scores as input for the weights (Van Eck, Waltman, Dekker, & van den Berg, 2010). This paper uses that idea and creates a simple dissimilarity score, that is derived from the $\mathbf{P}$ matrix: $1 - p_{ij}$.

In summary, MDS will be applied in three different fashions: 1) Using $\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and w_{ij} constructed from δ_{ij} 2) Using $\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and w_{ij} constructed from $1 - p_{ij}$ and 3) Using $\delta_{ij} = 1 - p_{ij}$ and w_{ij} constructed from $1 - p_{ij}$. In Appendix A a table can be found where an overview is provided of which MDS implementations will be done. This structure will also be followed for the other two data sets.

The t-SNE parameter setting to analyze the MNIST data set is constructed in accordance with Van Der Maaten and Hinton (2008). That means that a perplexity of 30 and learning rate of 500 is be used. Fifty random initialization are considered, after which the result with the lowest Kullback Leibler divergence will be analyzed. More details can be found in the supplementary material and code.

The final low dimensional embedding can be found in Figure 5. Clear clusters can be identified from this embedding, although not all clusters are separated from each other. Moreover, there are some points that seem to be represented in the wrong cluster.

[2] https://lvdmaaten.github.io/tsne/

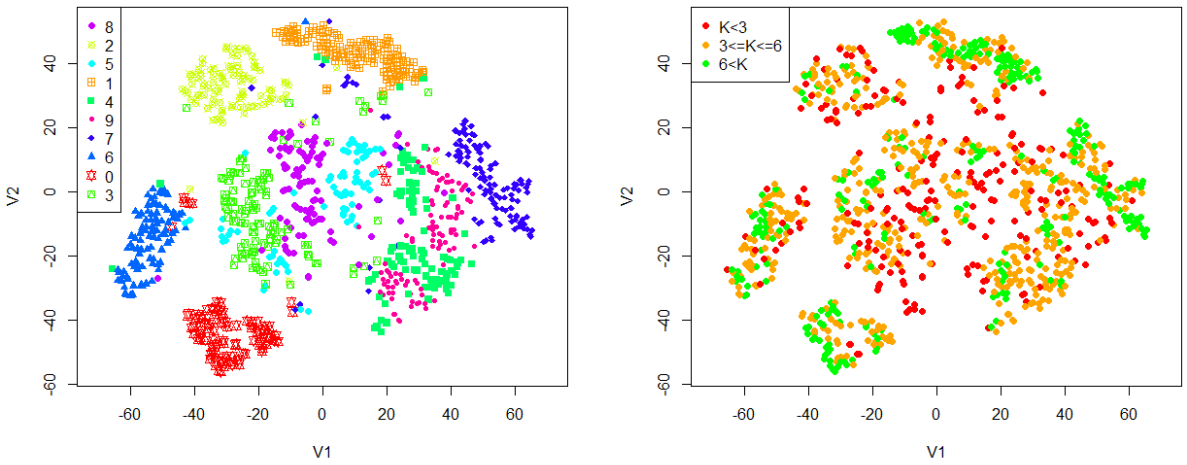

Figure 5: t-SNE: Left: Low dimensional embedding MNIST data set. Right: Overlap local neighbourhoods of each point in high and low dimension (K: number of points which are in both the high and low dimensional neighbourhoods of point i)

However, when considering the corresponding picture, those can be attributed to anomalies in the way the digits were drawn. Figure 6 shows two of these cases. The left to pictures show the drawing of a 6 and 1, which corresponds to the observation of 6 that is found between the 1s. The right figure denotes a 9 and 4 that are mapped very closely together. From these pictures, we can see that it is not surprising that t-SNE is not well able to separate these values, as they look very similar, even though their label is different.

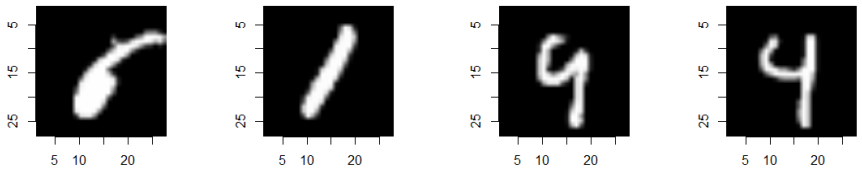

Figure 6: Comparison seemingly wrong embedded points. From left to right: 6, 1, 9, 4

Moreover, we can see that the points corresponding to the digits 3, 5 and 8, and 4 and 9 are not clearly separable as clusters. This can also be explained by the previous argument that digits can be drawn in similar ways and have similar grayscale values for the pixels. For these digits, we can also observe that t-SNE is less able to preserve the local structure of the data in the low dimensional embedding. The right plot in Figure 5 shows for each point the level of overlap between the nearest

neighbourhoods (of size 10) around the point when comparing the high and low dimension. The better separated a cluster is, the higher the overlap seems to be and as such the better the local structure is preserved.

The results for the MDS approach in embedding the MNIST data in two dimensions are highly variable. We start with the implementations that use $\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and use δ_{ij} as input for w_{ij}. In Figure 7 three examples of the δ_{ij}s are given for which the weight is larger than zero. Here we see a desired spread of the values that is needed for MDS. For the Local MDS histogram, there seems to be hardly any variation. However, the very small bar on the left of the histogram is equivalent to the middle histogram corresponding to k-nearest neighbour weights, when one would zoom in.

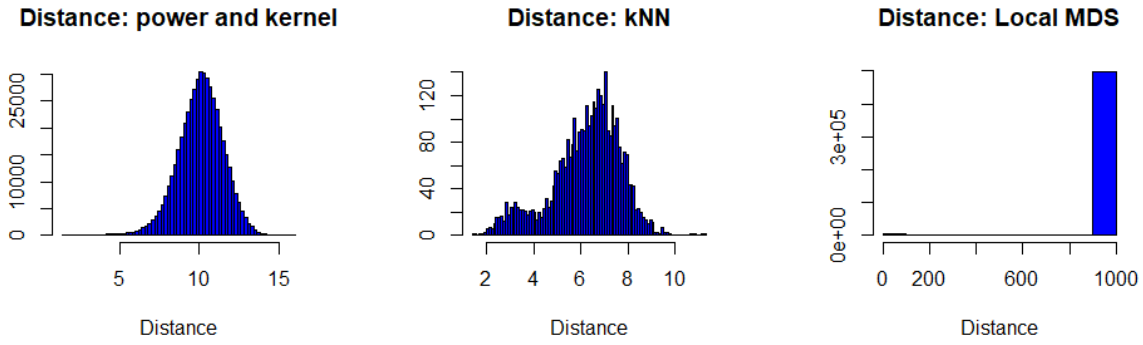

Figure 7: MNIST: Histograms $\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ for which $w_{ij} > 0$. Note: k-nearest neighbour with $k = 5$ and Local MDS with $k = 5$ and $c = 0.001$

In Appendix C, the plots corresponding to the embedding with the highest average overlap coefficient (M_{MDS}) for all types of weights (power, kernel and kNN) can be found. In this section, we present and focus on the best results from the different MDS approaches. In general, we can observe that MDS has more trouble differentiating between the digits than t-SNE. Even though some clusters in the data are apparent (e.g. corresponding to 0s, 1s and 2s), the method is unable to clearly separate these values from the rest of the values for the power, k-nearest neighbour and kernel weighting methods. This can be seen in the left plot in Figure 8, corresponding to the MDS embedding using power weights ($q = 2$), which resulted in the largest overlap coefficient for the power, kernel and k-nearest neighbour ($M_{\text{MDS,power}} = 0.181$). It is somewhat surprising that this overlap coefficient is highest for this particular power, as a positive power suggests the relative high importance of high dissimilarities.

It is not entirely surprising that these weight types are not able to separate the categories from one another, as these weights do not specifically impose a condition that drives dissimilar points

away from each other. When introducing this force with Local MDS, we can see that for certain values for c and k, some of the clusters can be separated from the other digits. The right plot in Figure 8 shows that the groups corresponding to the zero and six digit have drifted away from the other data points. Moreover, there seems to be a clear cluster of 2s and 3s in the top of the plot, which, however, have not drifted away much from the other points. Lastly, the local MDS implementation yields a higher average overlap coefficient than the other three weighting methods: $M_{\text{LMDS}} = 0.291$, with $c = 0.001$ and $k = 5$, making it better at retaining the local structure.

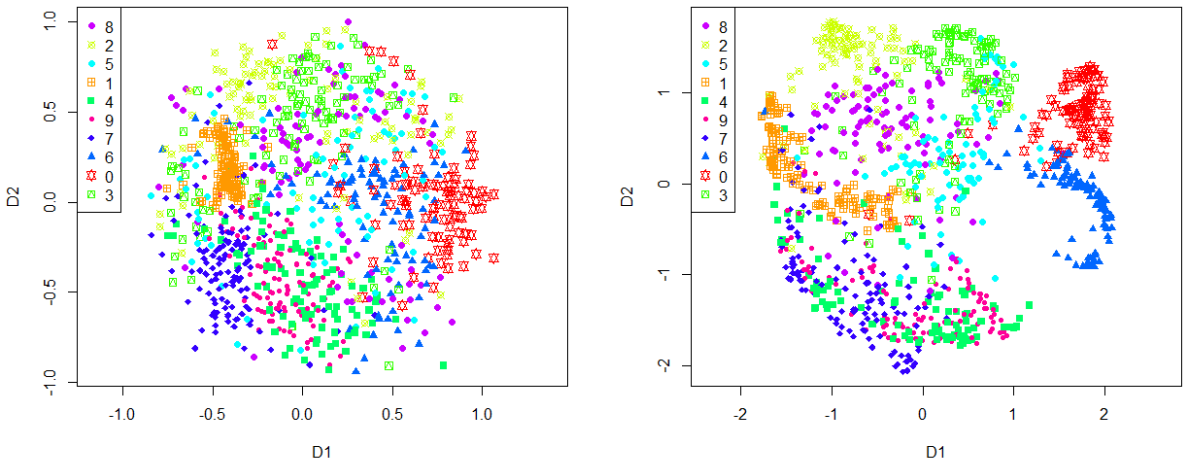

Figure 8: MDS solutions for MNIST data set using $\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and use δ_{ij} as input for w_{ij}. Left: MDS with power weights ($q = 2$). Right: Local MDS ($c = 0.001$, $k = 5$)

Extending the standard MDS implementation with the Euclidean distance as input for the weighting function, we now briefly discuss using the $\mathbf{P}$ matrix from t-SNE as input to create the MDS weights, while still using $\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$. Due to the large number of points and little variation in the p_{ij} values (Figure 10), using power weights does not differ significantly from 'plain' MDS, as the weights are close to the same value for all pairs of points. The kernel based method suffers from the same phenomenon and also produces similar results. Further normalization of these values does not help to solve this. Hence, using the similarity matrix as input for the weights does not improve the embeddings significantly. Furthermore, the k-nearest neighbour weighting function yields similar weights when $\mathbf{P}$ is used as input, since the p_{ij} values are constructed from the Euclidean distances and therefore, the neighbourhoods created from distances and p_{ij} values are highly similar. Their average overlap coefficient is also very similar: $M_{\text{kNN,distance,distance}} = 0.0309$ and $M_{\text{kNN,P,distance}} = 0.051$ ($k = 5$) . In Appendix C, the embeddings for these methods can be

found and their similarity to the distance based weight solutions can be checked. Overall, it can be observed that the MDS approach that uses the $\mathbf{P}$ matrix from t-SNE as input for the weighting functions does not perform much different from the standard implementation.

