Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Ehsan Amid

Application of α-Divergence for Stochastic Neighbor Embedding in Data Visualization

Master's Thesis
Espoo, August 11, 2014

Supervisor:	Professor Erkki Oja, Aalto University
Advisors:	Onur Dikmen, D.Sc. (Tech)

Aalto University
School of Science

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author:	Ehsan Amid		
Title:			
Application of α-Divergence for Stochastic Neighbor Embedding in Data Visualization			
Date:	August 11, 2014	**Pages:**	68
Major:	Information and Computer Science	**Code:**	T-110
Supervisor:	Professor Erkki Oja		
Advisors:	Onur Dikmen, D.Sc. (Tech)		

Dimensionality reduction and information visualization are fundamental steps in data processing, information extraction and reasoning. In real-world applications, the number of measurements or variables per a single observation is so large that handling the raw data in a specific problem such as regression or classification becomes infeasible or even impractical. Moreover, in many applications, a faithful representation of the data for a first step analysis and hypothesis development becomes crucial. Recently, the SNE method has become tremendously popular for data visualization and feature extraction. The more recent algorithms such as t-SNE and HSSNE extend the basic SNE algorithm by considering general heavy-tailed distributions in the low-dimensional space, while the others, such as NeRV, consider different parameterized cost functions to achieve the desired embedding by tuning the parameter. In this thesis, we provide another extension to the SNE method by investigating the properties of α-divergence for neighbor embedding, focusing our attention on a particular range of α values. We show that α-divergence, with a proper selection of the α parameter effectively eliminates the crowding problem associated with the early methods. However, we also provide the extensions of our method to distributions having heavier tail than Gaussian. Contrary to some earlier methods like HSSNE and NeRV, no hand-tuning is needed, but we can rigorously estimate the optimal value of α for given input data. For this, we provide a statistical framework using a novel distribution called Exponential Divergence with Augmentation. This is an approximate generalization of Tweedie distribution and enables α-optimization after a non-linear transformation. We evaluate the performance of our proposed method by considering two sets of experiments: first, we provide a number of visualizations using our method and its extensions and compare the results with the earlier methods. Second, we conduct a set of experiments to confirm the effectiveness of our α-optimization method for finding the optimal α for the data distribution, and its consistency with standard quality measures of dimensionality reduction.

Keywords:	Dimensionality Reduction, Information Visualization, Stochastic Neighbor Embedding, α-Divergence, Exponential Divergence with Augmentation.
Language:	English

Acknowledgements

I would like to sincerely thank my instructor, Dr. Onur Dikmen, for his support and assistance throughout my Master's thesis, without whom this work would not have been possible. I would also like to express my sincere gratitude to my supervisor, Prof. Erkki Oja, for all his guidance and motivation as well as his precious comments on the work. It has been a great honor for me working under his supervision. Many thanks to my family and all my friends for supporting me during my studies and especially, writing my thesis. Last but not least, I would like to thank Aalto University and in particular, the Department of Information and Computer Science for providing the opportunity to pursue my Master's studies, here.

تقديم به مادر عزيزم

Espoo, August 11, 2014

Ehsan Amid

Contents

1 Introduction 10

1.1 Motivation . . . 10

1.2 Scope and Purpose of the Thesis . . . 11

1.3 Structure of the Thesis . . . 12

2 Dimensionality Reduction and Information Visualization 13

3 Information Divergence 21

3.1 Csiszár Divergences . . . 22

3.2 Bregman Divergence . . . 24

3.3 β-Divergence . . . 27

3.4 α-Divergence . . . 28

4 Stochastic Neighbor Embedding 30

4.1 Probabilistic Model of Neighborhood . . . 30

4.1.1 Conditional Similarities . . . 30

4.1.2 Symmetric Similarities . . . 31

4.1.3 Student t-Distribution and Distributions with Heavier Tails . . . 32

4.1.4 Random Walk Based Similarities . . . 32

4.2 Neighbor Embedding . . . 34

5 Stochastic Neighbor Embedding with α-Divergence 37

5.1 α-SNE Method . . . 37

5.2 Extension to Heavy-tailed Distributions . . . 41

5.3 α-Optimization . . . 43

5.3.1 Tweedie Distribution . . . 43

5.3.2 Exponential Divergence with Augmentation . . . 44

6 Experimental Results **46**
6.1 Visualization Results . . . 46
6.1.1 Datasets . . . 46
6.1.2 Results . . . 47
6.1.3 Fine-grained Analysis of the Embedding . . . 50
6.2 α-Optimization Results . . . 54
6.2.1 Quality Measures . . . 54
6.2.2 Datasets . . . 55
6.2.3 Results . . . 56

7 Conclusions and Future Work **60**

A Calculation of the Gradient **67**

List of Tables

3.1 Basic divergences expressed as the Csiszár f-divergence. . . . 23
3.2 Basic symmetric divergences expressed as the Csiszár f-divergence. 24
3.3 Basic divergences expressed as the Bregman divergence. 27
3.4 Special cases of β-divergence. 28
3.5 Special cases of α-divergence. 29

List of Figures

4.1 Equidistant datapoints A, B and C among the undisplayed datapoints. There are many undisplayed datapoints between A and B but there is no undisplayed datapoints between A and C. 33

5.1 The effect of varying λ in $\mathcal{C}_{\mathrm{NeRV}}$ (left) and α in D_α (right). In both figures, $p = 0.5$ is fixed and the values are calculated for different values of $q \in [0, 1]$ and the corresponding parameter. As can be seen, the α-divergence has smoother variation compared to $\mathcal{C}_{\mathrm{NeRV}}$. Two functions coincide exactly at the end-points where $\lambda = 0(\alpha = 0)$ and $\lambda = 1(\alpha = 1)$. 38
5.2 Visualization of the sphere dataset with α increasing from 0 to 1. 39
5.3 Gradients of (a) SNE, (b) t-SNE and (c) α-SSNE as a function of the pairwise Euclidean distances between two points in the high-dimensional space and the low-dimensional image. α-SSNE (and, α-SNE, correspondingly) also produces more balanced gradients compared to SSNE (SNE). Note the different color scales. 41

6.1 An example of the objects in the COIL-20 dataset in which the images are taken in different poses (only 36 images are shown). 47
6.2 Examples of the images drawn from the Sculpt Faces dataset. 48
6.3 Visualization of the Iris dataset: first row: SNE (left), t-SNE (middle), and HSSNE with $\omega = 1.2$ (right). Second row: α-SNE (left), αt-SNE (middle), and α-HSSNE with $\omega = 1.2$ (right). $\alpha = 0.25$ for all the results in the second row. 49

6.4 Visualization of the COIL-20 dataset: first row: SNE (left), t-SNE (middle), and HSSNE with $\omega = 0.5$ (right). Second row: α-SNE (left), αt-SNE (middle), and α-HSSNE with $\omega = 0.5$ (right). $\alpha = 0.9$ for all the results in the second row. 50
6.5 Visualization of the SculptFaces dataset: first row: SNE (left), t-SNE (right). Second row: HSSNE with $\omega = 0.5$ (left) and α-SNE with $\alpha = 0.5$ (right). 51
6.6 Pointwise quality measure for the embeddings shown in Figure 6.5: The values are shown using a hot colormap where white indicates the best performance while dark specifies the worst performance. 52
6.7 Examples of the images drawn from the UMist Faces dataset. 55
6.8 Examples of the handwritten digits drawn from the MNIST dataset. 56
6.9 Performance curves: classification accuracy (first row), AUC (second row) and negative SM objective function of EDA (third row), for datasets: UMist Faces (first column), Texture (second column) and MNIST (third column). 57
6.10 Visualization results on datasets: UMist Faces (first row), Texture (second row) and MNIST (third row), using SNE (first column), t-SNE (second column) and α-SNE (third column). . 58

List of Symbols

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	n-dimensional real vector space
$\mathbf{A} = [a_{ij}] \in \mathbb{R}^{I \times J}$	I by J real matrix
a_{ij}	ij-th element of the matrix $\mathbf{A}$
$\mathbf{v} \in \mathbb{R}^n$	n-dimensional real vector
v_i	i-th element of the vector $\mathbf{v}$
∇f	Gradient of the real valued function f
$\nabla_{\mathbf{x}} f$	Gradient of the real valued function f evaluated at point $\mathbf{x}$
$\log$	Natural logarithm (base e)
$\log_a$	Logarithm of base a
$\exp$	Exponential function (e^x)
$D(\mathbf{p}\\|\mathbf{q})$	Divergence between two non-negative vectors $\mathbf{p}$ and $\mathbf{q}$
$H(\mathbf{p})$	Entropy of the discrete measure $p_i \geq 0$
$\mathcal{C}$	Cost function
μ	Mean of the random variable
$Var(x)$	Variance of the random variable x

Chapter 1

Introduction

1.1 Motivation

Dimensionality reduction (DR) and particularly, data visualization has been a prominent research track for the past few decades as an important step in data analysis. Real world data, e.g., digital images, biomedical measurements, audio signals, etc., generally contain a large number of measurements for every single observation datapoint. In the DR literature, each observation is commonly referred as a *datapoint* and the number of measurements recorded in each datapoint is called the *dimension* of the dataset. In order to process the data further while preventing undesired effects such as curse of dimensionality, or perform a visualization that reveals the intrinsic structure of the data, dimensionality reduction techniques need to be applied. Principal Component Analysis (PCA) [46] and classical scaling [56] are among the first linear methods which have been applied tremendously in different applications. However, classical linear methods may not always be sufficient to handle complex non-linear data [57].

Recently, many non-linear dimensionality reduction methods such as Sammon mapping [49], Isomap [54], Locally Linear Embedding [47], Stochastic Neighbor Embedding (SNE) [23], Maximum Variance Unfolding [62] and Laplacian Eigenmaps [9] have been proposed to overcome the shortcomings of the simple linear methods in handling data lying on several non-linear manifolds. Although all these methods have been successfully applied to several artificial as well as real-world datasets, they all suffer from one major drawback called the crowding problem: many points get crowded in the center of the projection and the margins between different clusters become indistinct.

The crowding problem in each of the above-mentioned methods can be explained by considering the specific assumptions and criteria underlying the objective function. However, generally, the crowding problem in the distance preserving methods is due to the excessive emphasis on the preservation of the short distances in the low-dimensional image. In this thesis, we only consider the crowding problem associated with the SNE method and its variants and refer the reader to [34] for a more general discussion on the topic.

In SNE, the crowding problem can be explained as a result of strong attraction forces in the gradient which mainly dominates the repulsion forces between the map points. The method of UNI-SNE [15] tackles the crowding problem by considering small repulsion forces between all the datapoints in the embedding. However, it is less applicable to real-world datasets due to difficult optimization of the cost function. More recently, t-Distributed Stochastic Neighbor Embedding (t-SNE) [58] and its generalization, Heavy-tailed Symmetric SNE (HSSNE) [64], have successfully overcome the crowding problem by considering distributions with heavier tail than Gaussian in the low-dimensional space.

One main drawback associated with t-SNE is the excessive separation of the clusters, which, in some cases, produces over-separated clusters. This may become unpleasant graphically or even misleading for analyzing the data. HSSNE is able to control the level of separation by introducing a parameter ω, called the *tail-heaviness* parameter, for the degree of the distribution. $\omega \to 0$ and $\omega = 1$ correspond to Gaussian and Student t-distributions, respectively. Distributions with heavier tails can be obtained using larger values of ω. However, there is no systematic way to estimate the optimal degree of the heavy-tailed distribution for a particular dataset.

1.2 Scope and Purpose of the Thesis

In this thesis, we consider the crowding problem by a different approach; instead of manipulating the distributions in the low-dimensional space, we instead consider an α-divergence as the cost function. This choice of divergence for the cost function covers the cost function of SNE and other well-known methods such as Neighborhood Embedding Visualizer (NeRV) [61] as special cases. We show that with a proper selection of the parameter α, our method produces results as good as t-SNE or, in some cases, considerably superior. For estimating the optimal value of α, we present a statistical framework based on a recently proposed distribution called Exponential Divergence with Augmentation (EDA) [19]. EDA is an approximate generalization of Tweedie distribution, which has a well-established relation to β-divergence. With a nonlinear transformation, an equivalence between β and α-divergences can be shown and EDA can also be used for estimation of α. The application of different divergence measures for SNE has been studied before (for example, see [12] and [33]). However, none of the previous work investigate the properties of α-divergence as the cost function and accordingly, its gradient for updating the mapped points, nor provide any systematic approach to estimate the optimal value of the divergence parameter for a given dataset.

1.3 Structure of the Thesis

The organization of the dissertation is as follows. We first start with briefly reviewing the different dimensionality reduction methods in Chapter 2. In Chapter 3, we introduce the families of the Csiszár f-divergence and the Bregman divergence and then, consider two important classes of divergences, namely β- and α-divergences and represent their relation. These divergences are essential for the development of our new visualization algorithm. We consider the SNE, NeRV and t-SNE methods in Chapter 4. Then, in Chapter 5, we propose our new method of α-divergence for SNE. We provide the motivation for using α-divergence and explore the characteristics of its gradient. Next, we discuss the possible extensions of our method to distributions having heavier tail than Gaussian. Finally, we present our framework for estimating the optimal value of α for a given data distribution. We provide our experimental results in Chapter 6 and finally, draw the conclusions and present tracks for future work in Chapter 7.

Chapter 2

Dimensionality Reduction and Information Visualization

In this chapter, we briefly review a number of linear and non-linear dimensionality reduction methods. A comprehensive study of the nonlinear dimensionality reduction methods can be found in [34]. A comparative review of the different dimensionality reduction methods can be found in [57]. We first start with classical methods such as PCA and MDS and then, consider the more recent nonlinear methods later.

Principal Component Analysis

Principal Component Analysis (PCA) [46] is an orthogonal linear transformation which converts a set of observations (datapoints) consisting of a number of possibly correlated variables into a set of values which are linearly uncorrelated. This amounts to converting the covariance matrix of the data into a diagonal matrix. PCA is closely related to Singular Value Decomposition (SVD) [37], a method used to convert an arbitrary matrix into a product of an orthogonal matrix, a diagonal matrix and another orthogonal matrix, respectively.

PCA finds a set of orthogonal coordinates or axes called principal components such that the first principal component captures the largest variance or direction of variation in the data. The second component points to the direction of second largest variance, with additional constraint that it is orthogonal to the first component. The rest of the principal components are found in a similar manner such that each component is orthogonal to all the previous components.

PCA is often the first dimensionality reduction method to apply on an unknown dataset due to its fast but satisfactory performance. A low-dimensional representation of the data can be obtained by considering the k first principal components and projecting the data onto these components. For a visualization application, generally $k = 2$ or 3. PCA can also be used to remove the undesired effects in the data e.g., noise, by considering only the directions that capture the main variation in the data. It can be shown that the PCA is the optimal method to approximate given data in the sense of least-squares error.

Multidimensional Scaling

Multidimensional Scaling (MDS) [11] refers to a set of distance preserving visualization techniques. An MDS algorithm aims to find a map in which the pairwise distances between datapoints are preserved as much as possible. The objective function to be minimized is called the *stress function*, in this context, and the optimization is mainly performed using a procedure called *stress majorization* [29]. Only the matrix of pairwise distances is sufficient to find the map and therefore, the coordinates of the datapoints in the high-dimensional space need not be known. This property makes MDS suitable for situations such as visualization of the results of a psychological test, where only the (dis)similarity values between different objects in the dataset are available. The dimension of the map points is not restricted to $k = 2$ or 3, but it should be provided to the algorithm beforehand, by the user.

There exist several variants of MDS, e.g., Classical MDS, Metric MDS, Non-Metric MDS, etc. Classical MDS is equivalent to the PCA method. However, other variants consider different stress functions and input matrices of distances with weights or other means for calculating the pairwise distances in the high-dimensional space.

Sammon's Mapping

Sammon's Mapping [49] is a nonlinear metric MDS method which emphasized more on preserving the short distances rather than the long ones. The projection can be found by iterative methods such as gradient descent [45]. The number of iterations should be set experimentally and there is no guarantee for the convergence. However, it has been shown that the PCA works as a proper initial configuration [36].

Isomaps

Isomap [54] is a metric MDS method which considers the geodesic distances as the pairwise distances between the high-dimensional datapoints. The distances are calculated by finding the shortest-path distances (computed for example using Dijkstra's algorithm [18]) on a weighted neighborhood graph. The distance matrix is then applied as the input to the MDS algorithm. The motivation for using the geodesic distance is to consider the manifold structure of the data. Instead of using the straight Euclidean distances as in MDS, the distances are calculated by summing up the edge weights in the shortest path between a pair of datapoints.

Self-organizing Map

Self-organizing Map (SOM) or Kohonen map [28] is a type of artificial neural network which maps the input space onto a low-dimensional (typically two) discretized representation. The map consists of a fixed lattice, called the *grid*, in which the neurons are arranged with respect to a predefined neighborhood structure. For example, in a square grid, we can consider a 4 or 8 neighborhood while, in a hexagonal grid, each neuron will have 6 neighbors. Each neuron in the network represents a prototype, which is a vector that has the same dimension as the input space. Additionally, each neuron has a number of neighbors on the map (or so called, the grid). The map is trained via competitive learning; each input is fed to the network and compared to all the prototype vectors in the map. The neuron with the prototype having the lowest Euclidean distance with the input vector (the winning neuron) is updated according to the input vector. This update also affects a neighborhood of the winning neuron on the grid. However, the neighborhood region shrinks while the training proceeds.

After the training phase, the map can also be used for mapping where each new input is classified into one of the existing prototype vectors. This property can be used to find a low-dimensional representation of the input data on the grid. Therefore, SOM is one of the most commonly used methods for data visualization. SOM can also be used as a vector quantizer where each input vector is mapped into a codeword from a finite dictionary. This property is similar to that of the k-means algorithm [20] for small networks. However, it is shown that the larger networks rearrange the data in a more fundamentally topological way [28].

Curvilinear Component Analysis

Curvilinear Component Analysis (CCA) [17] has been proposed as a visualization method in which the fundamental idea is inherited from SOM. However, unlike SOM, the output is not restricted to a fixed lattice. Therefore, the map is a continuous space which can take the shape of the submanifold. The learning consists of two steps: vector quantization of the submanifold of the data and nonlinear projection of the quantized vectors onto the output space. Authors also provide training strategies as an alternative to the stochastic gradient descent used in similar methods. Additionally, CCA can be used both for continuous forward and backward mapping; that is, a new input can be mapped to a new output representation, with respect to its neighboring datapoints and vice versa.