The Local MDS implementation, however, does give different shaped embeddings when $\mathbf{P}$ is solely used to construct the weights. This can be seen as a consequence of the changes that are made to the weights as a consequence of the penalty term c. Even though the k-nearest neighbours are similar due to the high correlation between the distances and p_{ij} values, the penalty term emphasises and magnifies their difference. Figure 9 shows the plot of this embedding, for which some clusters can be separated from each other. It is especially interesting to see that these clusters are similar to the clusters that t-SNE is able to separate relatively well from the other points. However, when comparing this solution to the t-SNE embedding, it can be seen that the digits that are not well distinguishable from others show less structure. The points corresponding to, for example, fours and nines are plotted more separate in the t-SNE embedding than in the MDS plot. Moreover, comparing the average overlap coefficients for t-SNE and Local MDS suggests that t-SNE is better able to capture the local structure of the data: $M_{\text{t-SNE}} = 0.515$ and $M_{\text{LMDS}} = 0.318$ ($k = 5$, $c = 0.001$). Visually, this can also be observed from the plot on the right (Figures 5 and 9), where the local neighbourhoods are preserved better using t-SNE, considering both the relatively well separated clusters and the clutter of points which have not drifted apart.

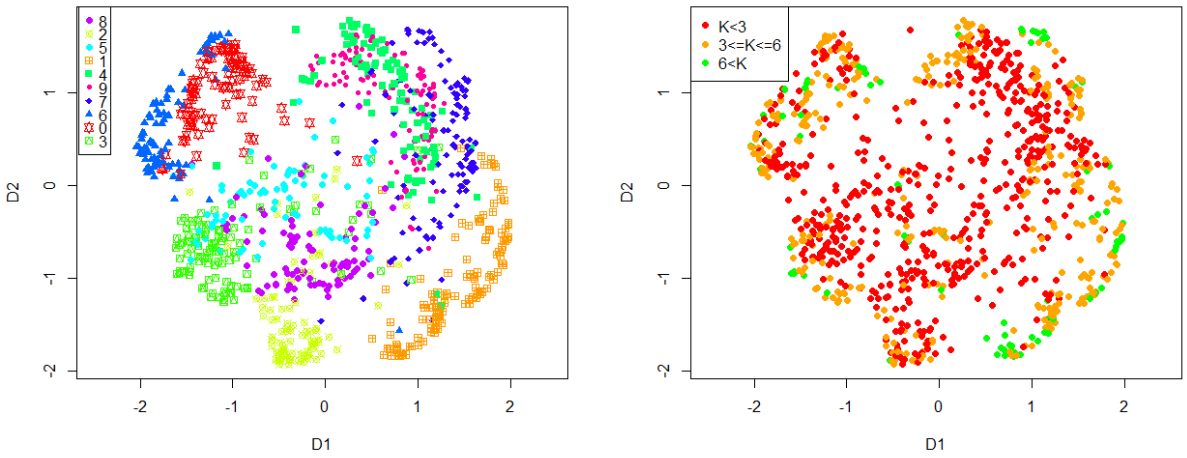

Figure 9: MDS solutions for MNIST data set using $\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and use $1 - p_{ij}$ as input for w_{ij}. Left: Local MDS ($c = 0.001$, $k = 5$). Right: Overlap local neighbourhoods of each point in high and low dimension (K: number of points which are in both the high and low dimensional neighbourhoods of point i)

Next to the previous approach, we can use $\mathbf{P}$ as input for both the weights and to construct a dissimilarity matrix ($\delta_{ij} = 1 - p_{ij}$) that is used for MDS. In general, this does not yield good results for the power and kernel weighting types. Again, since the values in $\mathbf{P}$ are all very close to zero and do not differ highly, the elements of the dissimilarity matrix that is constructed ($\delta_{ij} = 1 - p_{ij}$) are very similar for power and kernel weights (see Figure 10). When this is the case, all points want to be spaced equidistantly, which creates a typical 'ball' of points (see Appendix C). From Figure 10 we can see that the k-nearest neighbour weight do show variation, making them suitable for MDS. This, however, yields a worse embedding than using the Euclidean distance as dissimilarity and $\mathbf{P}$ as input for the weights: $M_{\text{kNN,P,P}} = 0.0336$.

Since similar results for MDS using $\mathbf{P}$ to construct the weights and dissimilarity matrix are also visible for the other data sets, this will not be discussed again later in detail. Hence, we will focus on the Local MDS using $\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and t-SNE embeddings for the other data sets.

The main disadvantage and limitation of the analysis described above is that the underlying low dimensional structure in the data is unknown (if it even exists). That brings us to the next method of comparison, which aims to compare the two visualization techniques for simulated data with a known structure.

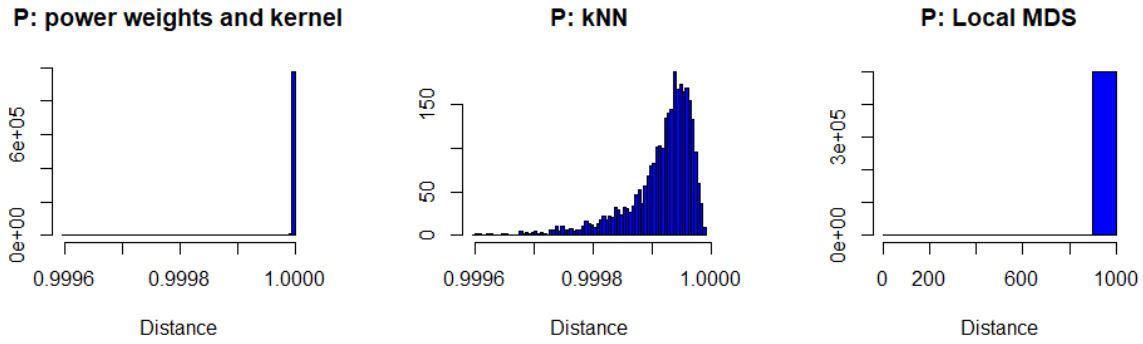

Figure 10: MNIST: Histograms $\delta_{ij} = 1 - p_{ij}$ for which $w_{ij} > 0$. Left: power and kernel weights. Middle: k-nearest neighbour weights with $k = 5$. Right: Local MDS ($k = 5$, $c = 0.001$)

4.2.2 Simulated Data Set

The second comparison will be based on a simulated two dimensional data set containing five of clusters of data, each containing 100 data points. This low dimensional data set will then be blown up to twenty dimensions using a rotation-expansion matrix. Using this approach, the distances and angles between the points in the data will preserved in the high dimension.

To create the data set, the **clusterSim** [3] package in R will be used. This packages uses a separate bivariate Gaussian distribution with a known mean and covariance matrix for each cluster. Two dimensional clusters are then generated by doing a number of random draws from the corresponding Gaussian distribution for each cluster. For this paper, five clusters with means: (0,0), (10,0), (0,10), (10,10) and (5,5) and unit covariance matrix are generated. The increase in dimensionality of the data is achieved by using the characteristics of square rotation(-expansion) matrices:

$$\mathbf{R}^T = \mathbf{R}^{-1} \iff \mathbf{R}^T\mathbf{R} = \mathbf{I} \iff \mathbf{R}\mathbf{R}^T = \mathbf{I}. \tag{15}$$

It is easy to show that using a rotation matrix to make transformation of the data preserves the distance between the data points:

$$\|\mathbf{R}\mathbf{x}\|^2 = (\mathbf{R}\mathbf{x})^T(\mathbf{R}\mathbf{x}) = \mathbf{x}^T\mathbf{R}^T\mathbf{R}\mathbf{x} = \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|^2,$$

where R denotes a rotation matrix characterized by Equation 15 and $\mathbf{x}$ is a vector.

Using these properties of rotation matrices, we can blow up the dimensionality of the artificial low dimensional data set. When we post-multiply our two dimensional simulated data set with the transpose of the first two columns of an arbitrary rotation matrix, we obtain a new data set with dimensionality equal to the dimension of the rotation matrix:

$$\mathbf{X}_{\text{high}} = \mathbf{X}_{\text{low}}{\mathbf{R}_{1,2}}^T,$$

where $\mathbf{X}_{\text{high}}$ denotes the high dimensional data set, $\mathbf{X}_{\text{low}}$ is the original two dimensional data and $\mathbf{R}_{1,2}$ denotes the first two columns of a rotation matrix with size $[N \times N]$, where N is the number of rows of $\mathbf{X}_{\text{low}}$. Similarly to the proof provided above, it can be shown that the distance and angles between the data points will be preserved in this higher dimension. The rotation matrix to blow up the dimensionality of the simulated data set will be obtained from the Singular Value Decomposition of a randomly generated square matrix of dimensionality [20 x 20]. This method decomposes a matrix in the following way:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

where $\mathbf{U}$ and $\mathbf{V}^T$ are orthogonal matrices, satisfying the characteristics of a rotation matrix, and Σ is a diagonal matrix containing the singular values of matrix $\mathbf{A}$.

One disadvantage of using this rotation matrix to blow up the dimensionality, is that if the distances do not change, MDS will be able to exactly recover the low dimensional structure. For that

[3] https://cran.r-project.org/web/packages/clusterSim/index.html

reason, Gaussian noise will be added to move the data way from the low dimensional manifold that is embedded in the high dimension. Two data sets will be created: 1) the high dimensional coordinates are contaminated with $\epsilon_i \sim N(0,1)$ 2) the high dimensional coordinates are contaminated with $\epsilon_i \sim N(0,2)$. As the noise is increased, both methods should have more trouble recreating the clusters in the low dimensional embeddings.

The comparison between t-SNE and MDS will be performed similarly to the analysis on the MNIST data set. As some of the results with respect to the MDS implementations are very similar, we will focus our attention on the Local MDS implementation in this case.

Figure 11 gives a visual representation of the low dimensional data, which results from the simulation.

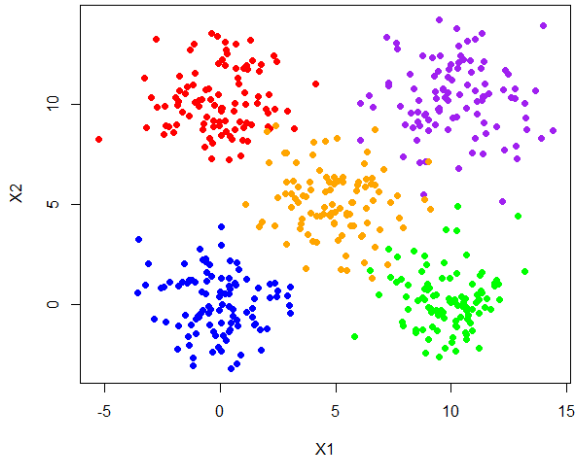

Figure 11: Low dimensional simulated data set with five clusters

The t-SNE implementation is performed with a perplexity of 50 and a learning rate of 50. This configuration of learning rate and perplexity was found by first assessing the effects of changing the learning rate while keeping the perplexity fixed. The optimal learning rate for this perplexity was chosen and kept fixed while changing the perplexity. This approach was favoured over a full grid search due to computational limitations. When t-SNE is applied to the simulated data, we can observe that the method is able to distinguish between the clusters is a good way for the data with low noise (Figure 12). When the higher variance contamination is considered, t-SNE is still able to give a relatively good overview of the structure of the data, but the method cannot clearly separate the clusters anymore, which is normally one of the main strenghts of t-SNE. Furthermore, we can see that for the low noise data, t-SNE does not always preserve the orientation among the clusters.

In Appendix D, a plot is shown for which the purple and blue clusters are shown to be adjacent, while in the true data representation they should be on opposite sides.

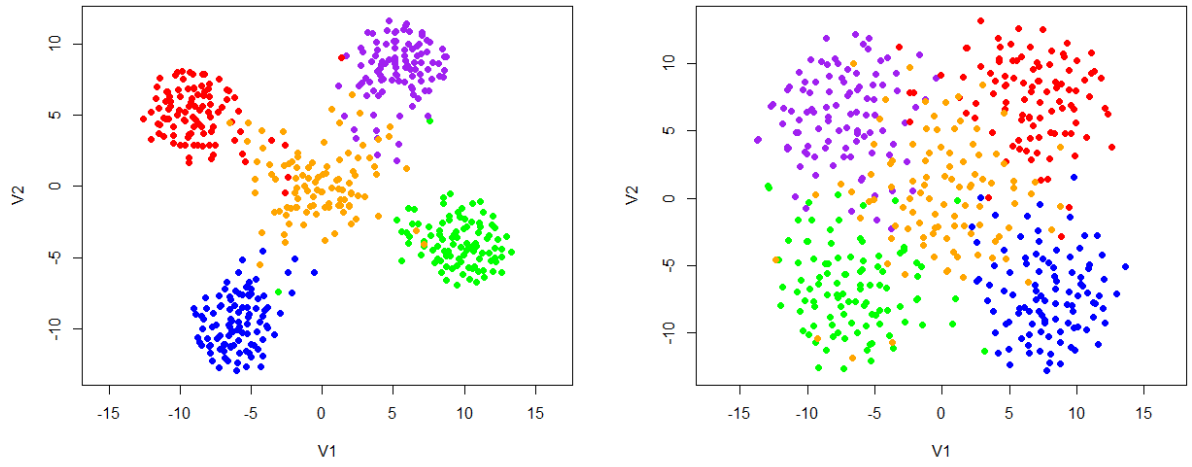

Figure 12: t-SNE Embeddings for simulated data. Left: low noise ($\sigma = 1$). Right: high noise ($\sigma = 2$)

The general results for the MDS implementations with power, kernel and k-nearest neighbour weights are very similar to the results for the MNIST data set. Overall, these implementations reconstruct the structure relatively well, especially for the little contamination data, without being able to let some clusters of similar points drift apart. Due to these similar observations as before, we do not present the embeddings in the main text.

The Local MDS, however, changes the weights and dissimilarity matrix and is, unlike the other MDS implementations, theoretically not able to perfectly reconstruct the low dimensional structure. Figure 13 shows the embeddings for Local MDS for the light contamination (left) and stronger noise data (right), corresponding to the configurations that attain the highest average overlap coefficient.

Just like for the MNIST data set, constructing the weights from $\mathbf{P}$ (with $\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$) creates embeddings that preserve the local neighbourhood in a better way, although the difference is minor ($M_{\text{LMDS, dist, } \sigma=1} = 0.303$ and $M_{\text{LMDS, P, } \sigma=1} = 0.311$). We can observe that the Local MDS approach seems to reconstruct tighter clusters of the same points than t-SNE. On the other hand, when considering the overlap coefficients, it becomes clear that t-SNE again performs better in the preservation of the local structure of the data ($M_{\text{t-SNE, } \sigma=1} = 0.4150$ and $M_{\text{LMDS, P, } \sigma=1} = 0.3108$). Especially for the higher noise case, the difference is apparent ($M_{\text{t-SNE, } \sigma=2} = 0.3452$ and $M_{\text{LMDS, P, } \sigma=2} = 0.1818$). Lastly, it is interesting to observe the behaviour of the Local MDS embeddings as the penalty parameter c is changed. For the simulated data set, we can clearly see

that when c is made smaller (and hence the penalty higher), the clusters move further away. When we interpret $\frac{1}{c}$ as the 'outside force', this seems logical. The retainment of the local structure, however, goes down as the clusters move further away from each other. This is visualized for the low contamination case in Appendix E.

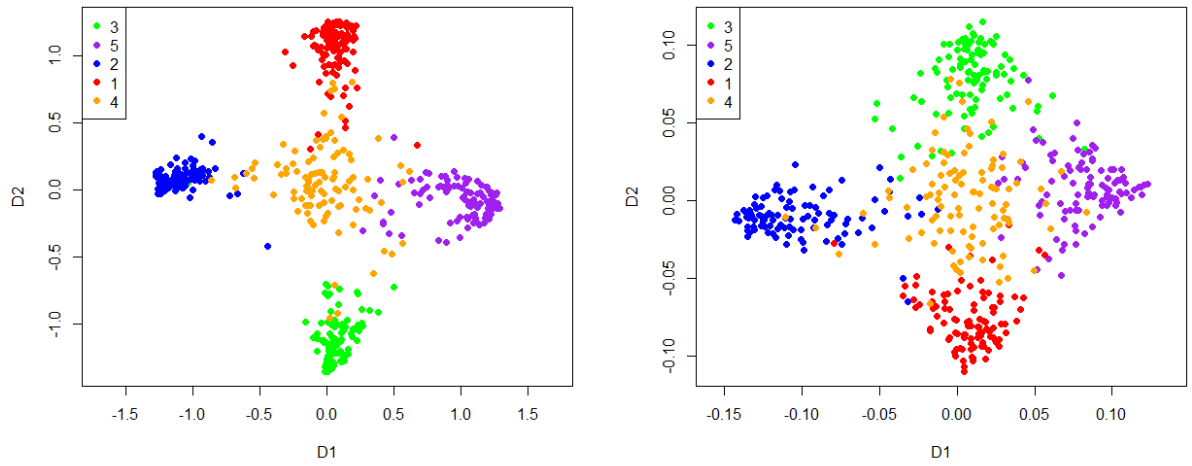

Figure 13: Local MDS Embeddings for simulated data. Left: low noise ($\sigma = 1$, k=5, c = 0.001). Right: high noise ($\sigma = 2$, c = 0.00001, k = 10)

4.3 Application: Country Characteristics

After assessing and analyzing the comparative performance of both t-SNE and MDS on the well known MNIST dataset and a simulated data set, we will now apply both methods in a new context of a high dimensional country comparison. The data is obtained from the World Government Summit 2019, World Visualization Prize, which is organized in collaboration with the organization Information Is Beautiful [4]. The data consists of 195 countries around the world. For each country, 32 different variables are recorded that characterize the country. These variables range from GDP to happiness scores and government effectiveness. In Appendix F, a list with all countries, variables and sources is provided. The main goal of this application is to investigate whether t-SNE and MDS can create visualizations that help to distinguish clusters and structure in the country data set, to gain insights in the similarity structure between countries.

Since the data set has a significant number of missing values (10.8%), we continue to describe how the final input for the methods is derived. The decision is made that if there are five or more

[4]https://informationisbeautiful.net/wdvp/

missing values for a country, the country is deleted from the list. This results in the deletion of 18 countries. From this set of 32 variables, 73 values out of 159 are missing for the variables Education Spending (% of GDP) and Education Spending ($ per capita) respectively. As such, these two variables are not considered. Due to the high number of missing values, imputation is not a reliable option. After the deletion of these two variables, the two happiness score variables have the highest number of missing values: 27. The countries with these missing values mainly include small pacific islands and African nations with low freedom and high levels of violence (e.g. Mali). Therefore, for these countries the mean of the lowest 50% happiness scores is imputed. For the other variables, the mean value of all remaining countries is inserted. This leaves us with a set of 159 countries.

Since the data set is smaller than the other two, it becomes computationally feasible to run a grid search to find the optimal values for the learning rate and perplexity. The perplexity is chosen from $Perp \in \{5, 10, 15, ..., 50\}$ and the learning rate is chosen from $\eta \in \{10, 30, 50, ..490\}$.

Whereas the Kullback Leibler divergence seems to be decreasing as the perplexity increases, the overlap coefficient is not much influenced by this. For a wide range of perplexity values, the overlap coefficient is around 55%. As Van Der Maaten and Hinton (2008) argue, the value for the perplexity should be chosen higher, as the data set increases in the number of observations. Since we are dealing with a relatively small data set, we now prefer to chose a lower value for the perplexity. Figure 14, shows two embeddings corresponding to a lower and higher chosen value for the perplexity. Note that for the sake of convergence, the number of iterations for the country data set is increased to 2000. Larger plots can be found in Appendix G for a more detailed overview.

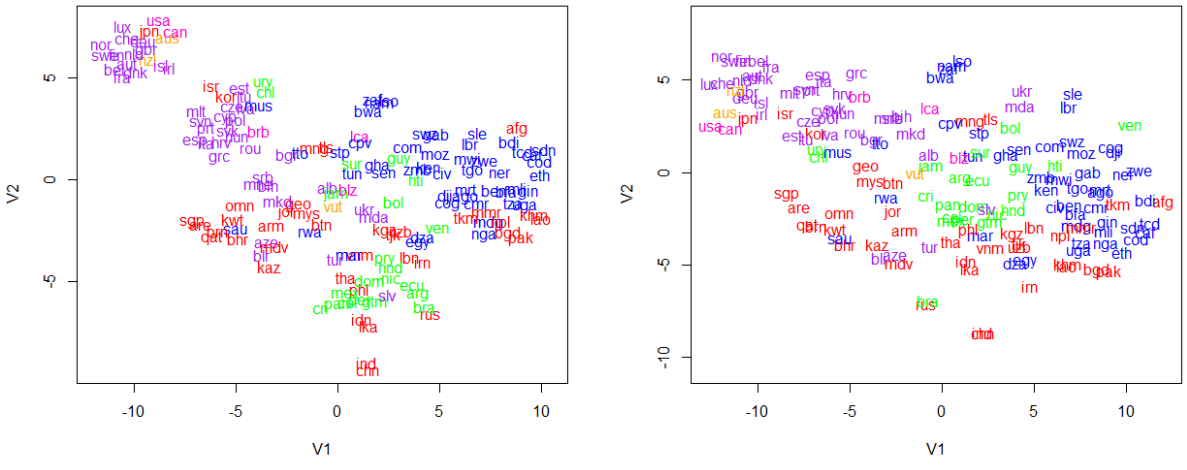

Figure 14: t-SNE embeddings for country data set. Left: perplexity = 10, learning rate = 250. Right: perplexity = 30, learning rate = 30

The colours correspond to the continents a country belongs to. To maximize the comparability of the results, we use a Procrustean rotation of the high perplexity result. This rotation aims to represent one embedding as close as possible to another embedding, using the distance preserving characteristic of rotations. The **procrustes** method from the **SMACOF** package is used to achieve this. It can be seen from the plots that the left embedding shows a stronger cluster structure. However, the rest of the structure seems to be very similar between both plots. Both embeddings tend to place western and northern European countries close together with the United States, Canada and Australia. Moreover, there seems to be a close link between some Middle Eastern countries, Singapore and eastern European countries like Azerbaijan. Closer inspection shows that these countries are characterized by relatively small populations, small surface area and high income. Hence, the t-SNE method indeed seems to be able to give structure to the data.