Curvilinear Distance Analysis

The idea in Curvilinear Distance Analysis (CDA) [31, 32] is similar to that in CCA, with the only difference that the Euclidean distances in the high-dimensional space are substituted with geodesic distances, similar to Isomaps. The rest of the algorithm remains the same. CDA improves the performance of CCA in handling non-linear manifold structures such as the Swiss roll dataset [55]. The Swiss roll dataset is obtained by sampling a set of datapoints from a 3D surface which is formed by rolling a 2D rectangular surface around a fixed axis. The dataset serves as a standard benchmark for evaluating the performance of the manifold learning techniques [34].

Locally Linear Embedding

Locally Linear Embedding (LLE) [47] makes the assumption that the underlying manifold in the data is smooth enough (and also there exist enough samples) such that it can be presented locally by a linear approximation. Thus, each datapoint in the high-dimensional space is represented by a convex sum of its k-nearest (or alternatively, ϵ-ball) neighbors. Then, the same set of weights are used to find the representation of the points in the low-dimensional space. It is shown in [9] that LLE is approximately equivalent to calculating the eigenfunctions of the iterated graph Laplacian $\mathcal{L}^2$.

Laplacian Eigenmaps

Laplacian Eigenmaps [9] is closely related to spectral method for data clustering. Spectral methods refer to a set of techniques which are based on the evaluation of the eigenvalues and eigenvectors of a properly formed matrix [44, 51, 52]. The idea in the Laplacian eigenmaps is to first find the adjacency graph (also known as the neighborhood graph[1]) for the whole dataset by using a heat kernel or any other kernel and then, calculate the map using the first k eigenvectors of the graph Laplacian having the smallest

[1] The simplest example of a neighborhood graph is a graph having a node for each datapoint and edges (with unit weights) between every neighboring pair of datapoints.

eigenvalues (while ignoring the eigenvector corresponding to eigenvalue 0). The authors show the relation of the graph Laplacian to the Laplace Beltrami operator [48] on manifolds and also draw the connection between the method and spectral clustering and the LLE method. The method performs well for datasets having natural intrinsic clusters, however, it provides moderate results for the other cases where methods such as PCA or Isomaps may be more preferable.

Maximum Variance Unfolding

Maximum Variance Unfolding (MVU) [62] is a manifold learning method based on semidefinite programming [59]. The idea is to preserve the local isometry by imposing a number of constraints on the neighborhood graph, which is formed in a similar manner as the previous methods. These constraints are translated into a number of constraints on the Gram matrix of the high- and low-dimensional datapoints. The objective to maximize with respect to the previous constraints is the variance or the sum of pairwise Euclidean distances between the map points. Thus, the final objective reduces to a semidefinite trace maximization problem with respect to the isometry constraints. The final embedding is obtained from the dominant eigenvectors of the Gram matrix, learned by semidefinite programming. MVU is particularly suitable for unfolding the low-dimensional manifolds embedded in a space with a higher dimension. It is also able to correctly estimate the underlying dimensionality of the data sets.

Kernel Principal Component Analysis

Kernel Principal Component Analysis or Kernel PCA [50] is an extension of the PCA algorithm to a reproducing kernel Hilbert space [7] by first mapping the datapoint to the new space by using a kernel and then, performing the eigendecomposition of the resulting covariance matrix. This can be used in cases where the data is not linearly separable but, it can be separated by hyperplanes in a space having a higher (possibly infinite) dimension. The datapoints are not actually evaluated in the new space and all the computations are performed implicitly using the so called kernel trick. Therefore, the method can be used only to project the datapoints to a new space but the corresponding principal components are not obtained explicitly.

Kernel Information Embedding

Kernel Information Embedding [38] is an information theoretic approach for dimensionality reduction and regression. In this method, a latent random variable lying on a low-dimensional space is obtained by maximizing the mutual information between the random variable in the input space and latent random variable. Using the unsupervised kernel density estimation methods for the both random variables, the mutual information is expanded to find the gradient with respect to the latent variable. This method can also be used to perform regression between a given input and an output space. Again, an intermediate latent variable is defined such that the conditional mutual information between the input and the output random variables, given the value of the latent variable, is minimized.

Stochastic Neighbor Embedding

Stochastic Neighbor Embedding (SNE) [23] is a dimensionality reduction method which aims to preserve the local neighborhood structure of each datapoint when mapping from a high-dimensional space into a low-dimensional space. The neighborhood probabilities for each datapoint are defined as the probability of a random walker jumping to any of the neighboring datapoints when starting from the given datapoint. The probability mass around each datapoint is assumed to be normally distributed, with a proper value of the variance. Therefore, the probabilities are calculated by normalizing the exponents in the Gaussian distribution. The consistency between the distributions in the high-dimensional and the low-dimensional spaces is computed by considering sum of Kullback-Leibler divergences between the distributions in the data space and the map over all the datapoints. The map is found by initializing the map points randomly close to the origin and then, using gradient descent for optimization. SNE performs well in many cases, however, it suffers from the crowding problem, discussed earlier.

Aspect Maps

The basic SNE algorithm can be extended in such a way that each datapoint can occur in several different maps with a different mixing proportion such that the sum of the proportions equals to one. These maps are called aspect maps [15]. In the aspect maps, the conditional neighborhood probabilities are substituted with symmetric joint probabilities over the whole datapoints. In addition to the aspect maps, a single map can be obtained by allocating part of the density to a background distribution. This causes a small repulsion force between the map points which avoids the crowding of the points in the center of the map. The method which is assumed to be a degenerate version of the aspect maps is called UNI-SNE. The UNI-SNE is not used much in practice due to the difficult optimization of its cost function.

t-Distributed Stochastic Neighbor Embedding

t-Distributed Neighbor Embedding (t-SNE) [58] is an extension of the SNE algorithm with two major differences: first, a joint distribution is assumed over the whole datapoints in both the high-dimensional and the low-dimensional space, similar to the aspect maps. Second, a Student t-distribution with single degree of freedom is used in the low-dimensional space to calculate the similarities. Student t-distribution is obtained by adding up an infinite number of Gaussian distributions with the same mean but different variances. The result is a distribution which has a heavier tail than the Gaussian. This property enables the t-distribution to be more *robust* to the outliers. The outliers occur if data is created by a process which has an underlying heavy-tailed distribution or, a few datapoints are incorrectly labeled [10].

t-SNE cost function has better convergence properties than the SNE and UNI-SNE, and usually converges to a proper solution without the need for considering jitter noise[2] to escape the local minima. t-SNE also effectively avoids the crowding problem by considering larger distances between the datapoints in the map. However, in some cases, the results may seem over separated and may not faithfully reflect the true distribution.

Heavy-tailed Symmetric Stochastic Neighbor Embedding

The tail-heaviness of the distribution can be controlled by parameterizing the heavy-tailed distribution in the t-SNE. The method, called Heavy-tailed Symmetric Stochastic Neighbor Embedding (HSSNE) [64], covers both Gaussian and t-distribution as well as distributions residing between the two cases and those having heavier tails. The authors provide a method based on Lagrange multipliers to calculate the gradient for a general form of distribution. Additionally, a fixed point multiplicative algorithm with local mixture interpretation is provided to calculate the map as an alternative to the gradient descent method adopted in SNE and t-SNE.

Multi-view Stochastic Neighbor Embedding

Multi-view Stochastic Neighbor Embedding (m-SNE) [63] combines the information from multiple views of the same dataset to obtain a single map.

[2] Jitter noise refers to the perturbation which is added in the optimization process to escape from poor local minima and converge to a better solution. This can be a random Gaussian noise with a small magnitude. The process mimics the simulated annealing technique [27] which is commonly used in the optimization in order to find a better optimum.

The different views may be obtained from different feature extractors, applied to the same set of datapoints. For instance, for an image dataset, these can be the shape, texture and color features. Neighborhood probabilities from different views are combined to calculate the similarities using a set of combination coefficients. The map points are obtained in a similar manner as in SNE. The combination coefficients are then obtained by solving a convex optimization problem. Nesterov's accelerated first-order method [25, 42, 43] is used as a fast solver for the convex problem. The process is repeated m times where m denotes the number of views.

Neighbor Retrieval Visualizer

Neighbor Retrieval Visualizer (NeRV) [61] considers the dimensionality reduction as an information retrieval task where the objective becomes to establish a balance between precision and recall. The authors also provide new interpretation for Kullback-Leibler and inverse Kullback-Leibler divergences between the probabilities (similarities) in the high-dimensional and low-dimensional spaces as generalization of the recall and precision, respectively. Therefore, the cost function becomes a convex combination of the sum of Kullback-Leibler divergences (called mean smoothed recall) and sum of inverse Kullback-Leibler divergences (called mean smoothed precision) over all datapoints. NeRV achieves different maps by adjusting the parameter in the summation based on the desired aspects of the visualization.

Chapter 3

Information Divergence

Information divergence originates from the estimation theory where a divergence maps two probability distributions to a non-negative dissimilarity or distance. In other words, it measures the difference between a known distribution $\mathbf{p}$ and its approximation $\mathbf{q}$. The objective is then defined to minimize the divergence between the observed data and the approximation. Information divergences are widely used in different applications such as non-negative matrix/tensor factorization [14], Bayesian network optimization [39], coding theory [16], and Stochastic Neighbor Embedding [23].

Among different families of divergence measures, those of separable types are of huge interest. The family of distance-type separable measures satisfy the condition

$$D(\mathbf{p}\|\mathbf{q}) = \sum_{i=1}^{n} d(p_i, q_i) \geq 0\,, \tag{3.1}$$

where $\mathbf{p}$ and $\mathbf{q}$ are two n-dimensional probability distributions and the equality is achieved if and only if $\mathbf{p} = \mathbf{q}$. However, distance-type measures do not necessarily satisfy the properties of a metric on the space $\mathcal{P}$ of all probability distributions. In other words, they are not necessarily symmetric,

$$D(\mathbf{p}\|\mathbf{q}) = D(\mathbf{q}\|\mathbf{p})\,, \tag{3.2}$$

and do not necessarily satisfy the triangular inequality,

$$D(\mathbf{p}\|\mathbf{q}) \leq D(\mathbf{p}\|\mathbf{z}) + D(\mathbf{z}\|\mathbf{q})\,. \tag{3.3}$$

In this chapter, we briefly introduce two important families of distance-type measures, namely the Csiszár f-divergence and the Bregman divergence. These two families include several well-known divergences as special cases. We then proceed by introducing the β-divergence and its properties and then, consider the α-divergence and also its relation to the β-divergence.

3.1 Csiszár Divergences

The Csiszár f-divergence between to distributions $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ is defined as

$$D_f(\mathbf{p}\|\mathbf{q}) = \sum_i q_i f\left(\frac{p_i}{q_i}\right) , \tag{3.4}$$

where f is a real-valued convex function over the open interval $(0, +\infty)$ and satisfies $f(1) = 0$. By convention, $0f(0/0)$ and $0f(p/0) = \lim_{q\to 0^+} qf(\frac{p}{q}) = pf'(\infty)$. It can be easily verified that $D_f(\mathbf{p}\|\mathbf{q}) \geq 0$ with the equality if and only if $\mathbf{p} = \mathbf{q}$. The ratio p_i/q_i is called the "likelihood ratio".

The Csiszár f-divergence is related to a generalized entropy of the form

$$H_f(\mathbf{p}) = -\sum_i f(p_i) \,. \tag{3.5}$$

The Shannon entropy can be obtained as a special case where $f(p) = p \log p$.

Csiszár f-divergence has the property that for a positive constant c, we have

$$D_{cf}(\mathbf{p}\|\mathbf{q}) = cD_f(\mathbf{p}\|\mathbf{q}) \,. \tag{3.6}$$

So, we can normalize f to have $f''(1) = 1$. This is referred as the problem of scale. Additionally, the function

$$\tilde{f}(u) = f(u) - c(u - 1) \tag{3.7}$$

produces the same divergence as f, that is,

$$D_f(\mathbf{p}|\mathbf{q}) = D_{\tilde{f}}(\mathbf{p}\|\mathbf{q}) \,. \tag{3.8}$$

We set $c = f'(1)$ to have $\tilde{f}'(1) = 0$ and

$$\tilde{f}(u) \geq 0 \,, \tag{3.9}$$

with equality if and only if $u = 1$. The class of convex functions with properties $f(1) = 0$, $f'(1) = 0$ and $f''(1) = 1$ is denoted by $\mathcal{F}$. For class of differentiable functions, there is no loss of generality to use $f \in \mathcal{F}$. Furthermore, if the function f is bounded, we have $\tilde{f}(0) = \lim_{u\to 0^+} \tilde{f}(u)$.

The Csiszár f-divergence is defined originally for probability distributions. However, it can be extended to positive measures $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$, for which the constraints $\sum_i \tilde{p}_i = 1$ and $\sum_i \tilde{q}_i = 1$ are discarded. For a convex function $f \in \mathcal{F}$, the divergence takes the same form

$$D_f(\tilde{\mathbf{p}}\|\tilde{\mathbf{q}}) = \sum_i \tilde{p}_i f\left(\frac{\tilde{q}_i}{\tilde{p}_i}\right) . \tag{3.10}$$

Name	Function $f(u), \quad u = \frac{p}{q}$	Divergence $D_f(\mathbf{p}\|\mathbf{q})$
Kullback-Leibler divergence	$u \log u$	$\sum_i p_i \log(p_i/q_i)$
Total variation distance	$\|u-1\|$	$\sum_i \|p_i - q_i\|$
Pearson Chi-square distance	$(u-1)^2$	$\sum_i \frac{(p_i-q_i)^2}{q_i}$
Neyman Chi square	$\frac{(u-1)^2}{u}$	$\sum_i \frac{(p_i-q_i)^2}{p_i}$
Rukhin distance	$\frac{(u-1)^2}{a+(1-a)u}$	$\sum_i \frac{(p_i-q_i)^2}{(1-a)p_i+aq_i}, a \in [0,1]$
Triangular Discrimination (TD)	$\frac{(u-1)^2}{u+1}$	$\sum_i \frac{(p_i-q_i)^2}{p_i+q_i}$
Squared Hellinger distance	$(\sqrt{u}-1)^2$	$\sum_i(\sqrt{p_i}-\sqrt{q_i})^2$
Matsusita distance	$\|u^\alpha - 1\|^{\frac{1}{\alpha}}, 0 \leq \alpha \leq 1$	$\sum_i \|p_i^\alpha - q_i^\alpha\|^{\frac{1}{\alpha}}$
Piuri and Vinche divergence	$\frac{\|1-u\|^\gamma}{(u+1)^{\gamma-1}}, \gamma \geq 1$	$\sum_i \frac{\|p_i-q_i\|^\gamma}{(p_i+q_i)^{\gamma-1}}$
Arimoto distance	$\sqrt{1+u^2} - \frac{1+u}{\sqrt{2}}$	$\sum_i \left(\sqrt{p_i^2+q_i^2} - \frac{p_i+q_i}{\sqrt{2}}\right)$

Table 3.1: Basic divergences expressed as the Csiszár f-divergence.

However, for a general f with $f'(1) = c_f \neq 0$, we need to use

$$D_f(\tilde{\mathbf{p}}\|\tilde{\mathbf{q}}) = c_f \sum_i (\tilde{q}_i - \tilde{p}_i) + \sum_i \tilde{p}_i f\left(\frac{\tilde{q}_i}{\tilde{p}_i}\right). \tag{3.11}$$

When $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$ are probability distributions, (3.11) reduces to (3.4).

The class of Csiszár f-divergences contains many well-known divergences between two probability distributions. The list of divergence measures which can be derived from Csiszár f-divergence along with the corresponding function f is presented in Table 3.1. As an important case, the α-divergence, which we will consider in more detail in Section 3.4, can be expressed formally as the Csiszár f-divergence with $f(u) = u(u^{\alpha-1}-1)/(\alpha^2-\alpha) + (1-u)/\alpha$.

In general, Csiszár f-divergence does not need to be symmetric; that is, $D_f(\mathbf{p}\|\mathbf{q})$ does not necessarily equal $D_f(\mathbf{q}\|\mathbf{p})$. However, we can define the conjugate generated function $f^*(u) = uf(1/u)$ such that

$$D_f(\mathbf{p}\|\mathbf{q}) = D_{f^*}(\mathbf{q}\|\mathbf{p}). \tag{3.12}$$

Name	Function $f(u), \quad u = \frac{p}{q}$	Divergence $D_f(\mathbf{p}\|\mathbf{q})$
Squared Hellinger distance	$\frac{1}{2}(\sqrt{u} - 1)$	$\frac{1}{2}\sum_i(\sqrt{p_i} - \sqrt{q_i})^2$
Triangular Discrimination	$\frac{(u-1)^2}{u+1}$	$\sum_i \frac{(p_i-q_i)^2}{p_i+q_i}$
Symmetric Chi-squared	$\frac{(u-1)^2(u+1)}{u}$	$\sum_i \frac{(p_i-q_i)^2(p_i+q_i)}{p_i q_i}$
J-divergence	$(u-1)\log u$	$\sum_i (p_i - q_i)\log\frac{p_i}{q_i}$
Jensen-Shannon divergence	$\frac{u}{2}\log u + \frac{u+1}{2}\log\frac{2}{u+1}$	$\sum_i p_i \log\left(\frac{2p_i}{p_i+q_i}\right) + q_i \log\left(\frac{2q_i}{p_i+q_i}\right)$
A-G Mean divergence	$\frac{u+1}{2}\log\left(\frac{u+1}{2\sqrt{u}}\right)$	$\sum_i \left(\frac{p_i+q_i}{2}\right)\log\left(\frac{p_i+q_i}{2\sqrt{p_i q_i}}\right)$

Table 3.2: Basic symmetric divergences expressed as the Csiszár f-divergence.

Thus, for an arbitrary Csiszár f-divergence, we can form the convex function $f_s(u) = f(u) + f^*(u)$ to obtain the symmetric divergence $D_{f_s}(\mathbf{p}\|\mathbf{q})$. The list of symmetric divergences which can be expressed as the Csiszár f-divergence is shown in Table 3.2.

3.2 Bregman Divergence

In this section, we consider another important family of divergence measures called the Bregman divergence. It is widely used in non-negative matrix factorization [14], clustering [8], and data visualization [53]. It includes a number of well-known divergences such as the β-divergence, which will be considered in more details in the next section.