For the MDS approach, MDS with power weights (with $q = 1$, hence equivalent to Sammon mapping) and Local MDS (with $\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and δ_{ij} to create weights) give the best results for the overlap coefficient. Power weight MDS attains an average overlap coefficient of 0.484 and Local MDS (with $k = 5$ and $c = 0.01$) attains an average overlap coefficient of 0.468. These values are lower than the t-SNE embeddings and hence MDS seems less able to retain the local structure of the data. When inspecting the embeddings (Figure 15), we can see that even though the average overlap coefficient is similar for both embeddings, the Local MDS approach is able to let the countries drift apart further. A clear cluster of north/western European countries is visible, which is very similar to a cluster found in the t-SNE embedding. For a more detailed overview, larger plots are provided in Appendix G.

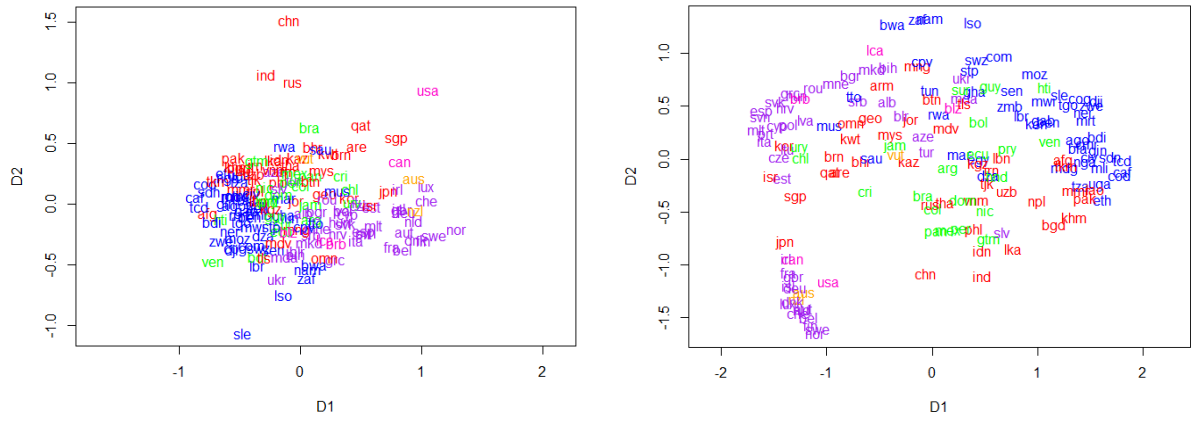

Figure 15: MDS embedding country data set. Left: power weights ($q = 1$). Right: Local MDS ($k = 5$ and $c = 0.01$)

5 Conclusion and Discussion

The purpose of this paper was twofold. Firstly the relationship and comparison between two highly popular data visualization techniques, MDS and t-SNE, was researched, as a comparison of the two methods was lacking in the literature. In addition, the methods were introduced in a new context to assess whether they can be used as a powerful tool for cross country comparisons in many policy fields. In this respect, the paper aims to answer to the research question "How does the performance of t-SNE compare to MDS in the field of visualizing high dimensional country similarities?".

With regards to the comparison between various forms of MDS and t-SNE, this paper shows that Local MDS can be used as an alternative for t-SNE in some situations. Interestingly, introducing the similarity matrix from t-SNE as input for the weights in MDS greatly improves the performance of the MDS embeddings. This is an interesting result that might improve MDS implementations in other settings as well. The preservation of the local neighbourhood, however, is superior when one considers the t-SNE method. For the MNIST data set, t-SNE seems to be better at generating clearly separated clusters of data, but Van Der Maaten and Hinton's (2008) observation that MDS-like methods are not able to separate the clusters in the low dimensional embedding must be nuanced. Even though the MDS approaches with power, k-nearest neighbour and kernel weighting functions are indeed not well able to separate the clusters, Local MDS is able to do so and at the same time retain the local structure of the data in a superior fashion to the other MDS approaches. Also for this, however, it holds that t-SNE is superior when it comes to retaining the local structure as measured by the average overlap coefficient. For the simulated data set, we can conclude that MDS and t-SNE can both yield good well separated representations of the initial clusters. Overall, though, t-SNE clearly performs better for all data sets when we consider the average overlap coefficient. Using this metric, we can conclude that t-SNE is better able than various MDS approaches to retain the high dimensional local structure in the low dimensional embedding.

When considering the second part of the research question, we see that the implementation of t-SNE and MDS leads to interesting insights with respect to the comparison of different countries, based on 30 variables. Even though this paper does not aim to find new relationships between countries, some well known groups of similar countries can be identified. As was mentioned in the previous section, the northern European countries are represented close together, and also countries like Italy, Portugal and Spain, who are often considered similar in many respects, are represented close together. As such, we can conclude that t-SNE and MDS can indeed be used to represent a

high dimensional data set of country characteristics in a 2D map. This analysis, however, is only based on 30 high level macroeconomic variables. The advantage of using this broad data set is that patterns in this data are well known and hence can be easily verified by glancing over the embedding. For the final conclusion on how powerful these methods are in this field and how policy advisors can use these data visualization methods, it is interesting to look at other data sets in the future that are highly specific with respect to the policy area they attempt to address.

One of the main limitations of this paper is that the specific relationship between the penalty coefficient, c, and the embeddings from Local MDS is not thoroughly researched. Moreover, it is not entirely clear what the effect of the simplification of Chen and Buja's (2009) implementation is. In their work, Chen and Buja (2009) take a more complex approach to model the weights and corresponding penalized dissimilarities. It would therefore be interesting for future research to investigate whether this more complex formulation of Local MDS yields different results in the context of the data sets considered in this paper, espacially because Local MDS yields superior results than the other MDS implementations.

Moreover, the interpretability of the results with regards to the simulated data set is subject to some limitations. As has been described in the paper, the first step of the construction of the high dimensional data preserves the distances from the low dimension. Only after contaminating the data with Gaussian noise, MDS will not yield the exact initial low dimensional representation of the clusters. When the distances stay exactly equal in the high dimension, this means that the data actually lies on a manifold that has an intrinsic dimension lower that the dimensionality of the data. This paper then only considers two levels of Gaussian noise. The results thus, have to be interpreted in this context, where the noise induces the data to 'move away' from the manifold. This means that the comparison of t-SNE and MDS for the simulated data set is mainly applicable to data sets that are already close to a lower dimensional manifold. In many cases, this might be the case, but for future research it might be interesting to look at data that has been constructed in a different fashion to extend the analysis presented in this paper. Lastly, the average overlap coefficient M is determined only for neighbourhood size 10. In the future, robustness checks for this should be provided.

In summary, this paper presents a more nuanced comparison between MDS and t-SNE than previous literature has provided. Even though t-SNE's performance in retaining local structure is once again confirmed, we have seen that Local MDS is able to yield similar embeddings that show a cluster structure in the data.

References

Anderson, G., & Hussey, P. S. (2001). Comparing health system performance in oecd countries. *Health Affairs*, *20*(3), 219–232.

Bellman, R. E., & Dreyfus, S. E. (1962). *Applied dynamic programming*. Princeton university press.

Chen, L., & Buja, A. (2009). Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, *104*(485), 209–219.

Cox, M. A., & Cox, T. F. (2008). Multidimensional scaling. In *Handbook of data visualization* (pp. 315–347). Springer.

De Leeuw, J., & Heiser, W. J. (1980). Multidimensional scaling with restrictions on the configuration.

De Leeuw, J., & Mair, P. (2009). Multidimensional scaling using majorization: Smacof in r. *Journal of Statistical Software*, *31*(3), 1–30.

Gashi, I., Stankovic, V., Leita, C., & Thonnard, O. (2009). An experimental study of diversity with off-the-shelf antivirus engines. *2009 Eighth IEEE International Symposium on Network Computing and Applications*, 4–11.

Gisbrecht, A., Schulz, A., & Hammer, B. (2015). Parametric nonlinear dimensionality reduction using kernel t-sne. *Neurocomputing*, *147*, 71–82.

Groenen, P. J., & Van De Velden, M. (2016). Multidimensional scaling by majorization: A review. *Journal of Statistical Software*, *73*(8), 1–26.

Hamel, P., & Eck, D. (2010). Learning features from music audio with deep belief networks. In *Ismir* (Vol. 10, pp. 339–344).

Hinton, G. E., & Roweis, S. T. (2003). Stochastic neighbor embedding. In *Advances in neural information processing systems* (pp. 857–864).

Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. *Neural networks*, *1*(4), 295–307.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, *22*(1), 79–86.

Lee, J. A., & Verleysen, M. (2007). *Nonlinear dimensionality reduction*. Springer Science & Business Media.

Li, W., Cerise, J. E., Yang, Y., & Han, H. (2017). Application of t-sne to human genetic data. *Journal of bioinformatics and computational biology*, *15*(04), 1750017.

Saggi, K., Maskus, K. E., & Hoekman, B. (2004). *Transfer of technology to developing countries: Unilateral and multilateral policy options*. The World Bank.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, *100*(5), 401–409.

Saul, L. K., Weinberger, K. Q., Ham, J. H., Sha, F., & Lee, D. D. (2006). Spectral methods for dimensionality reduction. *Semisupervised learning*, 293–308.

Van Der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, *9*(Nov), 2579–2605.

Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, *10*(66-71), 13.

Van Eck, N. J., Waltman, L., Dekker, R., & van den Berg, J. (2010). A comparison of two techniques for bibliometric mapping: Multidimensional scaling and vos. *Journal of the American Society for Information Science and Technology*, *61*(12), 2405–2416.

6 Appendix

6.1 Appendix A: Overview MDS Implementations

Table 1: Overview MDS implementations

Name	**Weight**		**Input**
Power Weights	$w_{ij} = \delta_{ij}^q$	$q = \{-10, -9, \ldots, 9, 10\}$	$\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$
k-NN	$w_{ij} = \begin{cases} 1 & \text{if } j \text{ is in the } k \text{ NN of } i \\ 0 & \text{otherwise.} \end{cases}$	$k \in \{5, 10, \ldots, 25, 30\}$	$\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$
Kernel	$w_{ij} = e^{-\delta_{ij}^2}$		$\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$
Local MDS	$w_{ij} = \begin{cases} 1 & \text{if } j \text{ is in the } k \text{ NN of } i \\ c & \text{otherwise.} \end{cases}$	$c \in \{0.01, 0.001, 0.0001, 0.00001\}$	$\delta_{ij}^* = \begin{cases} \delta_{ij} & \text{if } j \text{ is in the } k \text{ NN of } i \\ \frac{\delta_{ij}}{c} & \text{otherwise.} \end{cases}$
Power Weights	$w_{ij} = (1 - p_{ij})^q$	$q = \{-10, -9, \ldots, 9, 10\}$	$\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$
k-NN	$w_{ij} = \begin{cases} 1 & \text{if } j \text{ is in the } k \text{ NN of } i \\ 0 & \text{otherwise.} \end{cases}$	$k \in \{5, 10, \ldots, 25, 30\}$	$\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$
Kernel	$w_{ij} = e^{-(1-p_ij)^2}$		$\delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$
Local MDS	$w_{ij} = \begin{cases} 1 & \text{if } j \text{ is in the } k \text{ NN of } i \\ c & \text{otherwise.} \end{cases}$	$c \in \{0.01, 0.001, 0.0001, 0.00001\}$	$\delta_{ij}^* = \begin{cases} \delta_{ij} & \text{if } j \text{ is in the } k \text{ NN of } i \\ \frac{\delta_{ij}}{c} & \text{otherwise.} \end{cases}$
Power Weights	$w_{ij} = (1 - p_{ij})^q$	$q = \{-10, -9, \ldots, 9, 10\}$	$\delta_{ij} = 1 - p_{ij}$
k-NN	$w_{ij} = \begin{cases} 1 & \text{if } j \text{ is in the } k \text{ NN of } i \\ 0 & \text{otherwise.} \end{cases}$	$k \in \{5, 10, \ldots, 25, 30\}$	$\delta_{ij} = 1 - p_{ij}$
Kernel	$w_{ij} = e^{-(1-p_{ij})^2}$		$\delta_{ij} = 1 - p_{ij}$
Local MDS	$w_{ij} = \begin{cases} 1 & \text{if } j \text{ is in the } k \text{ NN of } i \\ c & \text{otherwise.} \end{cases}$	$c \in \{0.01, 0.001, 0.0001, 0.00001\}$	$\delta_{ij}^* = \begin{cases} \delta_{ij} & \text{if } j \text{ is in the } k \text{ NN of } i \\ \frac{\delta_{ij}}{c} & \text{otherwise.} \end{cases}$

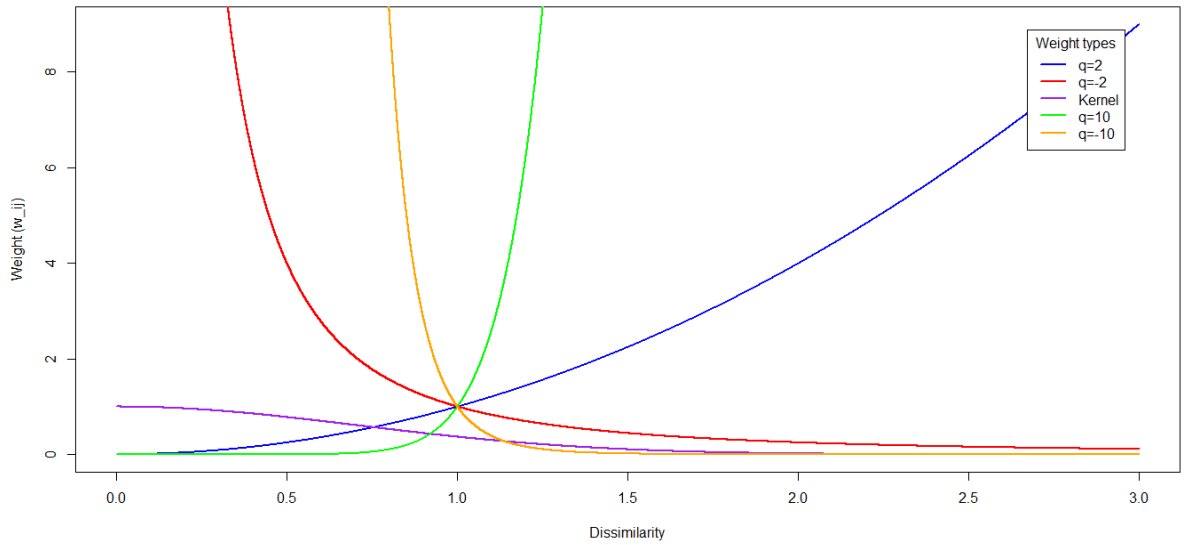

Figure 16: Weights as function of the dissimilarity for power weights ($q = 2$, $q = 10$, $q = -2$, $q = -10$) and Kernel weights

6.2 Appendix B: Comparison R and MATLAB implementations

Table 2: Initialization comparison R and MATLAB Implementations for t-SNE for the IRIS data set

n	y_1	y_2	n	y_1	y_2	n	y_1	y_2	n	y_1	y_2	n	y_1	y_2
1	-1.78E-05	-0.00011	31	1.01E-05	0.000118	61	-0.00011	4.82E-05	91	0.000127	7.49E-05	121	-0.00015	0.000155
2	-4.11E-05	0.000134	32	-0.00019	2.81E-05	62	2.28E-05	4.41E-05	92	-0.00016	3.02E-05	122	-0.00012	3.92E-05
3	2.10E-05	-5.41E-05	33	-3.91E-05	9.94E-05	63	-0.00019	-2.45E-05	93	3.47E-05	-4.41E-05	123	6.03E-06	0.000209
4	-0.00015	-4.29E-05	34	-8.01E-05	-6.45E-05	64	3.78E-05	9.08E-05	94	-1.48E-05	0.000105	124	-6.95E-05	-8.44E-05
5	0.000113	-0.00011	35	-0.00019	-0.00024	65	-0.0001	0.000166	95	8.01E-05	-2.99E-05	125	0.000115	7.42E-05
6	-3.03E-05	4.86E-05	36	-2.86E-06	-8.90E-05	66	-5.60E-05	-0.00022	96	8.19E-05	0.000118	126	6.27E-05	-7.64E-05
7	-2.26E-05	-0.00021	37	4.44E-05	-0.00013	67	-0.00013	0.00016	97	-0.00018	3.01E-05	127	6.46E-05	7.12E-05
8	4.49E-05	-5.04E-06	38	0.000148	-1.84E-05	68	4.91E-05	-1.80E-05	98	-0.00011	0.00024	128	-2.42E-05	-0.00013
9	0.000189	-0.00017	39	-5.50E-05	-0.00017	69	-0.00016	-1.13E-05	99	-9.00E-05	0.000258	129	-5.66E-05	7.15E-05
10	4.16E-05	0.000113	40	9.02E-05	-3.04E-05	70	2.03E-05	-1.26E-05	100	-3.10E-05	2.10E-05	130	8.18E-05	-1.75E-05
11	1.33E-05	6.30E-05	41	-2.76E-05	-0.00016	71	1.44E-05	-8.73E-06	101	2.17E-05	-7.89E-05	131	3.83E-05	-0.00017
12	0.000105	2.21E-05	42	0.000154	2.83E-05	72	-0.00017	-6.38E-05	102	7.76E-05	-0.00012	132	-6.29E-05	-1.19E-05
13	-0.00014	-5.79E-05	43	7.67E-05	0.000108	73	-2.33E-05	8.53E-05	103	-7.31E-05	-0.00011	133	-0.00014	-2.52E-05
14	0.000176	6.35E-05	44	0.000158	7.90E-07	74	2.32E-05	-6.78E-05	104	4.08E-05	-9.29E-05	134	-0.00014	2.07E-05
15	0.000129	1.99E-05	45	0.000312	3.54E-05	75	5.18E-05	-6.93E-05	105	8.47E-05	8.46E-05	135	8.81E-06	-0.00014
16	7.32E-05	1.72E-05	46	6.55E-05	-9.06E-05	76	-3.76E-05	-6.22E-06	106	-8.54E-06	-2.30E-05	136	9.26E-06	1.08E-05
17	1.86E-06	0.000151	47	5.28E-05	8.43E-05	77	0.000199	8.23E-05	107	8.86E-05	5.92E-05	137	0.000186	-5.46E-05
18	-0.0001	7.94E-05	48	-5.04E-05	6.33E-05	78	-6.35E-06	1.47E-05	108	-4.27E-05	6.86E-05	138	0.000194	0.000216
19	0.000184	-5.13E-05	49	-3.15E-05	0.000216	79	-1.50E-05	-7.97E-05	109	-3.13E-05	-5.50E-05	139	0.000211	7.75E-05
20	3.17E-05	6.00E-05	50	0.00013	-4.68E-05	80	9.55E-06	-7.89E-05	110	0.000207	-0.00015	140	5.95E-05	-8.20E-05
21	0.00022	6.19E-06	51	4.71E-05	-0.00015	81	-4.92E-05	7.67E-06	111	0.000129	-1.57E-05	141	-3.57E-05	1.80E-05
22	-9.21E-05	-1.43E-05	52	-4.96E-05	-8.87E-05	82	5.38E-06	0.000104	112	6.99E-05	-0.0002	142	-5.99E-05	-0.00015
23	8.75E-05	-1.17E-05	53	-4.41E-05	1.13E-05	83	-6.52E-05	-0.0001	113	4.02E-05	-3.67E-05	143	-8.63E-05	-1.47E-05
24	-6.31E-05	0.000132	54	4.03E-05	8.57E-05	84	-0.00018	3.26E-05	114	9.39E-05	0.000182	144	-7.02E-05	0.000151
25	6.55E-06	6.80E-05	55	2.03E-05	-3.60E-05	85	4.69E-05	-4.46E-05	115	-2.45E-05	-9.34E-05	145	7.84E-06	-0.00012
26	-7.43E-05	2.79E-05	56	-5.57E-05	0.000122	86	1.51E-06	-3.72E-05	116	0.000154	-0.00014	146	-6.40E-05	5.57E-05
27	-9.69E-05	-5.48E-05	57	-9.50E-05	0.000163	87	-8.79E-05	1.46E-05	117	9.83E-05	6.53E-05	147	-0.00024	-0.00012
28	1.67E-05	2.75E-05	58	0.000128	-0.00011	88	-0.00013	-6.61E-05	118	-8.75E-05	4.28E-05	148	2.51E-05	9.42E-05
29	-3.14E-05	-0.0003	59	0.000251	-8.91E-05	89	4.39E-05	0.000127	119	-6.56E-05	0.000248	149	1.33E-05	-3.01E-05
30	0.000111	-0.00015	60	-3.40E-05	5.31E-05	90	4.69E-05	0.000145	120	-9.44E-05	-2.44E-05	150	-0.00018	6.04E-06

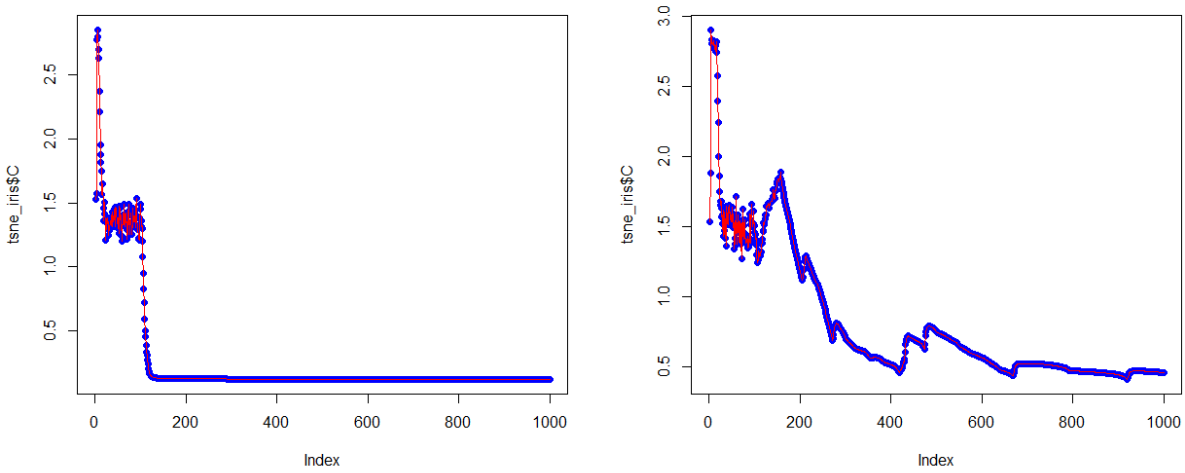

Figure 17: Value of Kullback-Leibler divergence for learning rates 500 (left) and 1000 (right)