The generalized ϕ-entropy of discrete measure $p_i \geq 0$ with respect to a strictly convex real-valued function $\phi(p)$ is defined as

$$H_\phi(\mathbf{p}) = -\sum_i \phi(p_i)\,, \tag{3.13}$$

and the Bregman divergence is given by

$$D_\phi(\mathbf{p}\|\mathbf{q}) = \sum_i (\phi(p_i) - \phi(q_i) - \phi'(q_i)(p_i - q_i))\,, \tag{3.14}$$

where $\phi'(q_i)$ denotes the derivative with respect to q_i.

In many applications, we are interested in separable divergences $D_\phi(\mathbf{p}\|\mathbf{q}) = \sum_i D_\phi(p_i\|q_i)$. However, a more general vectorized form of the Bregman divergence can be defined as follows.

Let $\phi(\mathbf{p}) : \mathbb{R}^n \to \mathbb{R}$ be strictly convex and first order differentiable. The corresponding Bregman divergence is defined as

$$D_\phi(\mathbf{p}\|\mathbf{q}) = \phi(\mathbf{p}) - \phi(\mathbf{q}) - (\mathbf{p}-\mathbf{q})^T \nabla\phi(\mathbf{q})\,, \tag{3.15}$$

where $\nabla\phi(\mathbf{q})$ represents the gradient of ϕ evaluated at $\mathbf{q}$. In a similar manner, $D_\phi(\mathbf{q}\|\mathbf{p})$ is defined as

$$D_\phi(\mathbf{q}\|\mathbf{p}) = \phi(\mathbf{q}) - \phi(\mathbf{p}) + (\mathbf{p}-\mathbf{q})^T \nabla\phi(\mathbf{p})\,. \tag{3.16}$$

The Bregman divergence is not generally symmetric. However, we can formulate the dual representation using the Legendre transformation as follows. Let $\phi(\mathbf{p})$ be a convex function. The one to one correspondence

$$\mathbf{p}^* = \nabla\phi(\mathbf{p}) \tag{3.17}$$

is regarded as the dual representation of $\mathbf{p}$. Thus, we can define the convex function

$$\phi^*(\mathbf{p}^*) = \max_{\mathbf{p}}\{\mathbf{p}^T\mathbf{p}^* - \phi(\mathbf{p})\}\,. \tag{3.18}$$

The following relation holds between the two convex functions

$$\phi(\mathbf{p}) + \phi(\mathbf{p}^*) - \mathbf{p}^T\mathbf{p}^* = 0\,, \tag{3.19}$$

when $\mathbf{p}$ and $\mathbf{p}^*$ correspond to each other. The reverse transformation has a similar form

$$\mathbf{p} = \bigtriangledown\phi^*(\mathbf{p}^*)\,. \tag{3.20}$$

Thus, the correspondence is dually coupled.

Using the dual representation, the Bregman divergence can be written in a symmetric form

$$D_\phi(\mathbf{p}\|\mathbf{q}) = \phi(\mathbf{p}) + \phi^*(\mathbf{q}^*) - \mathbf{p}^T\mathbf{q}^*\,, \tag{3.21}$$

which is the same as (3.14) and non-negative with value zero if and only if $\mathbf{p} = \mathbf{q}$.

The Bregman divergence has a nice geometrical interpretation as a measure of convexity of ϕ. When a tangent line is drawn to the convex function ϕ at point $\mathbf{q}$, the Bregman divergence can be considered as the vertical distance between ϕ and the line, evaluated at point $\mathbf{p}$.

The Bregman divergence also includes a number of well-known divergence measures as special cases. The list of divergence measures which can be derived from the Bregman divergence along with the corresponding convex function ϕ is represented in Table 3.3.

At the end, we list a number of properties of the Bregman divergence. For a more detailed discussion, please refer to [14].

1. $D_\phi(\mathbf{p}\|\mathbf{q})$ is convex in the first argument $\mathbf{p}$, but is not convex (in general) in the second argument $\mathbf{q}$. However, it is convex in the dual representation $\mathbf{q}^*$.

2. If $\phi(\mathbf{p})$ is strictly convex, we have $D_\phi(\mathbf{p}\|\mathbf{q}) \geq 0$ with equality if and only if $\mathbf{p} = \mathbf{q}$.

3. The Bregman divergence is not usually symmetric. Moreover, the triangular inequality does not hold in general, i.e., it is not a metric,
$$D_\phi(\mathbf{p}\|\mathbf{q}) \neq D_\phi(\mathbf{q}\|\mathbf{p})\,. \tag{3.22}$$

4. The gradient has the form
$$\nabla_{\mathbf{p}} D_\phi(\mathbf{p}\|\mathbf{q}) = \nabla\phi(\mathbf{p}) - \nabla\phi(\mathbf{q})\,. \tag{3.23}$$

5. Linearity: for $a > 0$, we have
$$D_{\phi_1 + a\phi_2}(\mathbf{p}\|\mathbf{q}) = D_{\phi_1}(\mathbf{p}\|\mathbf{q}) + aD_{\phi_2}(\mathbf{p}\|\mathbf{q})\,. \tag{3.24}$$

6. Invariance up to a linear term:
$$D_{\phi + a\mathbf{p} + c}(\mathbf{p}\|\mathbf{q}) = D_\phi(\mathbf{p}\|\mathbf{q})\,. \tag{3.25}$$

7. The three-points property which generalizes the law of cosines
$$D_\phi(\mathbf{p}\|\mathbf{q}) = D_\phi(\mathbf{p}\|\mathbf{z}) + D_\phi(\mathbf{z}\|\mathbf{q}) - (\mathbf{p} - \mathbf{z})^T(\nabla\phi(\mathbf{p}) - \nabla\phi(\mathbf{z}))\,. \tag{3.26}$$

8. Generalized Pythagoras Theorem:
$$D_\phi(\mathbf{p}\|\mathbf{q}) \geq D_\phi(\mathbf{p}\|P_\Omega(\mathbf{q})) + D_\phi(P_\Omega(\mathbf{q})\|\mathbf{q})\,, \tag{3.27}$$
where $P_\Omega(\mathbf{q}) = \arg\min_{\omega\in\Omega} D_\phi(\omega\|\mathbf{q})$ is the Bregman projection onto the convex set Ω. The equality holds when Ω is an affine set.

Name	Function $\phi(\mathbf{p})$	Divergence $D(\mathbf{p}\|\mathbf{q})$
Squared Euclidean distance	$\frac{1}{2}\|\mathbf{p}\|_2^2 = \sum_i p_i^2$	$\frac{1}{2}\|\mathbf{p}-\mathbf{q}\|_2^2 = \frac{1}{2}(p_i - q_i)^2$
Mahalanobis distance	$\frac{1}{2}\mathbf{p}^T\mathbf{W}\mathbf{p}, \mathbf{W}$-symmetric p.d.	$\frac{1}{2}(\mathbf{p}-\mathbf{q})^T\mathbf{W}(\mathbf{p}-\mathbf{q})$
Generalized KL divergence	$\sum_i(p_i \log p_i)$	$\sum_i \left(p_i \log \frac{p_i}{q_i} - p_i + q_i\right)$
Itakura-Saito distance	$-\sum_i \log p_i$	$\sum_i \left(\frac{p_i}{q_i} - \log\frac{p_i}{q_i} - 1\right)$
Inverse	$\sum_i \frac{1}{p_i}$	$\sum_i \left(\frac{p_i}{q_i^2} + \frac{1}{p_i} - \frac{2}{q_i}\right)$
Exponential	$\sum_i \exp(p_i)$	$\sum_i(e^{p_i} - (p_i - q_i + 1)e^{q_i})$

Table 3.3: Basic divergences expressed as the Bregman divergence.

3.3 β-Divergence

The discrete β-divergence is defined as

$$D_\beta(\mathbf{p}\|\mathbf{q}) = \frac{\sum_i p_i^{\beta+1} + \beta q_i^{\beta+1} - (\beta+1)p_i q_i^\beta}{\beta(\beta+1)}, \tag{3.28}$$

where $\beta \in \mathbb{R}$ ($\beta \neq 0$ and $\beta \neq -1$). It is easy to check that $D_\beta(\mathbf{p}\|\mathbf{q})$ is a valid divergence having non-negative values and is equal to zero if and only if $\mathbf{p} = \mathbf{q}$. That can also be deduced from the fact that the β-divergence can be derived from the Bregman divergence [22] with

$$\phi(p) = \begin{cases} \frac{p^{\beta+1}}{\beta(\beta+1)} - \frac{p}{\beta} + \frac{1}{\beta+1}, & \beta \neq 0, -1 \\ -\log p + p - 1, & \beta = -1 \\ p \log p - p + 1, & \beta = 0 \end{cases}. \tag{3.29}$$

The singular cases $\beta = 0$ and $\beta = -1$ are defined in the limit $\beta \to 0$ and $\beta \to -1$, respectively. For $\beta \to 0$, we obtain the generalized Kullback-Leibler

Formula	Name
$D_{\beta=1}(\mathbf{p}\|\mathbf{q}) = D_{\mathrm{EU}}(\mathbf{p}\|\mathbf{q}) = \frac{1}{2}\sum_i (p_i - q_i)^2$	Euclidean
$D_{\beta\to 0}(\mathbf{p}\|\mathbf{q}) = D_{KL}(\mathbf{p}\|\mathbf{q}) = \sum_i \left(p_i \log\frac{p_i}{q_i} - p_i + q_i\right)$	Generalized KL
$D_{\beta\to -1}(\mathbf{p}\|\mathbf{q}) = D_{\mathrm{IS}}(\mathbf{p}\|\mathbf{q}) = \sum_i \left(\log\frac{q_i}{p_i} + \frac{p_i}{q_i} - 1\right)$	Itakura-Saito
$D_{\beta=-2}(\mathbf{p}\|\mathbf{q}) = \sum_i \left(\frac{p_i}{2q_i^2} - \frac{1}{q_i} + \frac{1}{2p_i}\right)$	

Table 3.4: Special cases of β-divergence.

divergence[1]

$$D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) = \lim_{\beta\to 0} D_\beta(\mathbf{p}\|\mathbf{q}) = \sum_i \left(p_i \log\frac{p_i}{q_i} - p_i + q_i\right), \tag{3.30}$$

whereas for $\beta \to -1$, the Itakura-Saito distance is obtained as

$$D_{\mathrm{IS}}(\mathbf{p}\|\mathbf{q}) = \lim_{\beta\to -1} D_\beta(\mathbf{p}\|\mathbf{q}) = \sum_i \left(\log\frac{q_i}{p_i} + \frac{p_i}{q_i} - 1\right). \tag{3.31}$$

The special cases of β-divergence are listed in the Table 3.4.

3.4 α-Divergence

The α-divergence is a specific form of Csiszár f -divergence. It can also be derived from the Bregman divergence [6]. The (asymmetric) α-divergence over discrete distributions is defined by

$$D_\alpha(\mathbf{p}\|\mathbf{q}) = \frac{\sum_i p_i^\alpha q_i^{1-\alpha} - \alpha p_i + (\alpha-1)q_i}{\alpha(\alpha-1)}. \tag{3.32}$$

The α-divergence satisfies the basic property of an error measure by attaining value of zero for $\mathbf{p} = \mathbf{q}$, and having a positive value otherwise. This property is a corollary of the fact that the α-divergence is a convex function with respect to p_i and q_i [14]. It contains many well-known divergences as

[1] Note that we use the notation $D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q})$ for the generalized KL divergence; if $\mathbf{p}$ and $\mathbf{q}$ are probability distributions such that $\sum_i p_i = \sum_i q_i = 1$, this coincides with the standard KL divergence.

Formula	Name
$D_{\alpha=2}(\mathbf{p}\|\mathbf{q}) = D_{\mathrm{P}}(\mathbf{p}\|\mathbf{q}) = \frac{1}{2}\sum_i \frac{(p_i-q_i)^2}{q_i}$	Pearson Chi-square
$D_{\alpha\to1}(\mathbf{p}\|\mathbf{q}) = D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) = \sum_i \left(p_i \log\frac{p_i}{q_i} - p_i + q_i\right)$	Generalized KL
$D_{\alpha=1/2}(\mathbf{p}\|\mathbf{q}) = 2D_{\mathrm{H}}(\mathbf{p}\|\mathbf{q}) = 2\sum_i(\sqrt{p_i}-\sqrt{q_i})^2$	Hellinger
$D_{\alpha\to0}(\mathbf{p}\|\mathbf{q}) = D_{\text{I-KL}}(\mathbf{p}\|\mathbf{q}) = \sum_i \left(q_i \log\frac{q_i}{p_i} - q_i + p_i\right)$	Inverse KL
$D_{\alpha=-1}(\mathbf{p}\|\mathbf{q}) = D_{\mathrm{IP}}(\mathbf{p}\|\mathbf{q}) = \frac{1}{2}\sum_i \frac{(p_i-q_i)^2}{p_i}$	Inverse Pearson

Table 3.5: Special cases of α-divergence.

special cases including inverse Pearson (Neyman Chi-square), Hellinger and Pearson Chi-square distances for $\alpha = -1, 0.5$ and 2, respectively. Additionally, the singular points $\alpha = 0$ and $\alpha = 1$ are calculated in the limit $\alpha \to 0$ and $\alpha \to 1$, respectively, where we have

$$D_{\mathrm{KL}}(\mathbf{q}\|\mathbf{p}) = \lim_{\alpha\to0} D_\alpha(\mathbf{p}\|\mathbf{q}) = \sum_i \left(q_i \log\frac{q_i}{p_i} - q_i + p_i\right), \tag{3.33}$$

and

$$D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) = \lim_{\alpha\to1} D_\alpha(\mathbf{p}\|\mathbf{q}) = \sum_i \left(p_i \log\frac{p_i}{q_i} - p_i + q_i\right). \tag{3.34}$$

The special cases of the α-divergence are listed in Table 3.5

α-divergence and β-divergence are related to each other using a non-linear transformation between α and β. By letting $r_i = p_i^\alpha/\alpha^{2\alpha}$ and $s_i = q_i^\alpha/\alpha^{2\alpha}$ and $\beta = 1/\alpha - 1$ for $\alpha \neq 0$, we have

$$\begin{aligned} D_\beta(r_i\|s_i) &= \frac{1}{\beta(\beta+1)}\left(r_i^{\beta+1} + \beta s_i^{\beta+1} - (\beta+1)r_i s_i^\beta\right) \\ &= \frac{-\alpha^2}{\alpha-1}\left(\frac{p_i}{\alpha^2} + \frac{1-\alpha}{\alpha}\frac{q_i}{\alpha^2} - \frac{1}{\alpha}\frac{p_i^\alpha}{\alpha^{2\alpha}}\frac{q_i^{1-\alpha}}{\alpha^2(1-\alpha)}\right) \\ &= D_\alpha(p_i\|q_i)\,. \end{aligned} \tag{3.35}$$

This will be useful for applying the optimization framework for β to find the optimal value of α in Chapter 5.

Chapter 4

Stochastic Neighbor Embedding

The goal of neighbor embedding is to find a projection of the data in which the neighborhood structure or, in other words, the similarities between pairs of points in the high-dimensional space and its low-dimensional counterpart are similar by means of a divergence measure. We first start by defining the probabilistic models of the neighborhood which are non-negative similarity measures between pairs of points. Then, we review the cost functions of the previous approaches for neighbor embedding. In Chapter 5, we provide the intuition behind selecting α-divergence as the cost function and the range of α we are mainly interested in. We also represent the optimization framework for estimating the best value of α.

4.1 Probabilistic Model of Neighborhood

4.1.1 Conditional Similarities

Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^D$ be the set of high-dimensional datapoints, and let $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N \in \mathbb{R}^d$ be the corresponding set of low-dimensional images[1]. In most applications, such as visualization, we have $d \ll D$. For each datapoint i, the probabilistic neighborhood in the data space, p_{ij}, is defined as the conditional probability of choosing $\mathbf{x}_j$ as a neighbor when $\mathbf{x}_i$ is the starting point, under the assumption of a Gaussian distribution centered at $\mathbf{x}_i$ [23]:

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ik}^2)} \,. \tag{4.1}$$

[1] We use the words *map*, *image* and *embedding* interchangeably, to refer to the low-dimensional representation of the datapoints, denoted by $\mathcal{Y}$.

$d_{ij}(\cdot)$ can be any proper distance measure on the input space. A common choice is the Euclidean distance scaled with a parameter σ_i,

$$d_{ij}^2 = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma_i^2} . \tag{4.2}$$

The value of σ_i is chosen based on the density of the data at $\mathbf{x}_i$; that is, σ_i takes a comparatively smaller value when the distribution of the datapoints in the neighborhood of $\mathbf{x}_i$ is dense or, takes a larger value otherwise. Every value of σ_i induces a probability distribution $\mathbf{p}_i$ over the rest of the datapoints. The Shannon entropy of this distribution, defined as

$$H(\mathbf{p}_i) = -\sum_j \log_2(p_{ij}) , \tag{4.3}$$

is proportional to the value of σ_i, i.e., it increases as the value of σ_i increases. So, by fixing the entropy of the distribution or, equivalently, the *perplexity*, defined as

$$Perp(\mathbf{p}_i) = 2^{H(\mathbf{p}_i)} , \tag{4.4}$$

we can calculate the value of σ_i using a binary search. The σ_i is chosen such that the entropy of the distribution is equal to $\log_2 k$ where k is the number of effective neighbors or the perplexity. The perplexity is set by the user (typically, $k = 30$). This choice of σ_i adjusts the scale with respect to the density of the data.

The neighborhood probability in the low-dimensional space, q_{ij}, is formed in a similar manner:

$$q_{ij} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k \neq i} \exp(-\|\mathbf{y}_i - \mathbf{y}_k\|^2)} , \tag{4.5}$$

where in this case, a constant variance of $\sigma^2 = \frac{1}{2}$ is considered for all image points.

4.1.2 Symmetric Similarities

Symmetric SNE (SSNE), as the name implies, considers symmetric similarities $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$, $\forall i, j$, by replacing the conditional probabilities by a single joint probability distribution. In the low-dimensional image, the joint probability Q over all image points is defined by

$$q_{ij} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k \neq l} \exp(-\|\mathbf{y}_k - \mathbf{y}_l\|^2)} . \tag{4.6}$$

Furthermore, for the data space, the joint distribution P is obtained by first finding the the conditional probabilities p_{ij} as in 4.1, then, setting $p_{ij} = \frac{p_{ij}+p_{ji}}{2n}$. This prevents the outliers from becoming erratically placed without conforming with the rest of the points [58]. As we will see, SSNE yields to a simpler gradient term compared to SNE and produces results at least as good as or, in some cases, better than SNE.