6.3 Appendix C: MDS Embeddings MNIST data set

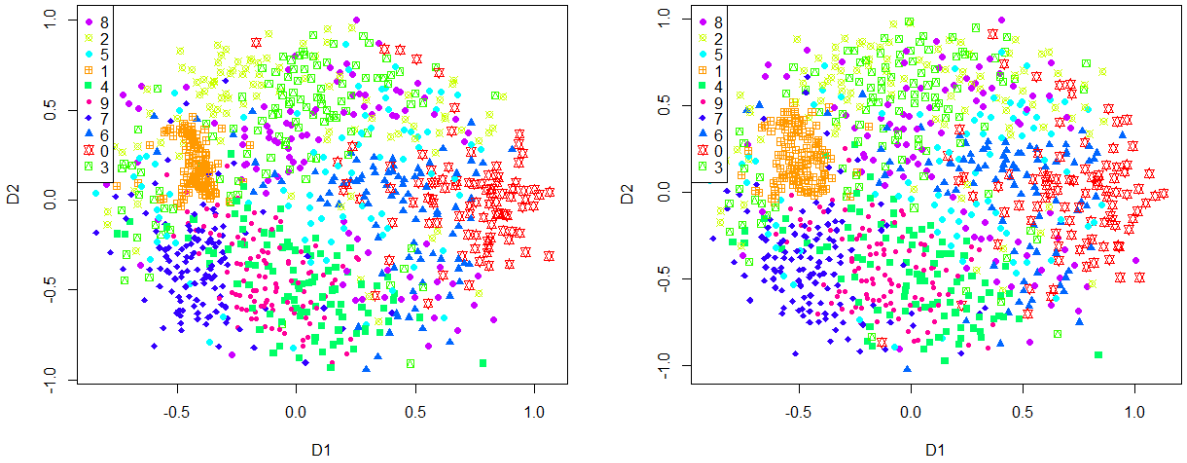

Figure 18: MNIST MDS Embeddings for Power weights: $q = 2$. Left: Euclidean distance as input for weights. Right: $\mathbf{P}$ as input for weights

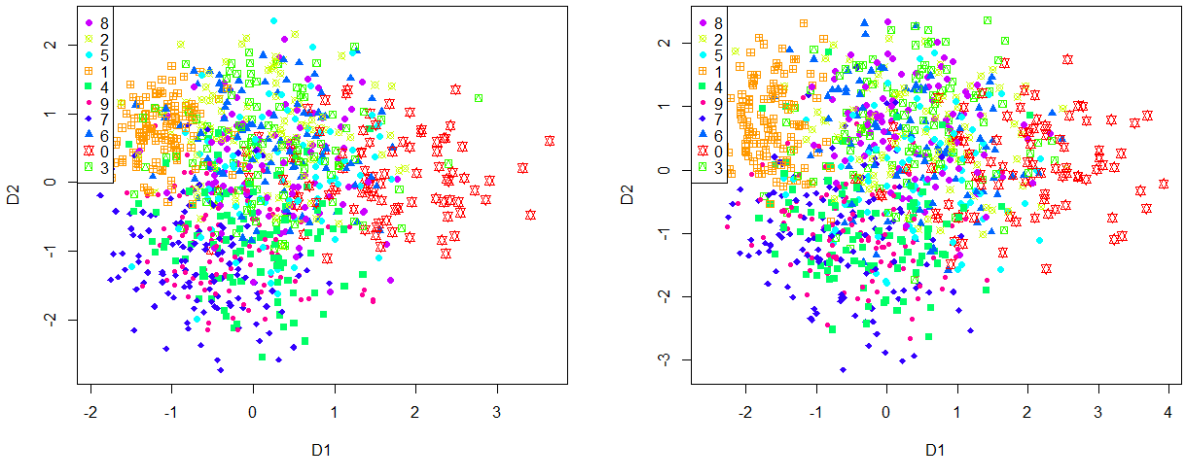

Figure 19: MNIST MDS Embeddings for k-nearest neighbours: $k = 5$. Left: Euclidean distance as input for weights. Right: $\mathbf{P}$ as input for weights

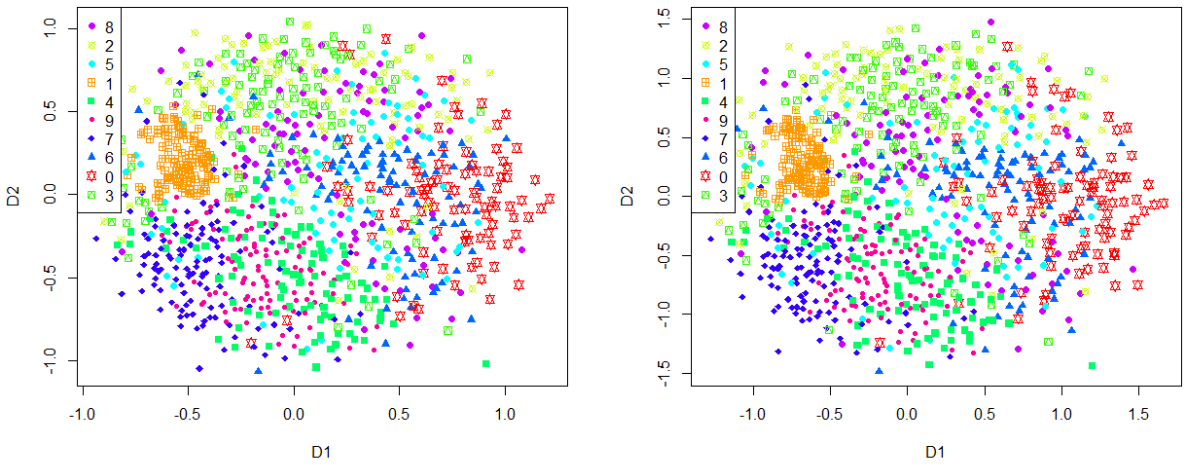

Figure 20: MNIST MDS Embeddings for Kernel type weights. Left: Euclidean distance as input for weights. Right: $\mathbf{P}$ as input for weights

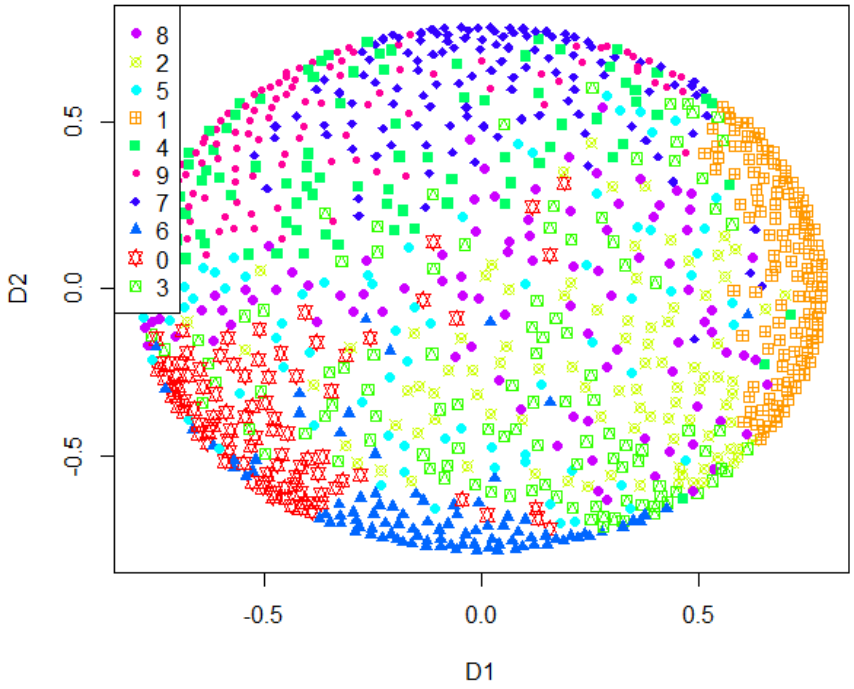

Figure 21: MNIST MDS Embedding using $\mathbf{P}$ as input for weighting function and dissimilarity matrix

6.4 Appendix D: Orientation t-SNE for Simulated Data

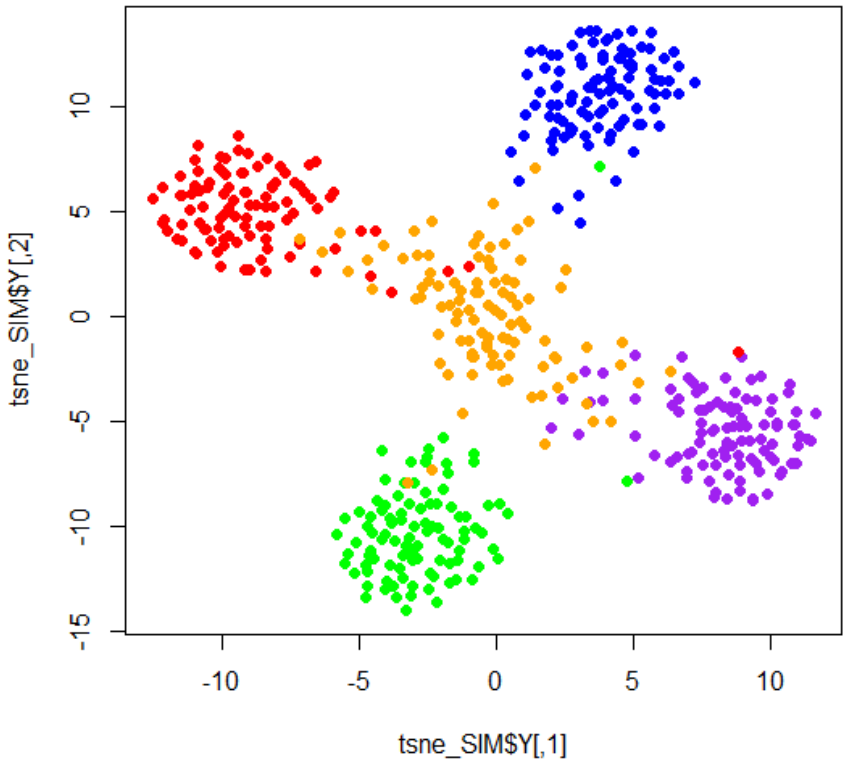

Figure 22: t-SNE does not preserve the orientation among the clusters

6.5 Appendix E: Simulated Data Local MDS

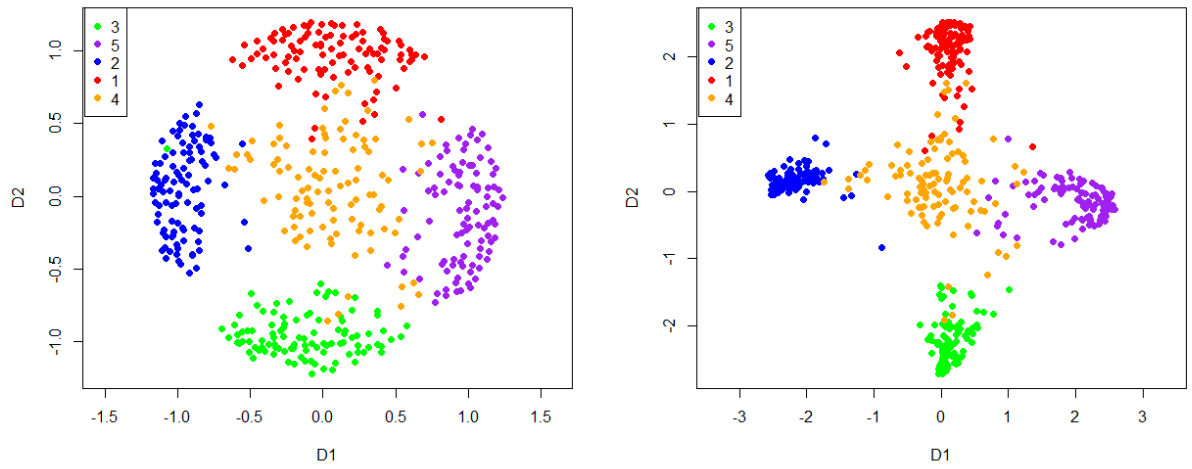

Figure 23: Local MDS implementation for Simulated data. Left: $k = 5$, $c = 0.01$. Right: $k = 5$, $c = 0.001$

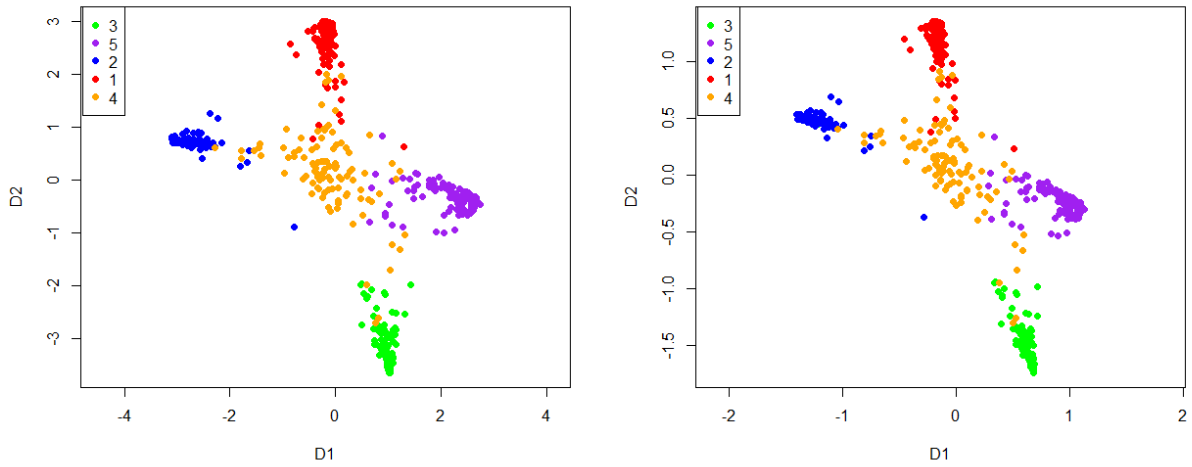

Figure 24: Local MDS implementation for Simulated data. Left: $k = 5$, $c = 0.0001$. Right: $k = 5$, $c = 0.00001$

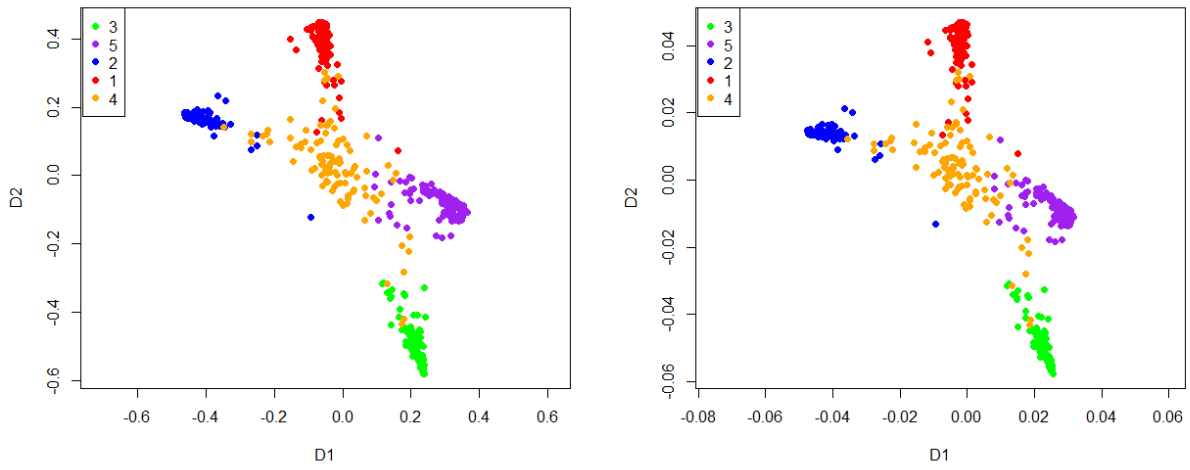

Figure 25: Local MDS implementation for Simulated data. Left: $k = 5$, $c = 0.000001$. Right: $k = 5$, $c = 0.00000001$

6.6 Appendix F: Country Data Information

Table 3: Variables in the country data set

Variable	Source	Mean	Std	Min	Max	Median	#NA
Population	World Bank	38518215.38	142562477.71	11000.00	1386000000.00	9000000.00	0.00
Surface Area	CIA World Factbook	662406.32	1832712.93	2.00	16377742.00	111890.00	0.00
GINI Index	CIA World Factbook	39.13	8.75	23.20	63.20	38.55	47.00
Happy Planet Index	Happy Planet Index	26.50	7.26	12.80	44.70	26.40	58.00
Human Develoment Index	UNDP	0.71	0.15	0.35	0.95	0.74	9.00
World Happiness Score	World Happiness Report	5.50	1.13	2.66	7.79	5.60	57.00
Sustainable Economic Development Assessment	Boston Consulting Group (BCG)	51.88	16.90	16.10	85.30	49.50	45.00
GDP	Heritage Foundation	649.04	2260.74	0.20	21291.80	73.20	11.00
GDP per capita	Heritage Foundation	20061.17	23252.97	652.00	160526.00	12262.00	14.00
GDP growth	Heritage Foundation	2.41	4.18	-28.10	10.20	2.95	11.00
Health Expenditure (% GDP)	World Bank	6.80	2.98	2.03	22.12	6.27	7.00
Health Expenditure ()	World Bank	1373.74	1675.21	32.00	9536.00	762.00	12.00
Education Expenditure (% GDP)	World Bank	4.69	1.52	1.02	7.68	4.84	106.00
Education Expenditure (*percapita*)	World Bank	951.50	1371.08	2.56	7465.68	314.31	108.00
School Life Expectancy	CIA World Factbook	13.35	3.16	5.00	20.00	13.00	42.00
Unemployment Rate	Heritage Foundation	8.58	6.29	0.20	31.40	6.60	16.00
Government Spending Score	Heritage Foundation	63.23	23.24	0.00	96.30	69.10	14.00
Government Expenditures (% GDP)	Heritage Foundation	33.67	13.11	11.00	117.60	32.05	15.00
Political Rights Score	Freedom House	3.43	2.19	1.00	7.00	3.00	0.00
Civil Liberties Score	Freedom House	3.33	1.93	1.00	7.00	3.00	0.00
Political Stability and Absence of violence	World Bank	-0.07	0.99	-2.96	1.65	0.03	1.00
Government Effectiveness	World Bank	-0.07	0.99	-2.48	2.21	-0.17	3.00
Regulatory Quality	World Bank	-0.07	0.98	-2.34	2.12	-0.19	3.00
Rule of Law	World Bank	-0.08	0.99	-2.31	2.03	-0.23	3.00
Control of corruption	World Bank	-0.07	1.00	-1.83	2.24	-0.27	3.00
Judicial Effectiveness Score	Heritage Foundation	46.64	20.17	5.00	93.80	44.50	12.00
Government Integrity Score	Heritage Foundation	41.87	18.50	7.50	95.70	36.80	12.00
Property Rights Score	Heritage Foundation	51.24	19.87	5.20	98.40	49.40	12.00
Tax Burden Score	Heritage Foundation	76.60	13.43	0.00	99.90	78.50	16.00
Overall Economic Freedom Score	Heritage Foundation	60.86	10.99	5.80	88.80	61.20	17.00
Financial Freedom Score	Heritage Foundation	48.38	19.32	0.00	90.00	50.00	16.00
Women MPs (%)	World Bank	21.34	11.78	0.00	61.30	20.00	2.00

The data can be retrieved from https://informationisbeautiful.net/wdvp/. Here, more information is provided on the data set.