4.1.3 Student t-Distribution and Distributions with Heavier Tails

In order to have larger distances in the low-dimensional map, the Gaussian distribution can be substituted with a distribution having a heavier tail. A typical choice is to use a Student t-distribution, as in [58]. t-distribution amounts to an infinite mixture of Gaussians with different sigmas. Using this distribution, the similarities q_{ij} in the low-dimentional space are defined as

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}} \,. \tag{4.7}$$

The choice of a t-distribution for finding the similarities allows the map points, which are moderately distant in the original space, to be mapped far away from each other and hence, prevents crushing of the points in the center of the map. It also approaches an inverse square law for map points having a large pairwise distance $\|\mathbf{y}_i - \mathbf{y}_j\|$. This means that the distant clusters act as individual points in the map and therefore, result in a projection conducted on different scales; that is, clusters of datapoints located in a far distance from each other are treated as single datapoints.

The choice of a t-distribution for mapped points can be extended to distributions having heavier tail. This can be done by considering a parameter α for the degree of the distribution [64]. Using this convention, the joint probabilities q_{ij} become

$$q_{ij} = \frac{(1 + \omega\|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1/\omega}}{\sum_{k \neq l}(1 + \omega\|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1/\omega}} \,. \tag{4.8}$$

$\omega \to 0$ and $\omega = 1$ amount to Gaussian and t-distribution, respectively. Distribution having heavier tails can be obtained by considering larger values of ω.

4.1.4 Random Walk Based Similarities

In some cases where the number of datapoints is too large, calculation of the map using all the datapoints becomes infeasible. In those cases, the

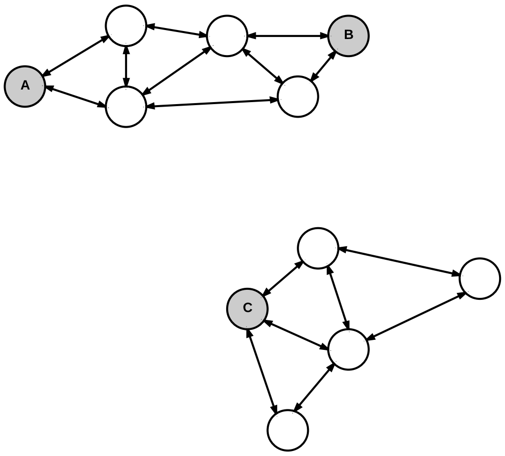


Figure 4.1: Equidistant datapoints A, B and C among the undisplayed datapoints. There are many undisplayed datapoints between A and B but there is no undisplayed datapoints between A and C.

map can be calculated using a smaller sub-sample of the dataset. However, simply considering only the selected datapoints neglects a significant amount of information about the underlying manifolds, provided by the rest of the datapoints. As an example, consider the arrangement of the equidistant datapoints A, B and C among the other undisplayed datapoints in Figure 4.1. Since there are more undisplayed datapoints lying between A and B compared to the pair A and C, the datapoints A and B are much more likely to be in the same cluster than A and C. Therefore, considering the effect of undisplayed datapoints in calculation of the similarities p_{ij} becomes crucial.

One way to consider this problem is the one similar to diffusion maps, that is, forming the neighborhood graph for all of the datapoints and calculating the topological distance such as shortest path on the graph between pair of landmark points to be displayed. A more convenient way to calculate the similarities is to consider the probability of a random walker starting from a landmark point to hit another landmark point. This can be done by first calculating the neighborhood graph for all the datapoints, using the similarities proportional to $\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2)$. This part is performed only once for the whole dataset. Then, the p_{ij} probabilities for the landmark point $\mathbf{x}_i$ can be found computationally by repeatedly starting the random walker from $\mathbf{x}_i$ and counting the number of times that it hits the landmark point $\mathbf{x}_j$. The result is normalized by calculating the ratio over the all landmark points.

An alternative analytical solution can also be obtained by solving the combinatorial Dirichlet problem to find a harmonic function subject to its boundary values on the neighborhood graph [21]. However, it is shown in [58] that there is no significant difference between the performance of computational approach and the analytical solution.

4.2 Neighbor Embedding

Let p_{ij}, where $\sum_j p_{ij} = 1$, be the pairwise similarities for the set of points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N} \in \mathbb{R}^D$, formed by assuming a Gaussian distribution as in (4.1). In the basic SNE algorithm, the consistency between the distributions in the high-dimensional space and the low-dimensional image is achieved by minimizing the sum of Kullback-Leibler (KL) divergences over pairs of distributions for all datapoints,

$$C_{\mathrm{SNE}} = \sum_i D_{\mathrm{KL}}(\mathbf{p}_i \| \mathbf{q}_i) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}\,, \tag{4.9}$$

where the probabilities q_{ij} are calculated using (4.5). KL divergence is the natural choice for a divergence between two probability distributions from the same probability space. It amounts to the cross-entropy between two distributions $\mathbf{p}_i$ and $\mathbf{q}_i$ (up to a constant entropy term) and measures the average information loss when $\mathbf{q}_i$ is used to represent the true distribution, $\mathbf{p}_i$ [16].

Minimization of the cost function (4.9) can be performed by finding the gradient and using a standard gradient descent method. The gradient with respect to $\mathbf{y}_i$ takes a rather simple form

$$\frac{\partial C_{\mathrm{SNE}}}{\partial \mathbf{y}_i} = 2 \sum_j (\mathbf{y}_i - \mathbf{y}_j)\,(p_{ij} - q_{ij} + p_{ji} - q_{ji})\ . \tag{4.10}$$

The gradient (4.10) has the physical interpretation of sum of forces exerted from springs between $\mathbf{y}_i$ and each of the other points. The force between each pair of points is proportional to their distance as well as the stiffness of the spring, which is simply the mismatch $(p_{ij} - q_{ij} + p_{ji} - q_{ji})$ between the conditional probabilities in the data space and the image.

For SSNE, the summation in (4.9) is replaced by a single sum over the joint probability distributions in both high-dimensional and low-dimensional spaces. The gradient of SSNE takes an even simpler form

$$\frac{\partial C_{\mathrm{SSNE}}}{\partial \mathbf{y}_i} = 4 \sum_j (\mathbf{y}_i - \mathbf{y}_j)\,(p_{ij} - q_{ij})\ . \tag{4.11}$$

NeRV method [61] provides a generalization of the SNE cost function by introducing a new interpretation for KL and inverse-KL divergences. It is shown in [61] that $D_{\mathrm{KL}}(\mathbf{q}_i \| \mathbf{p}_i)$ and $D_{\mathrm{KL}}(\mathbf{p}_i \| \mathbf{q}_i)$ can be viewed as generalizations of *precision* and *recall*, respectively, for point i in the mapping $\mathcal{X} \to \mathcal{Y}$. Bearing this in mind, the cost function can be chosen to make a trade off between maximizing precision or recall, by considering a convex sum of KL and inverse-KL divergences over all points:

$$C_{\mathrm{NeRV}} = \lambda \sum_i D_{\mathrm{KL}}(\mathbf{p}_i \| \mathbf{q}_i) + (1 - \lambda) \sum_i D_{\mathrm{KL}}(\mathbf{q}_i \| \mathbf{p}_i) \,, \tag{4.12}$$

parameterized by $\lambda \in [0, 1]$. In case of $\lambda = 1$, the cost function reduces to the cost function of SNE. This corresponds to maximizing recall in the embedding. On the other hand, setting $\lambda = 0$ corresponds to maximizing precision. A choice of $\lambda \in (0, 1)$ encourages a balance between these two extreme cases. However, the selection of λ is done manually by the user.

In a parametric method, the selection of a proper parameter is an important step since it can highly affect the result of the visualization. The parameter may be set by an expert having some intuition about the task or possible prior knowledge about the type of the data. However, in many tasks, visualization is mainly the first step to gain knowledge about the structure of the data. An alternative approach to overcome this problem is to perform several visualizations using different values of the parameter and then, select the one which produces the best result by means of subjective assessment or other objective quality measures. Again, objective measures may sound more reliable in this case. Nevertheless, quality measures may not always refer to the faithfulness of the visualization in describing the data. Therefore, a more desirable approach is to select the parameter which faithfully represents the data, as much as possible.

Maximum likelihood setting provides an standard framework for the selection of the parameters in a learning problem [10]. However, this approach is impractical for almost all of the well-known stochastic neighbor embedding methods since either there exists no known compact form for the distribution of the error or the error function itself is not even a natural divergence measure. On the other hand, unintuitively changing the divergence may not always be desirable since not all the well-known divergence measures, corresponding to assumption of known underlying distributions for the error, are appropriate for a visualization task [12].

The above considerations impose a need for adopting an alternative error function while maintaining two important goals: first, the error function should cover our range of interest, namely the convex sum of inverse KL and KL divergences, by providing a parameterization to attain the desired properties in the resulting embedding. Second, the distribution of the error should consent to a maximum likelihood framework to perform automatic selection of the parameter. In the next chapter, we introduce our approach for considering these two objectives.

Chapter 5

Stochastic Neighbor Embedding with α-Divergence

5.1 α-SNE Method

We considered the properties of the α-divergence in Chapter 3. We now focus our attention to the interval $\alpha \in [0, 1]$: when $\alpha = 1$, α-divergence corresponds exactly to the SNE cost function for a single datapoint. Furthermore, the points $\alpha = 0$ and $\alpha = 1$ amount to the cost function of NeRV when $\lambda = 0$ and $\lambda = 1$, respectively. More generally, when α varies from 0 to 1, α-divergence passes smoothly through all values of NeRV cost function for $\lambda \in (0, 1)$ since the divergence itself is a continuous function of α. However, the mapping from λ to α is not onto; this can be easily seen by considering two arbitrary distributions and varying λ and α and, finding the value of each. Thus, α-divergence covers even a wider range compared to NeRV cost function. However, there exists no closed form solution to write α divergence as a convex some of KL and inverse-KL divergences. Figure 5.1 illustrates the effect of varying λ and α in $\mathcal{C}_{\text{NeRV}}$ and D_α, respectively. The results are obtained by fixing $p = 0.5$ and calculating the values as a function of $q \in [0, 1]$ and the corresponding parameter. Two functions coincide exactly when $\lambda = 0$ ($\alpha = 0$) and $\lambda = 1$ ($\alpha = 1$).

The aforementioned properties promote investigating the sum of α-diver-

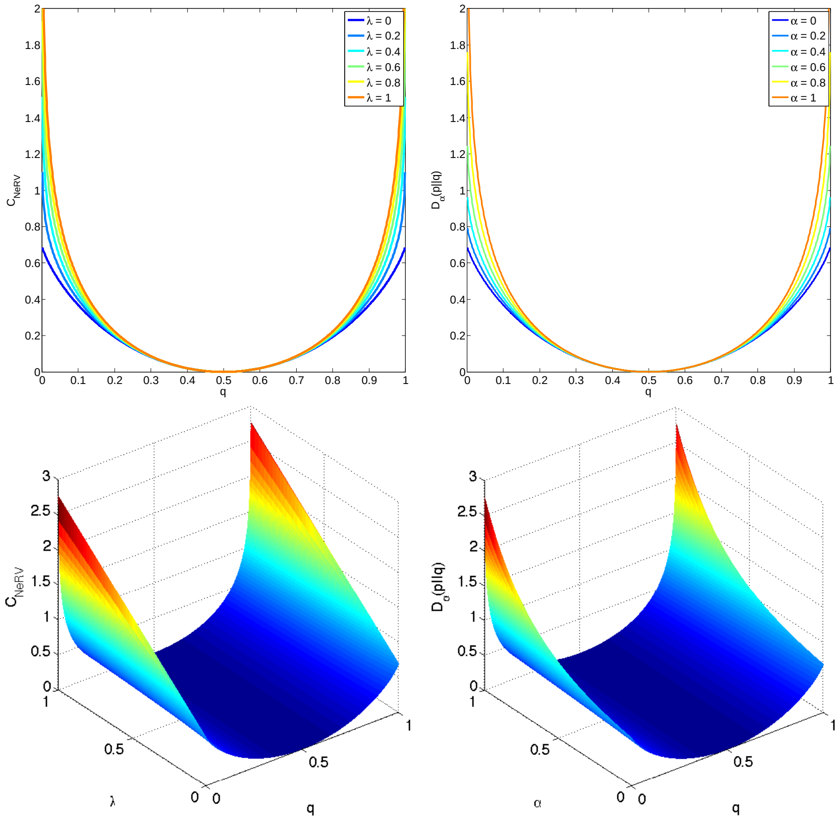


Figure 5.1: The effect of varying λ in $\mathcal{C}_{\text{NeRV}}$ (left) and α in D_α (right). In both figures, $p = 0.5$ is fixed and the values are calculated for different values of $q \in [0, 1]$ and the corresponding parameter. As can be seen, the α-divergence has smoother variation compared to $\mathcal{C}_{\text{NeRV}}$. Two functions coincide exactly at the end-points where $\lambda = 0(\alpha = 0)$ and $\lambda = 1(\alpha = 1)$.

gences over all pairs of distributions as the cost function:

$$C_{\alpha\text{-SNE}} = \sum D_\alpha(\mathbf{p}_i \| \mathbf{q}_i) = \begin{cases} \sum_{ij} \dfrac{p_{ij}^{\alpha} q_{ij}^{1-\alpha} - \alpha p_{ij} + (\alpha - 1) q_{ij}}{\alpha(\alpha - 1)}, & \alpha \neq 0, 1 \\ \sum_{ij} q_{ij} \log(q_{ij}/p_{ij}) = \sum D_{\text{KL}}(\mathbf{q}_i \| \mathbf{p}_i), & \alpha = 0 \ . \\ \sum_{ij} p_{ij} \log(p_{ij}/q_{ij}) = \sum D_{\text{KL}}(\mathbf{p}_i \| \mathbf{q}_i), & \alpha = 1 \end{cases} \tag{5.1}$$

We call the new method stochastic neighbor embedding with α-divergence (α-SNE).

Figure 5.2 illustrates the effect of varying α from 0 to 1 on the sphere dataset. The dataset includes 1000 uniform samples from a unit sphere in

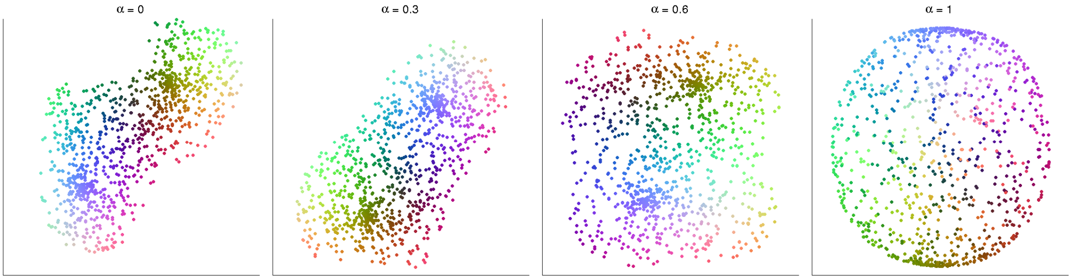


Figure 5.2: Visualization of the sphere dataset with α increasing from 0 to 1.

$\mathbb{R}^3$. The map starts with unfolding the sphere for $\alpha = 0$, then continues by smoothly expanding the map from the corners until it reaches the round shape when $\alpha = 1$. It is worth mentioning that the value of α is not restricted to the interval $[0, 1]$ and can take any value in $\mathbb{R}$.

More interesting properties are revealed by considering the gradient. The calculation of the gradient is presented in Appendix A. Direct calculation of the gradient is tedious, however, it can also be found easily using the generalized framework in [12] based on Fréchet derivatives. The gradient has the form

$$\frac{\partial C_{\alpha\text{-SNE}}}{\partial \mathbf{y}_i} = \frac{2}{\alpha} \sum_{j \neq i} (\mathbf{y}_i - \mathbf{y}_j) \left(p_{ij}^{\alpha} q_{ij}^{1-\alpha} - \theta_i q_{ij} + p_{ji}^{\alpha} q_{ji}^{1-\alpha} - \theta_j q_{ji} \right), \quad \alpha \neq 0, \tag{5.2}$$

in which, $0 \leq \theta_i = \sum_{j \neq i} p_{ij}^{\alpha} q_{ij}^{1-\alpha}$ is called *compatibility factor* for point i, in this paper, with the following properties: $\theta_i = 1$ if $\mathbf{p}_i = \mathbf{q}_i$ and $\theta_i \leq 1$ for $\alpha \in (0, 1]$ and $\theta_i \geq 1$ elsewhere (except $\alpha = 0$[1]). The gradient has similar interpretation of springs between points with stiffness proportional to the mismatch in the probability distributions. However, comparing the gradient with the gradient of SNE (4.10), it can be seen that the attraction terms p_{ij} and p_{ji} are replaced by $p_{ij}^{\alpha} q_{ij}^{1-\alpha}$ and $p_{ji}^{\alpha} q_{ji}^{1-\alpha}$, respectively. On the other hand, the repulsion terms q_{ij} and q_{ji} are weighted by the compatibility factors for points i and j, respectively. Therefore, compatibility factor for point i can also be seen as sum of the attraction terms between i and rest of the points. Finally, the whole gradient is scaled by a factor of $1/\alpha$.

The properties above result in two major effects on the gradient for $\alpha \in (0, 1)$: first, the attraction and repulsion forces become more balanced by means of absolute strength. Consequently, α-SNE produces much smoother and stabler gradient compared to SNE. Second, the repulsion forces for mod-

[1] The case $\alpha = 0$ is treated separately (see Appendix A).

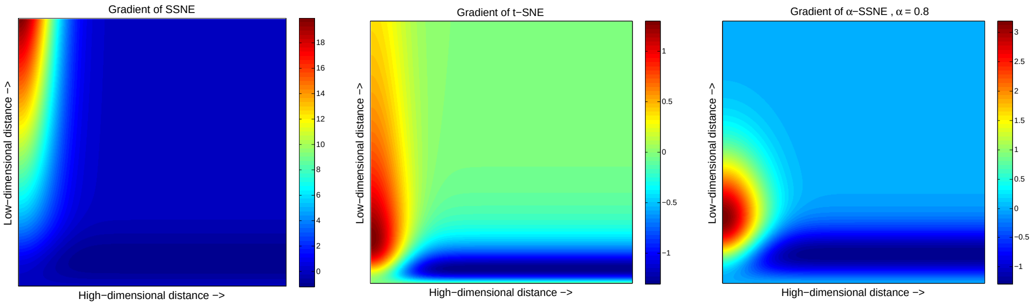
erately distant points which are mapped close together become relatively larger. This property resembles the behavior of the gradient of t-SNE method for a similar setting. However, in t-SNE, this property is governed by the extra terms $(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}$ amending the gradient, as an effect of using t-distribution in the image space. Alternatively, α-SNE, with a proper selection of α, also efficiently eliminates the crowding problem associated with SNE by introducing regularizing terms in the gradient. The generalization of α-SNE to symmetric version is quite straightforward. Symmetric α-SNE (α-SSNE) has properties similar to α-SNE. In this case, the gradient is

$$\frac{\partial C_{\alpha\text{-SSNE}}}{\partial \mathbf{y}_i} = \frac{4}{\alpha} \sum_{j \neq i} (\mathbf{y}_i - \mathbf{y}_j) \left(p_{ij}^{\alpha} q_{ij}^{1-\alpha} - \theta q_{ij} \right), \quad \alpha \neq 0 \,, \tag{5.3}$$

with $\theta = \sum \sum_j p_{ij}^{\alpha} q_{ij}^{1-\alpha}$.

Figure 2(a) to 2(c) show the gradients of SSNE, t-SNE and α-SSNE (with $\alpha = 0.8$), respectively, for a pair of points in a two-dimensional image, as a function of their Euclidean distances in the high-dimensional space and the low-dimensional space (i.e., as a function of $\|\mathbf{x}_i - \mathbf{x}_j\|$ and $\|\mathbf{y}_i - \mathbf{y}_j\|$). Positive values represent attraction while negative values correspond to repulsion. As it can be seen in Figure 2(a), SSNE exerts large attraction force for moderately close points which are mapped far from each other. However, the repulsion force is comparatively small for the opposite case (around 19 to 1). t-SNE (Figure 2(b)) results in a more balanced gradient, compared to SSNE, by damping the large attraction forces and further, strongly repelling the dissimilar datapoint which are mapped close together. Nevertheless, α-SSNE also achieves a balanced gradient, as in t-SNE. More interestingly, the attraction and repulsion forces cover a wider range, compared to t-SNE.

The optimization of the cost function can be achieved using standard methods e.g. steepest descent. In our early implementation, we used gradient descent method for optimization. A jitter noise with a constant variance can be used to model simulated annealing in early stages. A more effective trick to escape local minima is the one similar to "early compression" in [58], that is, for moderately small values of α, we start from a larger value, e.g., $\alpha = 1$, and gradually reduce α until we reach the true value. We proceed by performing a few number of iterations using the true value of α, until the final embedding is achieved. Thus, in the early stages of optimization, points tend to stay fairly close together and start to form initial clusters. While gradually decreasing α, these clusters get more separated and converge to the final embedding. This trick prevents clusters to get separated in the early iterations, because, if a cluster is torn into smaller pieces, located far away from each other, there will not be enough attraction forces to bind them



(a) Gradient of SSNE (b) Gradient of t-SNE (c) Gradient of α-SSNE

Figure 5.3: Gradients of (a) SNE, (b) t-SNE and (c) α-SSNE as a function of the pairwise Euclidean distances between two points in the high-dimensional space and the low-dimensional image. α-SSNE (and, α-SNE, correspondingly) also produces more balanced gradients compared to SSNE (SNE). Note the different color scales.

together in the later stages.

5.2 Extension to Heavy-tailed Distributions

Extensions of α-SNE to t-distribution and distributions with heavier tail can be achieved using the Lagrange method [64]. We start by defining unnormalized similarities $\bar{q}_{ij} = H(\|\mathbf{y}_i - \mathbf{y}_j\|)$, such that

$$q_{ij} = \frac{\bar{q}_{ij}}{\sum_{k \neq l} \bar{q}_{kl}} \,. \tag{5.4}$$

$H(\tau)$ can be any monotonically decreasing function of $\tau > 0$ and does not need to be a valid probability distribution since the normalization is performed in (5.4). For SSNE, $H(\|\mathbf{y}_i - \mathbf{y}_j\|) = \exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)$ and (5.4) reduces to (4.6). For t-SNE, $H(\|\mathbf{y}_i - \mathbf{y}_j\|) = (1 + \|\mathbf{y}_i - \mathbf{y}_j\|)^{-1}$ and we have (4.7).

We can now exchange the minimization of (5.1) with the following optimization problem

$$\begin{aligned} \underset{\bar{q},\mathcal{Y}}{\text{maximize}} \quad \mathcal{L}(\bar{q}, \mathcal{Y}) &= \frac{1}{\alpha(1-\alpha)} \sum_{ij} p_{ij}^{\alpha} \left(\frac{\bar{q}_{ij}}{\sum_{k \neq l} \bar{q}_{kl}} \right)^{1-\alpha} , && (5.5) \\ \text{subject to} \quad \bar{q}_{ij} &= H(\|\mathbf{y}_i - \mathbf{y}_j\|^2) \,. && (5.6) \end{aligned}$$

The extended objective function using Lagrange multipliers becomes

$$\bar{\mathcal{L}}(\bar{q}, \mathcal{Y}) = \frac{1}{\alpha(1-\alpha)} \sum_{ij} p_{ij}^{\alpha} \left(\frac{\bar{q}_{ij}}{\sum_{k \neq l} \bar{q}_{kl}} \right)^{1-\alpha} + \sum_{ij} \lambda_{ij} (\bar{q}_{ij} - H(\|\mathbf{y}_i - \mathbf{y}_j\|)) \,. \tag{5.7}$$

Setting the derivative $\frac{\partial \bar{\mathcal{L}}}{\bar{q}_{ij}}$ to zero yields $\lambda_{ij} = \frac{1}{\alpha}(\frac{p_{ij}^{\alpha} q_{ij}^{1-\alpha}}{\bar{q}_{ij}} - \frac{\theta}{\sum_{k \neq l} \bar{q}_{kl}})$ where $\theta = \sum_{ij} p_{ij}^{\alpha} q_{ij}^{1-\alpha}$. Taking the derivative with respect to $\mathbf{y}_i$ and substituting for λ_{ij}, we have

$$\frac{\partial C_{\alpha\text{-HSSNE}}}{\partial \mathbf{y}_i} = -\frac{\partial \bar{\mathcal{L}}}{\partial \mathbf{y}_i} = \frac{4}{\alpha} \left(p_{ij}^{\alpha} q_{ij}^{1-\alpha} - \theta q_{ij} \right) S(\|\mathbf{y}_i - \mathbf{y}_j\|^2)(\mathbf{y}_i - \mathbf{y}_j) \,, \tag{5.8}$$

where

$$S(\tau) = -\frac{d \log H(\tau)}{d\tau} \tag{5.9}$$

is the negative score function or tail-heaviness function. It maps any similarity function to a tail-heaviness function. For Gaussian similarity, $H(\tau) = exp(-\tau)$ and we have $S(\tau) = 1$. For Student t-distribution, $H(\tau) = (1+\tau)^{-1}$ and therefore, $S(\tau) = H(\tau)$.

The tail-heaviness function can further be parameterized by a power of H, that is, $S(\tau) = H(\tau)^{\omega}$ where $\omega \geq 0$. This choice of parameterization allows achieving distributions with heavier tail than t-distribution for $\omega > 1$. $H(\tau)$ itself can be obtained by solving a first order differential equation

$$-\frac{d \log H(\tau)}{d\tau} = [H(\tau)]^{\omega} \,, \tag{5.10}$$

which yields $H(\tau) = (\omega\tau + c)^{-1/\omega}$. The constant c is set to 1 in order to have consistency with SNE and t-SNE. We call the new method Heavy tailed Symmetric α-SNE (α-HSSNE).

Using (5.8), the gradient of t-distributed α-SNE (called αt-SNE, henceforth) takes the form

$$\frac{\partial C_{\alpha\text{t-SNE}}}{\partial \mathbf{y}_i} = \frac{4}{\alpha} \sum_{j \neq i} (\mathbf{y}_i - \mathbf{y}_j)(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1} \left(p_{ij}^{\alpha} q_{ij}^{1-\alpha} - \theta q_{ij} \right) \,. \tag{5.11}$$

Compared to the gradient of original t-SNE algorithm

$$\frac{\partial C_{\text{t-SNE}}}{\partial \mathbf{y}_i} = 4 \sum_{j \neq i} (\mathbf{y}_i - \mathbf{y}_j)(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1} \left(p_{ij} - q_{ij} \right) \,, \tag{5.12}$$

again, the attraction and repulsion terms are smoothed by $(q_{ij}/p_{ij})^{1-\alpha}$ and θ, respectively. Additionally, the whole gradient is scaled by a factor of $1/\alpha$.

The case $\alpha = 0$ can be obtained in the limit $\alpha \to 0$, where we have

$$\frac{\partial C_{\alpha\text{t-SNE}}}{\partial \mathbf{y}_i} = 4 \sum_{j \neq i} (\mathbf{y}_i - \mathbf{y}_j) \left(q_{ij} D_{\text{KL}}(Q \| P) - q_{ij} \log(q_{ij}/p_{ij}) \right) , \alpha = 0 \,. \quad (5.13)$$

in which, $D_{\text{KL}}(Q\|P) = \sum_{j \neq i} q_{ij} \log(q_{ij}/p_{ij})$ is calculated over the joint distributions P and Q.

5.3 α-Optimization

After defining the cost function of α-SNE and obtaining a method to appropriately optimize the cost function, there remains the problem of selecting the optimal value of α for a particular dataset. The optimality condition should be consistent with other quality measures of dimensionality reduction, as we will see in the next chapter.

5.3.1 Tweedie Distribution

The probability density function (pdf) of an exponential dispersion model (EDM) is defined as

$$p_{\text{EDM}}(x; \theta, \phi, p) = f(x, \phi, p) \exp\left(\frac{1}{\phi} \theta x - \kappa(\theta) \right) , \quad (5.14)$$

where θ is the canonical parameter, $\phi > 0$ is the dispersion parameter and κ is the cumulant function. For EDM, the mean μ and the variance $Var(x)$ are related to the first and second derivatives of κ, as follows:

$$\begin{aligned} \mu &= \kappa'(\theta) \,, & (5.15) \\ Var(x) &= \phi \kappa''(\theta) \,. & (5.16) \end{aligned}$$

A Tweedie distribution is a special case of EDMs where the variance is equal to the p'th power of the mean, that is,

$$\frac{1}{\phi} Var(x) = \mu^p \,, \quad (5.17)$$

where $p \in \mathbb{R} \backslash (0, 1)$. Setting the variance as in (5.17), the canonical parameter becomes

$$\theta = \begin{cases} \frac{\mu^{1-p}}{1-p}, & \text{if } p \neq 1 \\ \log \mu, & \text{if } p = 1 \end{cases} , \quad (5.18)$$

and for the cumulant function, we have

$$\kappa(\theta) = \begin{cases} \frac{\mu^{2-p}}{2-p}, & \text{if } p \neq 1, 2 \\ -1 - \log(-\theta), & \text{if } p = 2 \\ \exp(\theta), & \text{if } p = 1 \end{cases} . \tag{5.19}$$

Thus, substituting (5.18) and (5.19) into (5.14) and setting $\beta = 1 - p$, we have the pdf of the Tweedie distribution as follows

$$p_{\mathrm{Tw}}(x; \mu, \phi, \beta) = f(x, \phi, \beta) \exp\left[-\frac{1}{\phi}\left(\frac{x\mu^{\beta}}{\beta} - \frac{\mu^{\beta+1}}{\beta+1}\right)\right], \beta \neq -1, 0 \,. \tag{5.20}$$

The pdf for the cases $\beta = -1$ and $\beta = 0$ are obtained in a similar manner.

It is shown that Tweedie distribution is closely related to the β divergence in such a way that μ^* that maximizes the likelihood of p_{Tw} also minimizes D_β. While β-divergence does not provide a means to optimize β directly, our basic idea is that the likelihood of β stemming from p_{Tw} can be maximized for that purpose.

5.3.2 Exponential Divergence with Augmentation

The optimization of the α parameter is performed using Exponential Divergence with Augmentation (EDA) [19], a distribution proposed initially for maximum likelihood estimation of β in β-divergence $D_\beta(\mathbf{x}||\boldsymbol{\mu})$, where $\mathbf{x}$ and $\boldsymbol{\mu}$ are any positive vectors such as probability distributions. Typically $\mathbf{x}$ is known and $\boldsymbol{\mu}$ is a parametric approximation. Once the optimal β is found, the optimal α is obtained by a simple transformation. EDA is an approximation to the Tweedie distribution, $p_{\mathrm{Tw}}(\mathbf{x}; \boldsymbol{\mu}, \beta, \phi)$, which is related to β-divergence. Note that both α and β divergences are separable; so, we can operate component-wise and perform our analysis on a single component. Therefore, we will drop the vector notation, henceforth.

There are some shortcomings associated with Tweedie likelihood, especially, the pdf does not exist for $\beta \in (0, 1)$ and approximation of $f(x, \phi, \beta)$ is not well studied for $\beta > 1$. The EDA density is proposed to overcome these issues, while being a close approximation to Tweedie distribution. Using the relation with β-divergence and Laplace's method, its pdf is found to be of the form [19]

$$\begin{aligned} p_{\mathrm{EDA}}(x; \mu, \phi, \beta) = \frac{1}{Z_{\beta,\phi}} \exp\Bigg[&\frac{\beta - 1}{2} \log x + \\ &\frac{1}{\phi}\left(-\frac{x^{\beta+1}}{\beta(\beta+1)} + \frac{x\mu^{\beta}}{\beta} - \frac{\mu^{\beta+1}}{\beta+1}\right)\Bigg] , \end{aligned} \tag{5.21}$$

where $Z_{\beta,\phi}$ is a normalizing constant and $D_\beta(x||\mu)$ appears in the exponent. Evaluation of $Z_{\beta,\phi}$ requires an integration in one dimension. Although it is not available analytically in general[2], it can be evaluated numerically using standard statistical software. The parameters β and ϕ can be optimized either by maximizing the likelihood or using methods for parameter estimation in non-normalized densities, such as Score Matching (SM) [24]. Both of these methods have been successfully used to find optimal β values [19].

It is now possible to use EDA to optimize α, too, using the relation between α and β-divergences. Note that both divergences are separable and we can formulate the relation using just scalars. We have

$$D_\beta(x||\mu) = D_\alpha(u||m)\,, \tag{5.22}$$

with a nonlinear transformation $x = u^\alpha/\alpha^{2\alpha}$, $\mu = m^\alpha/\alpha^{2\alpha}$ and $\beta = 1/\alpha - 1$ for $\alpha \neq 0$. This relationship allows us to evaluate the likelihood of m and α using u and β:

$$p(u; m, \alpha, \phi) = p_{\mathrm{EDA}}(x; \mu, \phi, \beta)u^{-\beta}|\beta + 1|\,. \tag{5.23}$$

α can be optimized (alongside ϕ) by maximizing its likelihood given by $p(v_i; m, \alpha, \phi)$ or minimizing the SM objective function evaluated from above. It is more convenient to treat m as constant, fixed to the value which minimizes the α-divergence. It is also possible to optimize it using EDA.

To solve our original problem, we fix vector $\mathbf{u} = [u_1, u_2, \ldots, u_{n^2}]^T$ to the vectorized form of matrix $\mathbf{P} \in \mathbb{R}_+^{n\times n}$, which contains probabilities $\mathbf{p}_i$ in α-SNE in each column, or is the joint distribution over all datapoints in α-SSNE. We also set vector $\mathbf{m} = [m_1, m_2, \ldots, m_{n^2}]^T$ equal to the vectorized form of matrix $\mathbf{Q} \in \mathbb{R}_+^{n\times n}$ which is formed in a similar manner for the map points. We then compute the values x_i and μ_i, $i = 1, 2, \ldots, n^2$, from the above transformation and optimize jointly over (α, ϕ) by minimizing the score matching objective function of the unnormalized EDA density (5.23) and select the best α value.

[2] In fact, $Z_{\beta,\phi}$ is analytically available for $\beta = 1, 0, -1, -2$, which correspond to Gaussian, Poisson, Gamma and Inverse Gaussian distributions, respectively, which are also special cases of Tweedie distribution.

Chapter 6

Experimental Results

In this chapter, we illustrate the performance of our proposed Stochastic Neighbor Embedding with α-divergence method on several different datasets. We conduct two sets of experiments: firstly, we provide some visualization results obtained from α-SNE, αt-SNE and α-HSSNE using different tail-heaviness parameters. We also demonstrate the results obtained from SNE, t-SNE and HSSNE, and compare them visually with those of our methods. Secondly, we evaluate the performance of our α-optimization framework and compare the optimal value of α obtained from EDA with the one maximizing the quality measures of visualization. We also demonstrate the results visually and compare the results with those obtained from SNE and t-SNE.

6.1 Visualization Results

In this section, we illustrate some visualization results obtained from α-SNE, αt-SNE and α-HSSNE using different tail-heaviness parameters and compare them visually with those obtained from SNE, t-SNE and HSSNE methods.

6.1.1 Datasets

As the first dataset, we use Fisher's Iris dataset [4] which contains 50 samples from three different species of Iris (150 samples in total). Each sample consists of four different measurements.

We use COIL-20 image dataset [1, 41] as our second dataset. The dataset contains gray-scale images of 20 different objects. 72 images per object (1420 in total) are taken on a turntable at 5° pose intervals against a dark background. Each image is sized $128 \times 128 = 16,384$. An example of the objects in the COIL-20 dataset is shown in Figure 6.1.