Table 4: Overview Countries and Country Codes

Afghanistan	AFG	**Dominican Republic**	DOM	**Liberia**	LBR	**Saint Vincent and the Grenadines**	VCT
Albania	ALB	**Ecuador**	ECU	**Libya**	LBY	**Samoa**	WSM
Algeria	DZA	**Egypt**	EGY	**Liechtenstein**	LIE	**San Marino**	SMR
Andorra	AND	**El Salvador**	SLV	**Lithuania**	LTU	**Sao Tome and Principe**	STP
Angola	AGO	**Equatorial Guinea**	GNQ	**Luxembourg**	LUX	**Saudi Arabia**	SAU
Antigua & Barbuda	ATG	**Eritrea**	ERI	**Macedonia**	MKD	**Senegal**	SEN
Argentina	ARG	**Estonia**	EST	**Madagascar**	MDG	**Serbia**	SRB
Armenia	ARM	**Eswatini**	SWZ	**Malawi**	MWI	**Seychelles**	SYC
Australia	AUS	**Ethiopia**	ETH	**Malaysia**	MYS	**Sierra Leone**	SLE
Austria	AUT	**Fiji**	FJI	**Maldives**	MDV	**Singapore**	SGP
Azerbaijan	AZE	**Finland**	FIN	**Mali**	MLI	**Slovakia**	SVK
Bahamas	BHS	**France**	FRA	**Malta**	MLT	**Slovenia**	SVN
Bahrain	BHR	**Gabon**	GAB	**Marshall Islands**	MHL	**Solomon Islands**	SLB
Bangladesh	BGD	**Gambia, The**	GMB	**Mauritania**	MRT	**Somalia**	SOM
Barbados	BRB	**Georgia**	GEO	**Mauritius**	MUS	**South Africa**	ZAF
Belarus	BLR	**Germany**	DEU	**Mexico**	MEX	**South Sudan**	SSD
Belgium	BEL	**Ghana**	GHA	**Micronesia**	FSM	**Spain**	ESP
Belize	BLZ	**Greece**	GRC	**Moldova**	MDA	**Sri Lanka**	LKA
Benin	BEN	**Grenada**	GRD	**Monaco**	MCO	**Sudan**	SDN
Bhutan	BTN	**Guatemala**	GTM	**Mongolia**	MNG	**Suriname**	SUR
Bolivia	BOL	**Guinea**	GIN	**Montenegro**	MNE	**Sweden**	SWE
Bosnia and Herzegovina	BIH	**Guinea-Bissau**	GNB	**Morocco**	MAR	**Switzerland**	CHE
Botswana	BWA	**Guyana**	GUY	**Mozambique**	MOZ	**Syria**	SYR
Brazil	BRA	**Haiti**	HTI	**Myanmar**	MMR	**Taiwan**	TWN
Brunei	BRN	**Honduras**	HND	**Namibia**	NAM	**Tajikistan**	TJK
Bulgaria	BGR	**Hungary**	HUN	**Nauru**	NRU	**Tanzania**	TZA
Burkina Faso	BFA	**Iceland**	ISL	**Nepal**	NPL	**Thailand**	THA
Burundi	BDI	**India**	IND	**Netherlands**	NLD	**Timor-Leste**	TLS
Cabo Verde	CPV	**Indonesia**	IDN	**New Zealand**	NZL	**Togo**	TGO
Cambodia	KHM	**Iran**	IRN	**Nicaragua**	NIC	**Tonga**	TON
Cameroon	CMR	**Iraq**	IRQ	**Niger**	NER	**Trinidad and Tobago**	TTO
Canada	CAN	**Ireland**	IRL	**Nigeria**	NGA	**Tunisia**	TUN
Central African Republic	CAF	**Israel**	ISR	**Norway**	NOR	**Turkey**	TUR
Chad	TCD	**Italy**	ITA	**Oman**	OMN	**Turkmenistan**	TKM
Chile	CHL	**Jamaica**	JAM	**Pakistan**	PAK	**Tuvalu**	TUV
China	CHN	**Japan**	JPN	**Palau**	PLW	**Uganda**	UGA
Colombia	COL	**Jordan**	JOR	**Panama**	PAN	**Ukraine**	UKR
Comoros	COM	**Kazakhstan**	KAZ	**Papua New Guinea**	PNG	**United Arab Emirates**	ARE
Congo (Dem. Rep.)	COD	**Kenya**	KEN	**Paraguay**	PRY	**United Kingdom**	GBR
Congo (Rep.)	COG	**Kiribati**	KIR	**Peru**	PER	**United States**	USA
Costa Rica	CRI	**Korea (Dem. People's Rep.)**	PRK	**Philippines**	PHL	**Uruguay**	URY
Cote d'Ivoire	CIV	**Korea (Rep.)**	KOR	**Poland**	POL	**Uzbekistan**	UZB
Croatia	HRV	**Kosovo**	-	**Portugal**	PRT	**Vanuatu**	VUT
Cuba	CUB	**Kuwait**	KWT	**Qatar**	QAT	**Venezuela**	VEN
Cyprus	CYP	**Kyrgyzstan**	KGZ	**Romania**	ROU	**Vietnam**	VNM
Czech Republic	CZE	**Laos**	LAO	**Russia**	RUS	**Yemen**	YEM
Denmark	DNK	**Latvia**	LVA	**Rwanda**	RWA	**Zambia**	ZMB
Djibouti	DJI	**Lebanon**	LBN	**Saint Kitts and Nevis**	KNA	**Zimbabwe**	ZWE
Dominica	DMA	**Lesotho**	LSO	**Saint Lucia**	LCA		

6.7 Appendix G: Low Dimensional Embeddings Country Data

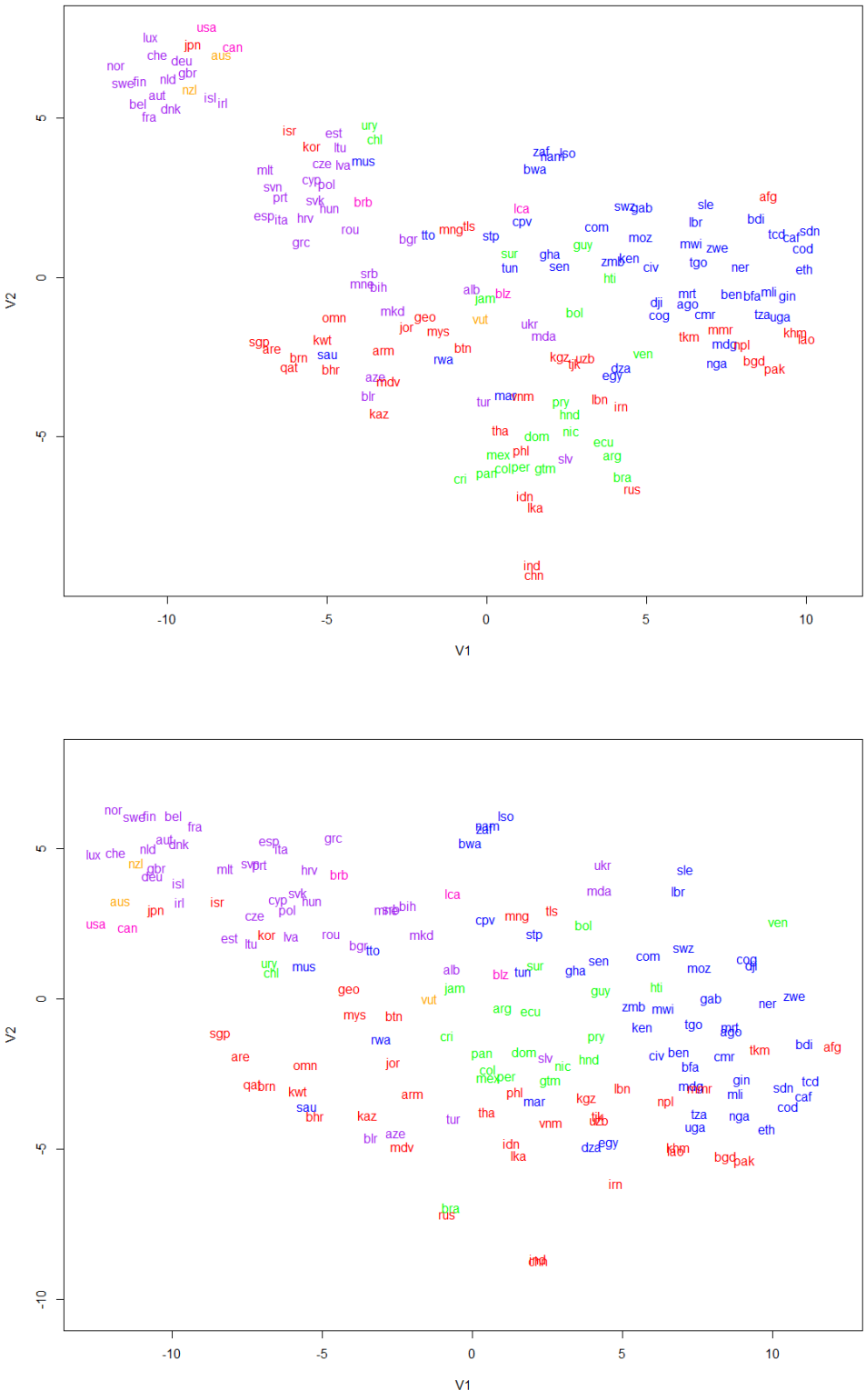

Figure 26: t-SNE embeddings for country data set. Top: perplexity = 10, learning rate = 250. Bottom: perplexity = 30, learning rate = 30

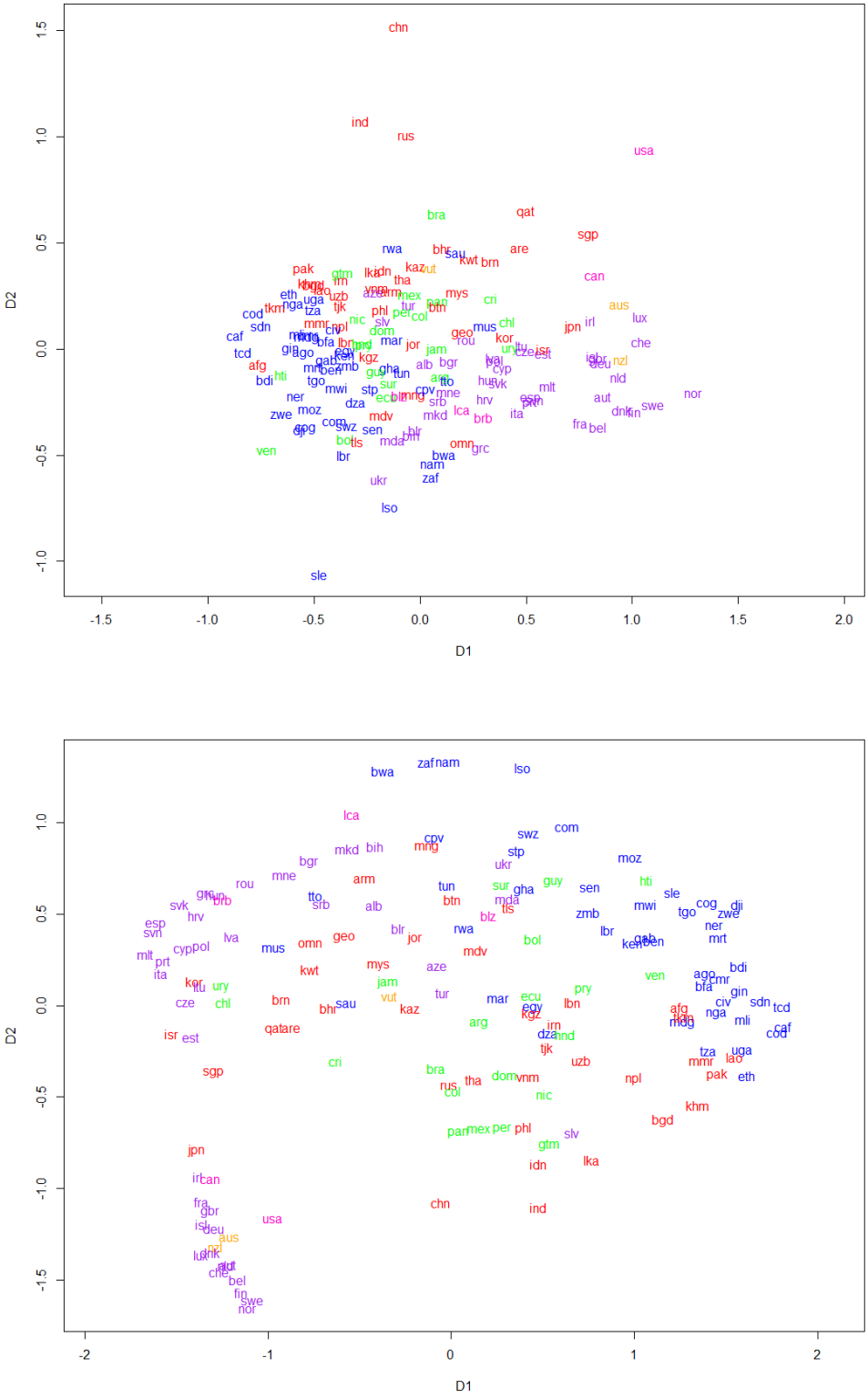

Figure 27: MDS embedding country data set. Top: power weights ($q = 1$). Bottom: Local MDS (k= 5 and $c = 0.01$)