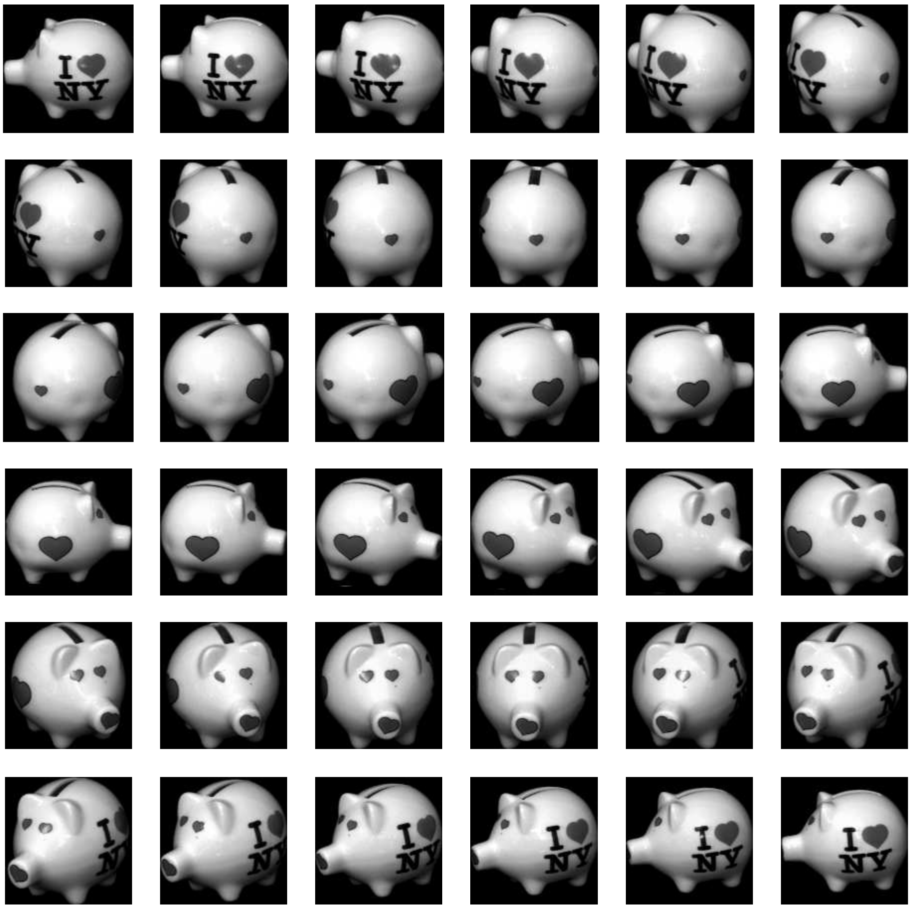


Figure 6.1: An example of the objects in the COIL-20 dataset in which the images are taken in different poses (only 36 images are shown).

Finally, we consider Sculpt Faces dataset [2] of synthetic face images. The dataset contains 698 synthetic images (each sized $64 \times 64 = 4,096$) of a faces with different poses and under different lighting conditions. The pose and lighting are such that the data form a manifold in the image space. Examples of the faces drawn from the Sculpt Faces dataset are shown in Figure 6.2.

6.1.2 Results

Figure 6.3 illustrate the visualization results of the Iris dataset. Comparing the result of SNE with α-SNE, it can be seen that α-SNE with a smaller α value ($\alpha = 0.25$) yields clusters with larger margins. On the other hand, using t-SNE results in clusters which are comparatively more distant than the SNE case. The same argument holds for α-tSNE, with additional property

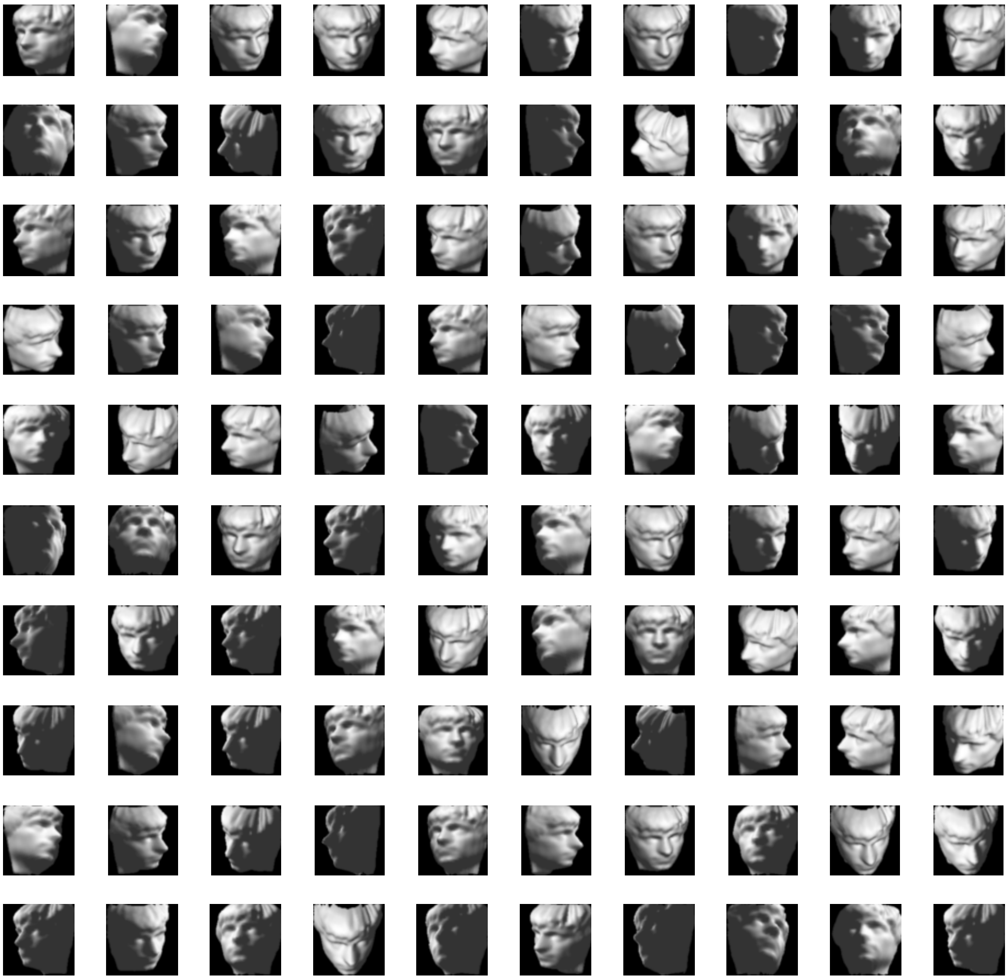


Figure 6.2: Examples of the images drawn from the Sculpt Faces dataset.

that the clusters become more concentrated. Therefore, omitting the scale (which is not shown in the figure), with αt-SNE the clusters again form larger margins than t-SNE. The results of adopting heavy tailed distributions (with $\omega = 1.2$) are similar to the former case where the previous effects become more elaborate for both methods.

The visualization results of the COIL-20 dataset is shown in Figure 6.4. Using SNE, curves corresponding to different object viewpoints are formed in the map. These curves appear to be closed since the imaging is performed from all sides of the objects, covering 360° in total. The overlap in some curves is due to the similarity between two opposing viewpoints of an object. However, many curves are crushed in the center of the map due to the crowding problem. On the other hand, by using α-SNE with a moderately

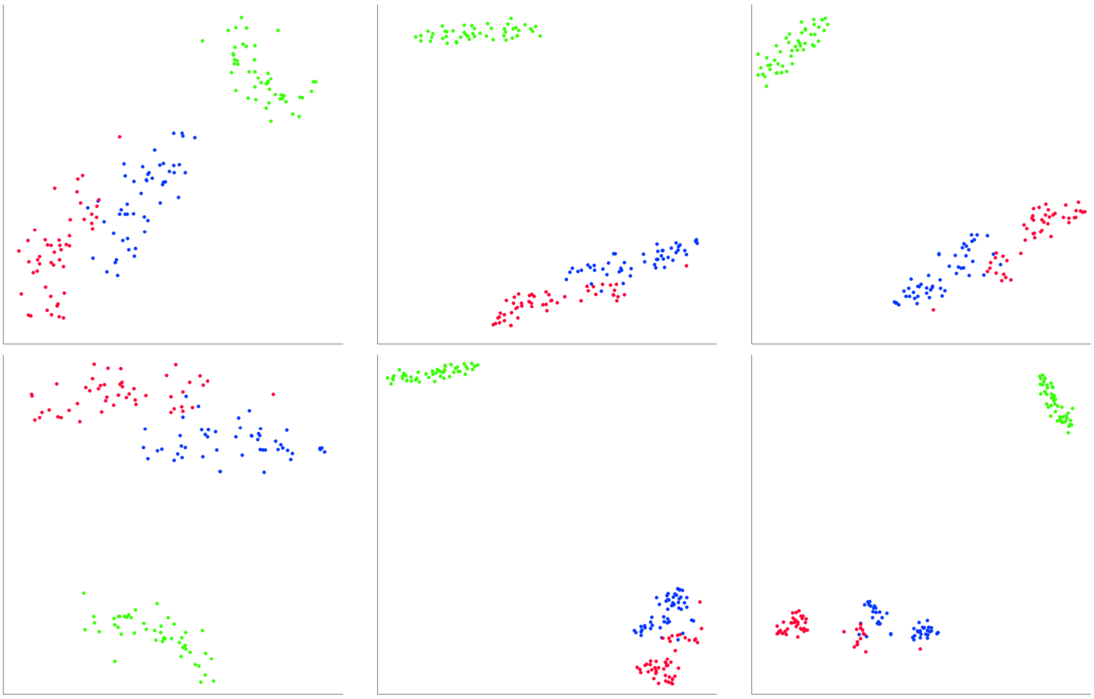


Figure 6.3: Visualization of the Iris dataset: first row: SNE (left), t-SNE (middle), and HSSNE with $\omega = 1.2$ (right). Second row: α-SNE (left), αt-SNE (middle), and α-HSSNE with $\omega = 1.2$ (right). $\alpha = 0.25$ for all the results in the second row.

smaller value of α ($\alpha = 0.9$), the curves become more distinct and result in a better visualization. t-SNE results in more distinguished clusters having larger margins, compared to SNE. The result of αt-SNE (using the same value of α) is similar to the one obtained from t-SNE, but the curvature of the lines is more preserved in some classes, especially for the four aligned cars (four aligned 'c' shaped curves).

The curves in t-SNE and αt-SNE might seem rather over-separated. In order to obtain a visualization which lies between the two former cases, HSSNE (and α-HSSNE, accordingly) with a moderate tail-heaviness parameter can be adopted. The last column in Figure 6.4 illustrates the effect of using HSSNE and α-HSSNE, respectively, with $\omega = 0.5$. As can be seen, the results lie between those obtained by using Gaussian and t-distribution in the map. However, again, α-HSSNE preserves the curvature better than HSSNE.

Figure 6.5 illustrates the visualization results of the Sculpt Faces dataset. The color of the scatter plots depicts the lighting direction. As can be seen, SNE performs well by projecting the pose and lighting on two different or-

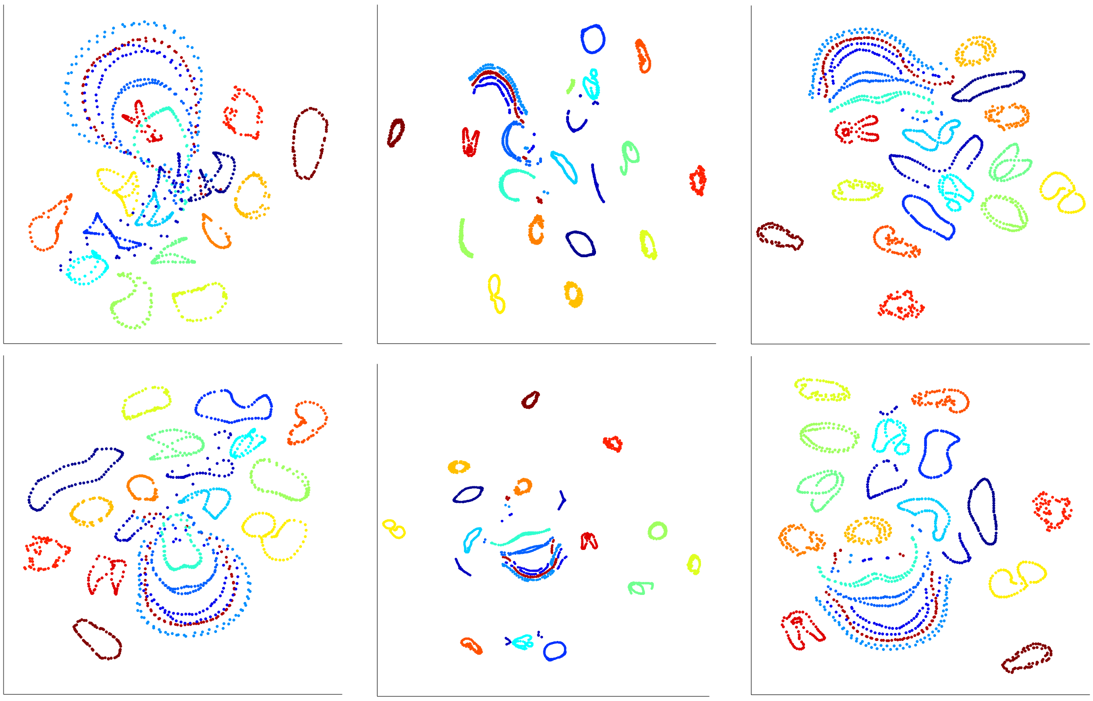


Figure 6.4: Visualization of the COIL-20 dataset: first row: SNE (left), t-SNE (middle), and HSSNE with $\omega = 0.5$ (right). Second row: α-SNE (left), αt-SNE (middle), and α-HSSNE with $\omega = 0.5$ (right). $\alpha = 0.9$ for all the results in the second row.

thogonal directions. t-SNE fails to preserve the lighting structure, however, it projects the pose properly. On the other hand, using a properly selected tail-heaviness parameter ($\omega = 0.5$), HSSNE results in a better visualization than the both former cases. α-SNE ($\alpha = 0.5$) also produces a satisfactory result which separates the pose and lighting information. However, the corners of the map are slightly shrunk compared to HSSNE.

6.1.3 Fine-grained Analysis of the Embedding

To obtain a more fine-grained representation of the embeddings by means of a pointwise quality measure, we adopt the approach in [40]. The pointwise quality measure provides a way to evaluate the faithfulness of the embedding in the sense of preserving the neighborhood structure of each datapoint. This is especially useful to analyze the performance of the DR technique on a dataset having an underlying manifold.

As before, let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^D$ be the set of high-dimensional datapoints

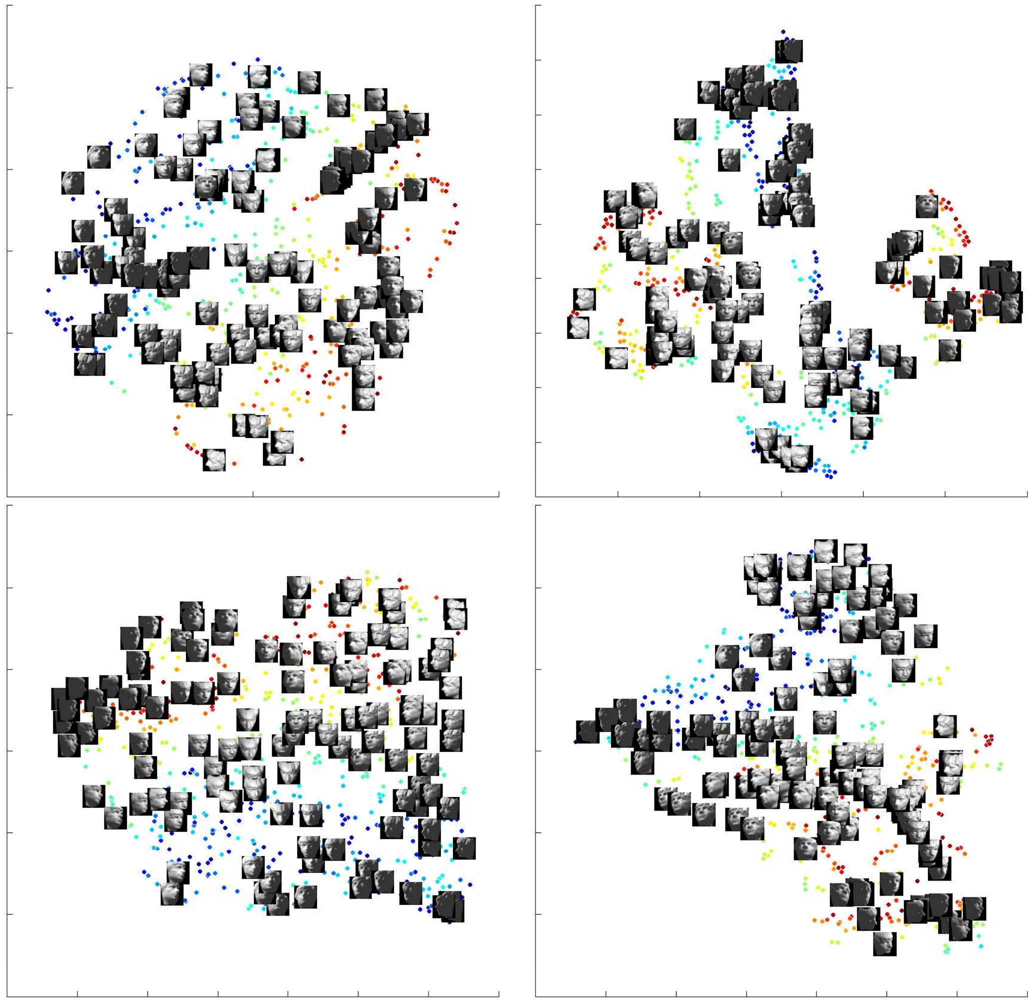


Figure 6.5: Visualization of the SculptFaces dataset: first row: SNE (left), t-SNE (right). Second row: HSSNE with $\omega = 0.5$ (left) and α-SNE with $\alpha = 0.5$ (right).

and let $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N \in \mathbb{R}^d$ be the set of low-dimensional map points, obtained by applying a DR method. Let Δ_{ij} be the distance from $\mathbf{x}_i$ to $\mathbf{x}_j$ in $\mathbb{R}^D$ and δ_{ij} be the distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ in $\mathbb{R}^d$. The rank of $\mathbf{x}_j$ with respect to $\mathbf{x}_i$ in $\mathbb{R}^D$ is defined as

$$\Xi_{ij} = |\{k|\Delta_{ik} < \Delta_{ij} \text{ or } (\Delta_{ik} = \Delta_{ij} \text{ and } 1 \leq k < j \leq N)\}|\,, \tag{6.1}$$

where in (6.1) an initial ordering for the points is assumed for the equality part. The rank of $\mathbf{y}_j$ with respect to $\mathbf{y}_i$ in $\mathbb{R}^d$ is defined in a similar manner

$$\xi_{ij} = |\{k|\delta_{ik} < \delta_{ij} \text{ or } (\delta_{ik} = \delta_{ij} \text{ and } 1 \leq k < j \leq N)\}|\,. \tag{6.2}$$

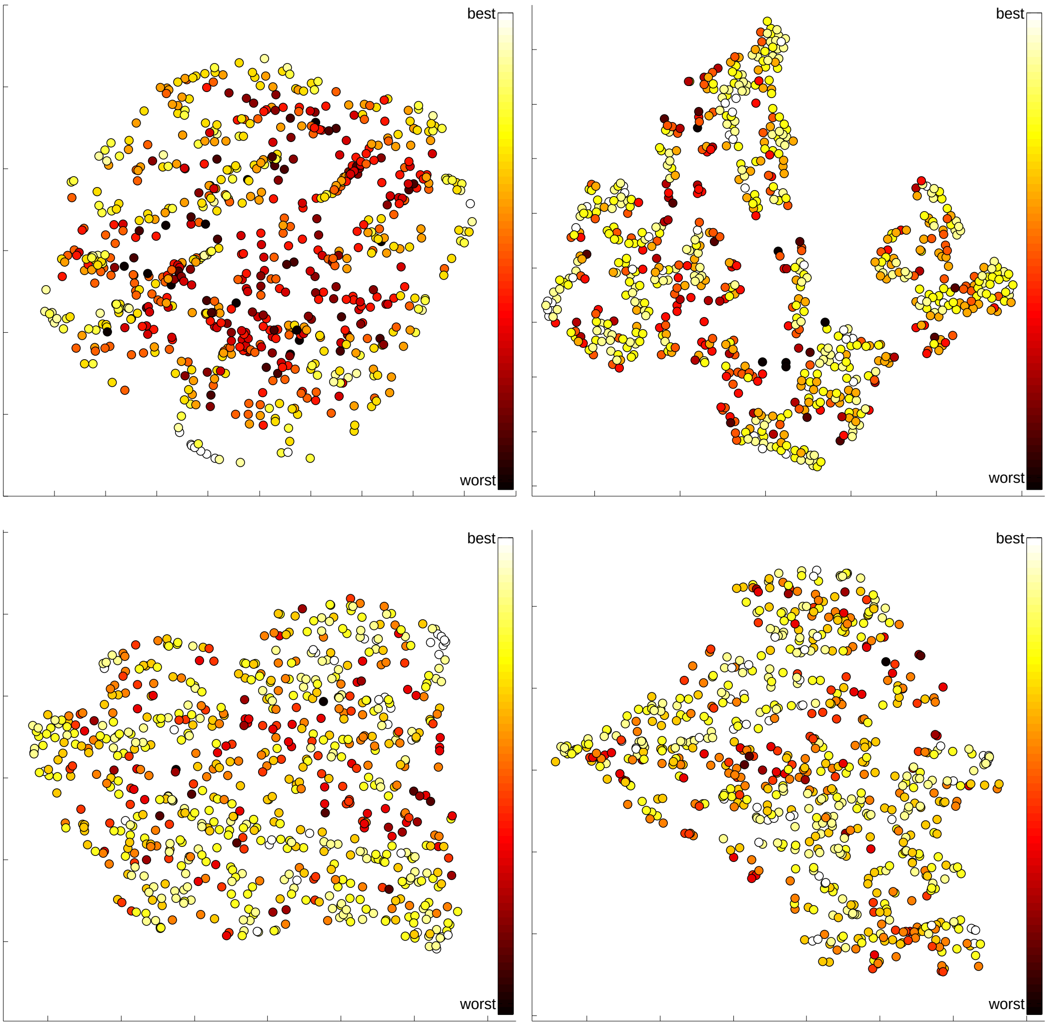


Figure 6.6: Pointwise quality measure for the embeddings shown in Figure 6.5: The values are shown using a hot colormap where white indicates the best performance while dark specifies the worst performance.

The difference $R_{ij} = \xi_{ij} - \Xi_{ij}$ is called the *rank error*. For a perfect embedding, all the ranks are preserved in the embedding and the rank error is zero for all pairs of points. However, this is impossible practically because of the limitations on embedding a high-dimensional data in a space with a considerably lower dimension (see [34] for more details). These cause the points to make intrusions ($\Delta ij > \delta_{ij}$) or extrusions ($\Delta ij < \delta_{ij}$). The histogram of the rank errors for an embedding is called the *co-ranking matrix* $\mathbf{C}$ [35], with entries

$$\mathbf{C}_{kl} = \left|\{(i,j)|\Xi_{ij} = k, \xi_{ij} = l\}\right| . \tag{6.3}$$

Many quality measures such as local continuity meta-criterion (LCMC) [13], trustworthiness & continuity (T&C) [26, 60], and mean relative rank errors (MRRE) [34] can be expressed as a weighted sum of the entries of the co-ranking matrix for entries $k \leq K$ and/or $l \leq K$ for a fixed number of neighbors K. Remarkably, we consider an unweighted sum of the entries of the co-ranking matrix $\mathbf{C}$ as a quality measure (called *Quality*) as follows

$$Q_{N\mathcal{X}}(K) = \frac{1}{NK}\sum_{k=1}^{K}\sum_{l=1}^{K}\mathbf{C}_{kl} = \frac{1}{NK}\sum_{i=1}^{N}\left|\mathcal{A}_{\mathbf{x}_i} \cap \mathcal{B}_{\mathbf{y}_i}\right| , \tag{6.4}$$

in which, $\mathcal{A}_{\mathbf{x}_i} = \{j|\Delta_{ij} \leq K\}$ and $\mathcal{B}_{\mathbf{y}_i} = \{j|\delta_{ij} \leq K\}$ are the sets of of indices of the K nearest neighbors of the datapoint $\mathbf{x}_i$ in the high-dimensional space and correspondingly, $\mathbf{y}_i$ in the embedding, respectively. Thus, $Q_{N\mathcal{X}}(K)$ can be seen as the average ratio of the K nearest neighbors which coincide in the mapping from the original data to the embedding.

Usually, a curve of $Q_{N\mathcal{X}}(K)$ is plotted for different values of K to compare the performances of different methods or, to compare the different embeddings obtained from a single method using different parameters or initialization. However, the user often needs a means to reason about the embedding; that is, to detect the regions where the original data structure is faithfully represented and also, distinguish those where the embedding may not be so reliable. Therefore, a more fine-grained measure of performance needs to be considered.

The co-ranking matrix can be seen as the joint histogram of the ranks in the high-dimensional space and the low-dimensional embedding [35]. Thus, the co-ranking matrix can be decomposed into the sum of pointwise co-ranking matrices evaluated for each point $\mathbf{x}_i \in \mathcal{X}$, that is,

$$\mathbf{C}^{\mathbf{x}_i}_{kl} = \left|\{j|\Xi_{ij} = k, \quad \xi_{ij} = l\}\right| . \tag{6.5}$$

Therefore, the pointwise contribution of each point to $Q_{N\mathcal{X}}(K)$ is calculated as

$$Q^{\mathbf{x}_i}_{N\mathcal{X}}(K) = \frac{1}{K}\sum_{k=1}^{K}\sum_{l=1}^{K}\mathbf{C}_{kl} = \frac{1}{K}\left|\mathcal{A}_{\mathbf{x}_i} \cap \mathcal{B}_{\mathbf{y}_i}\right| , \tag{6.6}$$

and the quality measure defined in (6.4) can be obtained by averaging $Q^{\mathbf{x}_i}_{N\mathcal{X}}(K)$ over all map points,

$$Q_{N\mathcal{X}}(K) = \frac{1}{N}\sum_{i} Q^{\mathbf{x}_i}_{N\mathcal{X}}(K) . \tag{6.7}$$

Each map point in the embedding can be colored based on the pointwise quality measure $Q_{N\mathcal{X}}^{\mathbf{x}_i}(K)$. This can be used to illustrate a sequential color scheme for a given embedding and evaluate the reliable regions for reasoning. The parameter K can be either set to a local extermum of the $Q_{N\mathcal{X}}(K)$ curve or, selected interactively based on the user's preferences.

Figure 6.6 illustrates the pointwise quality measures of the embeddings shown in Figure 6.5. In all the figures, the neighborhood size K is fixed to 10 and the Quality is shown as a hot colormap. As can be seen, the embedding obtained from SNE contains several unreliable regions, especially in the center of the map where the crowding problem takes place. t-SNE, on the other hand, yields a better result. However, the manifold is torn into several smaller patches. While the quality map indicates small error for the points in the inner parts of the patches, several strong topological mismatches can be seen from the dark points in the borders of the embedding. These regions are where the manifold is torn apart to be projected on a two-dimensional space. Finally, both HSSNE and α-SNE produce satisfactory results with respect to the pointwise quality measure. The overall $Q_{N\mathcal{X}}(K)$ value for these embeddings, averaged over all the map points, is equal to 0.7191 and 0.7378, respectively.

6.2 α-Optimization Results

In this section, we illustrate the performance of our proposed method on three different datasets and compare the results with those obtained with SNE and t-SNE. As previously stated, α-SNE and SNE methods coincide exactly when $\alpha = 1$. However, we show that this value is far from optimal, and a proper selection of the α parameter results in a substantial improvement. We also show that the optimal values of α obtained from EDA coincide with those obtained from quality measures.

6.2.1 Quality Measures

As the quality measures, we consider the classification accuracy using a k-nearest neighbors classifier with $k = 3$. We also use area under receiver operating characteristic (ROC) curve (AUC) which is the primary measure of performance in any retrieval task. We fix the neighborhood size in the input space to 20-nearest neighbors and vary the number of neighbors in the output space from 1 to 100 to calculate precision and recall. For each dataset, we repeat the experiments 20 times with different random initializations and report the averages over all the trials.

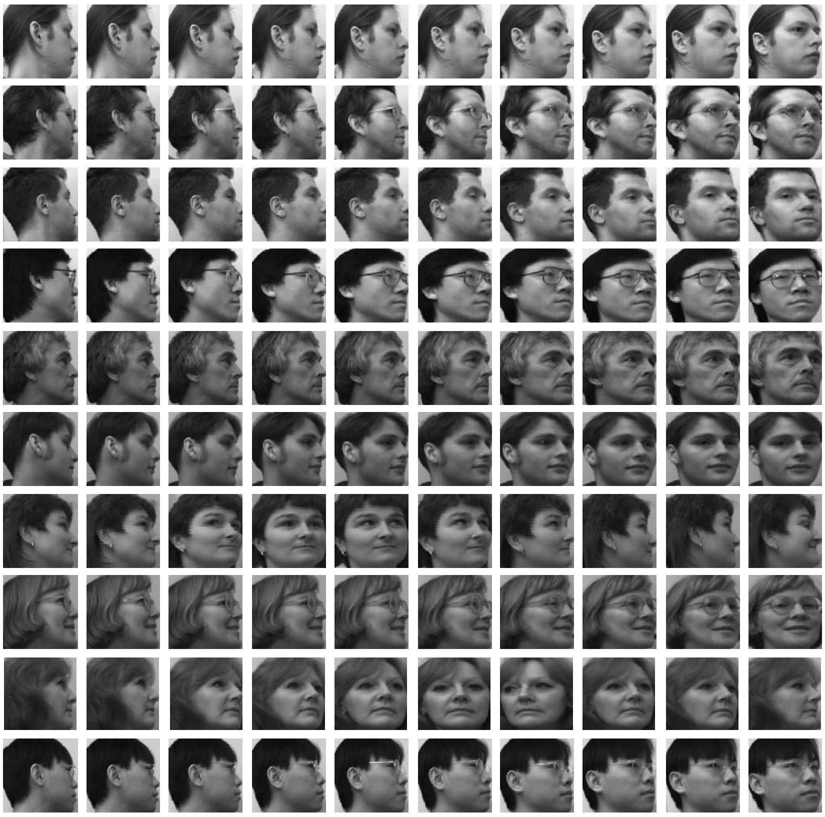


Figure 6.7: Examples of the images drawn from the UMist Faces dataset.

6.2.2 Datasets

As the first dataset, we consider UMist Faces dataset [3] which contains 112 × 92 sized images of 20 people from different views (575 images in total). Examples of the images from this dataset are shown in Figure 6.7.

For the second dataset, we consider the Texture database from UCL repository [5] which contains measurements of 10 fourth order modified moments in 4 different orientations (40 in total) for texture images from 11 classes (500 instances per each class). We take a subset of 6 classes (3000 instances). As the final dataset, we present our results on a subset of 6000 digits, randomly selected from the MNIST handwritten digits dataset [30]. The examples of the handwritten digits drawn from the MNIST are shown in Figure 6.8

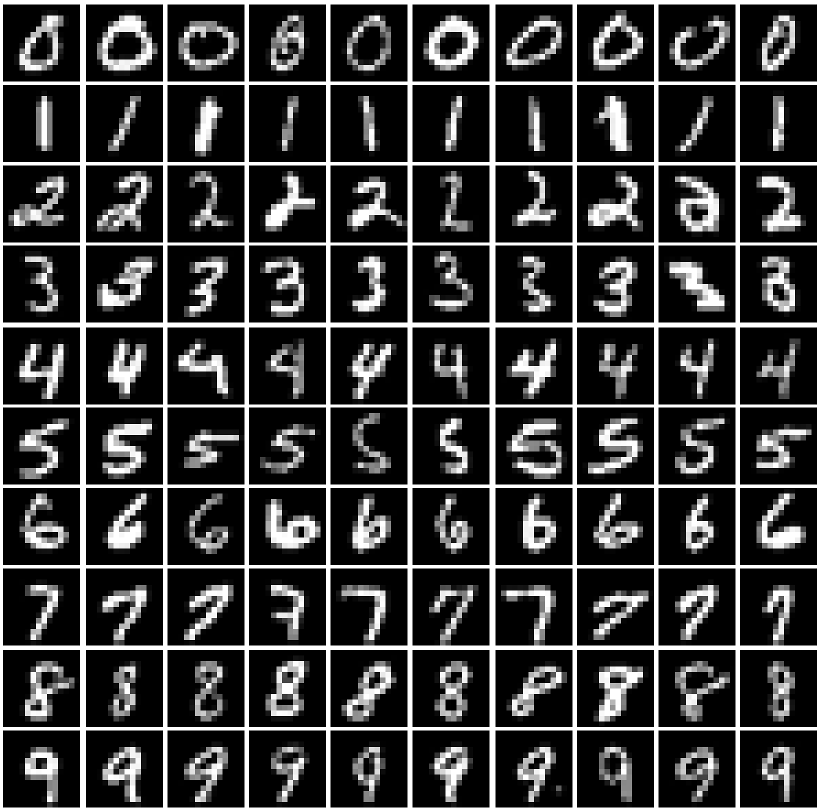


Figure 6.8: Examples of the handwritten digits drawn from the MNIST dataset.

6.2.3 Results

Figure 6.9 illustrates the classification accuracy and AUC curves as well as the result of α-optimization for different values of α. As can be seen, there is a considerable improvement over the original SNE method ($\alpha = 1$) in the sense of both classification accuracy and AUC by using a smaller value of α. Additionally, there exists consistency among the performance curves and the one obtained from α optimization using the EDA method. The optimal value of α obtained from EDA coincides with the optimal value of the two performance curves in most cases, or, at least, peaks near the optimal value which has a satisfactory performance itself. Please note that the criterion in EDA is to find the α which corresponds to the intrinsic characteristics of the data distribution, rather than the one maximizing accuracy or AUC. The criteria for finding the optimal value are completely different from those used for calculation of classification accuracy or AUC. Therefore, the optimal α by EDA may not always correspond to the point having the best perfor-

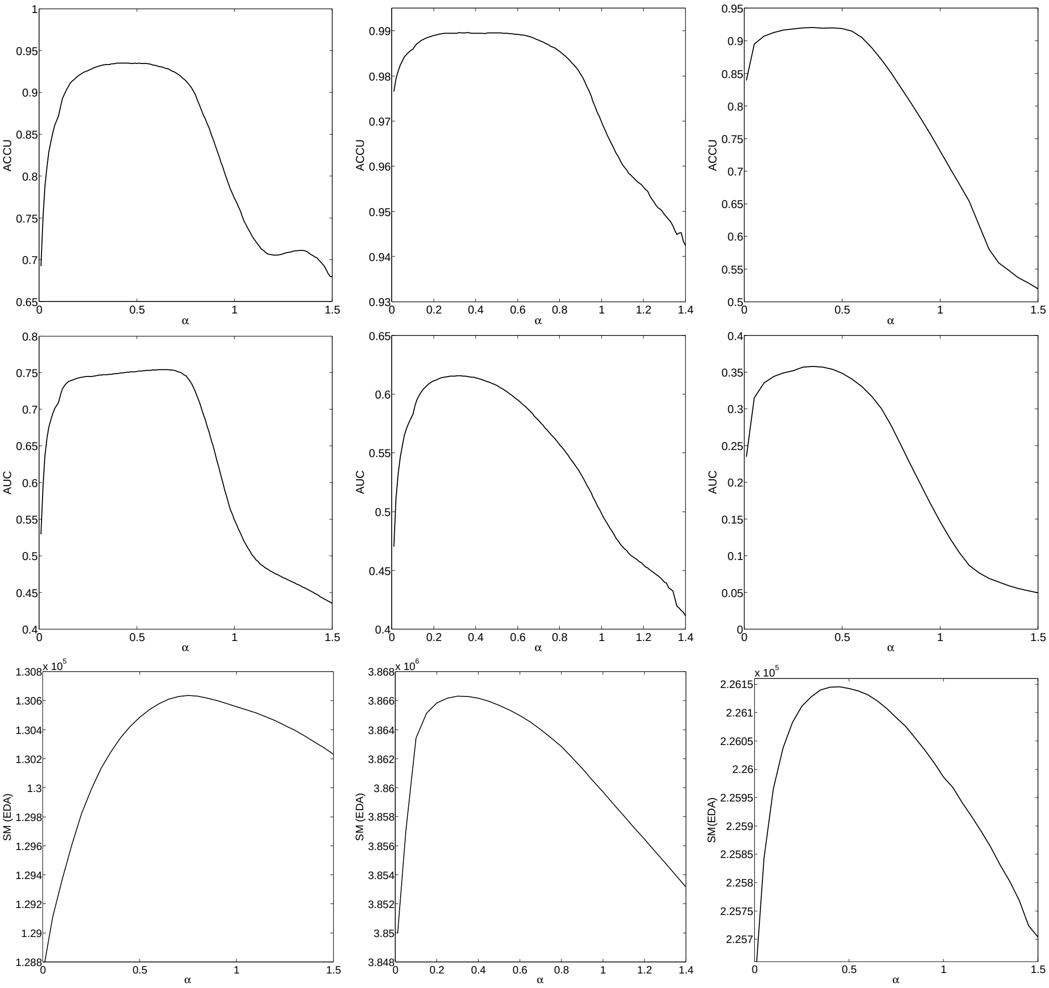


Figure 6.9: Performance curves: classification accuracy (first row), AUC (second row) and negative SM objective function of EDA (third row), for datasets: UMist Faces (first column), Texture (second column) and MNIST (third column).

mance, but the one which faithfully represents the data distribution. This can be easily seen in the following, by plotting the maps corresponding to the optimal value of α.

Figure 6.10 shows results of visualization of the datasets obtained from SNE and t-SNE along with those achieved from α-SNE using the optimal value of α found by EDA method. As it can be seen, in all cases, α-SNE has much better performance compared to SNE. In the UMist Faces dataset, SNE forms overlapping curves for images from different persons. However, both t-SNE and α-SNE result in well separated curves. Moreover, using SNE, the

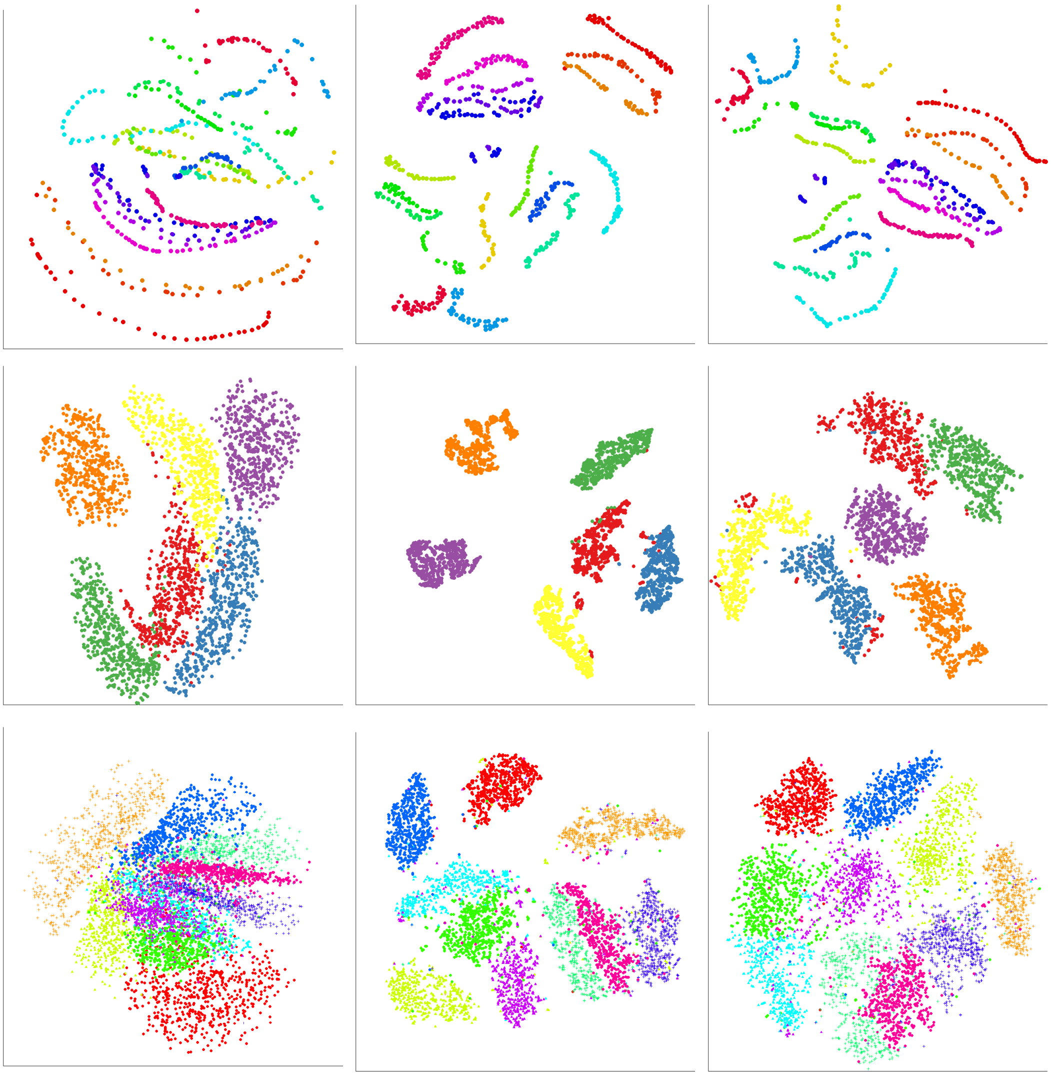


Figure 6.10: Visualization results on datasets: UMist Faces (first row), Texture (second row) and MNIST (third row), using SNE (first column), t-SNE (second column) and α-SNE (third column).

clusters become very close in the Texture dataset. t-SNE produces clusters which are well separated, but rather over-compressed. α-SNE establishes a balance between the two cases by producing separated but finely scattered clusters. On the subset of the MNIST dataset, SNE again produces clusters which are crowded in the center of the map and therefore, fails to separate the digits from different classes. On the other hand, t-SNE and α-SNE both produce well separated clusters. Again, α-SNE results in more spread clusters compared to t-SNE.

The α-optimization procedure is purely unsupervised and eliminates the need for any other information except the input data. This property makes the method suitable for unlabeled datasets where quality measures such as classification accuracy can not be applied. It is worth to mention that the joint optimization of (α, ϕ) is possible using only a small sub-sample of the data (in our case, using around 100 randomly selected datapoints produced satisfactory results). Thus, the method yields a significant speed-up compared to other quality measures such as AUC, when dealing with large datasets.

Chapter 7

Conclusions and Future Work

We presented a natural generalization to the basic SNE method by considering α-divergence as the cost function. The proposed method, α-SNE, avoids the crowding problem associated with the SNE by providing a much smoother gradient for optimization, having a better balance between the attraction and repulsion forces. This eliminates the need for considering distributions with heavier tail than Gaussian in the mapping. Furthermore, we provided a framework to select the optimal α value for a given dataset. The optimal value for α is the one which explains the data best (approximately). The results show that our proposed method, with a proper selection of the α parameter, outperforms the original SNE method by providing more distinguished clusters. The results are comparable with those obtained from t-SNE or in some cases, visually even better. Additionally, the results of our α-optimization framework are consistent with standard quality measures for dimensionality reduction. The α-optimization can be performed efficiently using only a small sub-sample of the data, providing a large speed-up over other quality measures.

Possible extensions of the proposed method would be online optimization of the α parameter using EDA, that is, performing α-optimization iteratively along with the optimization of the cost function. Our α-decay procedure for finding the map would be useful here such that we can gradually reduce α in each iteration until the optimal value is reached. The rest of the iterations would be fixing α and iteratively reducing the cost function until the minimum is obtained.

Bibliography

[1] Columbia University Image Library (COIL-20). `http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php`.

[2] Data sets for nonlinear dimensionality reduction Sculpt Faces Dataset. `http://web.mit.edu/cocosci/isomap/datasets.html`.

[3] The University of Sheffield Image Engineering Labratory UMist Faces Dataset. `http://www.sheffield.ac.uk/eee/research/iel/research/face`.

[4] UCI Machine Learning Repository Fisher's Iris Dataset. `https://archive.ics.uci.edu/ml/datasets/Iris`.

[5] UCL Repository Texture Dataset. `https://sites.uclouvain.be/elen/`.

[6] Amari, S.-i. Information geometry and its applications: Convex function and dually flat manifold. In *Emerging Trends in Visual Computing*, F. Nielsen, Ed., vol. 5416 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2009, pp. 75–102.

[7] Aronszajn, N. Theory of reproducing kernels. *Transactions of the American Mathematical Society 68* (1950).

[8] Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. Clustering with bregman divergences. *J. Mach. Learn. Res. 6* (Dec. 2005), 1705–1749.

[9] Belkin, M., and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS 14* (2001), pp. 585–591.

[10] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[11] Borg, I., and Groenen, P. *Modern Multidimensional Scaling: Theory and Applications.* Springer, 2005.

[12] Bunte, K., Haase, S., Biehl, M., and Villmann, T. Stochastic neighbor embedding (sne) for dimension reduction and visualization using arbitrary divergences. *Neurocomput. 90* (Aug. 2012), 23–45.

[13] Chen, L., and Buja, A. Local multidimensional scaling for nonlinear dimension reduction, graph layout and proximity analysis. *J. Am. Stat. Assoc. 104 (485)* (2006), 209 – 219.

[14] Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S.-i. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation.* Wiley Publishing, 2009.

[15] Cook, J., Sutskever, I., Mnih, A., and Hinton, G. E. Visualizing similarity data with a mixture of maps. In *J. Mach. Learn. Res. Proceedings* (2007), vol. 2, pp. 67–74.

[16] Cover, T. M., and Thomas, J. A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing).* Wiley-Interscience, 2006.

[17] Demartines, P., and Herault, J. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *Trans. Neur. Netw. 8*, 1 (Jan. 1997), 148–154.

[18] Dijkstra, E. A note on two problems in connexion with graphs. *Numerische Mathematik 1*, 1 (1959), 269–271.

[19] Dikmen, O., Yang, Z., and Oja, E. Learning the information divergence. *arXiv:1406.1385 [cs.LG]* (2014).

[20] Forgy, E. W. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics 21* (1965), 768–769.

[21] Grady, L. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 28*, 11 (Nov. 2006), 1768–1783.

[22] Hennequin, R., David, B., and Badeau, R. Beta-Divergence as a Subclass of Bregman Divergence. *Signal Processing Letters, IEEE 18*, 2 (Feb. 2011), 83–86.

[23] Hinton, G., and Roweis, S. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15* (2003), pp. 833–840.

[24] Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res. 6* (Dec. 2005), 695–709.

[25] Ji, S., and Ye, J. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning* (New York, NY, USA, 2009), ICML '09, ACM, pp. 457–464.

[26] Kaski, S., Nikkilä, J., Oja, M., Venna, J., Törönen, P., and Castrén, E. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics 4* (2003), 48.

[27] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by simulated annealing. *Science 220*, 4598 (1983), 671–680.

[28] Kohonen, T., Schroeder, M. R., and Huang, T. S., Eds. *Self-Organizing Maps*, 3rd ed. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[29] Kruskal, J. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (1964), 1–27.

[30] Lecun, Y., and Cortes, C. The MNIST database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

[31] Lee, J. A. A robust nonlinear projection method. *European Symposium on Artificial Neural Networks Bruges (Belgium)* (2000), 26–28.

[32] Lee, J. A., Lendasse, A., and Verleysen, M. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing 57* (2004), 49–76.

[33] Lee, J. A., Renard, E., Bernard, G., Dupont, P., and Verleysen, M. Type 1 and 2 mixtures of kullback-leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomput. 112* (July 2013), 92–108.

[34] Lee, J. A., and Verleysen, M. *Nonlinear Dimensionality Reduction*, 1st ed. Springer Publishing Company, Incorporated, 2007.

[35] Lee, J. A., and Verleysen, M. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing 72 (7-9)* (2009), 1431–1443.

[36] Lerner, B., Guterman, H., Aladjem, M., and Dinstein, I. On the initialisation of sammon's nonlinear mapping. *Pattern Anal. Appl. 3*, 1 (2000), 61–68.

[37] Madsen, R. E., Hansen, L. K., and Winther, O. Singular value decomposition and principal component analysis. Tech. rep., 2004.

[38] Memisevic, R. Kernel information embeddings. In *Proceedings of the 23rd International Conference on Machine Learning* (New York, NY, USA, 2006), ICML '06, ACM, pp. 633–640.

[39] Minka, T. Divergence measures and message passing. Tech. rep., 2005.

[40] Mokbel, B., Lueks, W., Gisbrecht, A., and Hammer, B. Visualizing the quality of dimensionality reduction. *Neurocomputing 112*, 0 (2013), 109 – 123.

[41] Nene, S. A., Nayar, S. K., and Murase, H. Columbia Object Image Library (COIL-20). Tech. Rep. CUCS-005-96, 1996.

[42] Nesterov, Y. Smooth minimization of non-smooth functions. *Math. Program. 103*, 1 (May 2005), 127–152.

[43] Nesterov, Y. *Gradient methods for minimizing composite objective function.* Center for operations research and econometrics (CORE), Catholic University of Louvein (UCL), 2007.

[44] Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *NIPS* (2001), MIT Press, pp. 849–856.

[45] Nocedal, J., and Wright, S. J. *Numerical Optimization*, 2nd ed. Springer, New York, 2006.

[46] Pearson, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine 2*, 6 (1901), 559–572.

[47] Roweis, S. T., and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science 290* (2000), 2323–2326.

[48] Rustamov, R. M. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), SGP '07, Eurographics Association, pp. 225–233.

[49] Sammon, J. W. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput. 18*, 5 (May 1969), 401–409.

[50] Schlkopf, B., Smola, A. J., and Müller, K. R. Kernel principal component analysis. *Advances in kernel methods: support vector learning* (1999), 327–352.

[51] Shi, J., and Malik, J. Normalized cuts and image segmentation. *Pattern Anal. Mach. Intell., IEEE Transactions on 22*, 8 (Aug 2000), 888–905.

[52] Simon, H. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering 2*, 2-3 (1991), 135 – 148.

[53] Sun, J., Crowe, M., and Fyfe, C. Extending metric multidimensional scaling with bregman divergences. *Pattern Recognition 44*, 5 (2011), 1137–1154.

[54] Tenenbaum, J. B., de Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319.

[55] Teng, L., Li, H., Fu, X., Chen, W., and Shen, I.-F. Dimension reduction of microarray data based on local tangent space alignment. In *IEEE ICCI* (2005), IEEE, pp. 154–159.

[56] Torgerson, W. Multidimensional scaling: I. theory and method. *Psychometrika 17*, 4 (1952), 401–419.

[57] Van der Maaten, L. An introduction to dimensionality reduction using matlab. *Report 1201* (2007), 07–07.

[58] Van der Maaten, L., and Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res. Proceedings 9* (Nov. 2008), 2579–2605.

[59] Vandenberghe, L., and Boyd, S. Semidefinite programming. *SIAM Review 38* (1994), 49–95.

[60] Venna, J., and Kaski, S. Local multidimensional scaling. *Neural Networks 19*, 6-7 (2006), 889 – 899.

[61] Venna, J., Peltonen, J., Nybo, K., Aidos, H., and Kaski, S. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J. Mach. Learn. Res. 11* (Mar. 2010), 451–490.

[62] Weinberger, K. Q., Sha, F., and Saul, L. K. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the Twenty-first International Conference on Machine Learning* (New York, NY, USA, 2004).

[63] Xie, B., Mu, Y., Tao, D., and Huang, K. m-sne: Multiview stochastic neighbor embedding. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 41*, 4 (Aug 2011), 1088–1096.

[64] Yang, Z., King, I., Xu, Z., and Oja, E. Heavy-tailed symmetric stochastic neighbor embedding. In *NIPS* (2009), pp. 2169–2177.

Appendix A

Calculation of the Gradient

The derivative of (4.9) with respect to $\mathbf{y}_i$ can be found using the chain rule

$$\frac{\partial C_{\alpha\text{-SNE}}}{\partial \mathbf{y}_i} = \sum_{j\neq i} \frac{\partial C_{\alpha\text{-SNE}}}{\partial q_{ij}} \frac{\partial q_{ij}}{\partial \mathbf{y}_i} + \sum_{k\neq i} \sum_{j\neq k} \frac{\partial C_{\alpha\text{-SNE}}}{\partial q_{kj}} \frac{\partial q_{kj}}{\partial \mathbf{y}_i} \,. \tag{A.1}$$

Each term in (A.1) is shown in more details as follows

$$\frac{\partial C_{\alpha\text{-SNE}}}{\partial q_{ij}} = \begin{cases} -\frac{1}{\alpha}\left(\frac{p_{ij}}{q_{ij}}\right)^{\alpha}, & \alpha \neq 0 \\ \log(\frac{q_{ij}}{p_{ij}}) + 1, & \alpha = 0 \end{cases} \quad , \tag{A.2}$$

$$\begin{aligned} \frac{\partial q_{ij}}{\partial \mathbf{y}_i} &= -2(\mathbf{y}_i - \mathbf{y}_j) q_{ij} + q_{ij} \sum_{k\neq i} 2(\mathbf{y}_i - \mathbf{y}_k) q_{ik} \\ &= 2\mathbf{y}_j q_{ij} - q_{ij} \sum_{k\neq i} 2\mathbf{y}_k q_{ik}, \quad j \neq i \,, \end{aligned} \tag{A.3}$$

$$\frac{\partial q_{ji}}{\partial \mathbf{y}_i} = 2(\mathbf{y}_j - \mathbf{y}_i) q_{ji}(1 - q_{ji}), \quad j \neq i \,, \tag{A.4}$$

$$\frac{\partial q_{kj}}{\partial \mathbf{y}_i} = -2(\mathbf{y}_k - \mathbf{y}_i) q_{kj} q_{ki}, \quad j \neq k, i \,. \tag{A.5}$$

So (A.1) can be written as follows

$$\begin{aligned} \frac{\partial C_{\alpha\text{-SNE}}}{\partial \mathbf{y}_i} = &- 2\sum_{j\neq i} \left[\frac{1}{\alpha} p_{ij}^{\alpha} q_{ij}^{1-\alpha} \left(\mathbf{y}_j - \sum_{k\neq i} \mathbf{y}_k q_{ik} \right) \right] \\ &- 2\sum_{j\neq i} \left[\frac{1}{\alpha} p_{ji}^{\alpha} q_{ji}^{1-\alpha} (1 - q_{ji})(\mathbf{y}_i - \mathbf{y}_j) \right] \\ &+ 2\sum_{k\neq i} \sum_{j\neq i,k} \left[\frac{1}{\alpha} p_{kj}^{\alpha} q_{kj}^{1-\alpha} q_{ki} (\mathbf{y}_k - \mathbf{y}_i) \right], \quad \alpha \neq 0 \,. \end{aligned}$$

Defining $\theta_i = \sum_{j \neq i} p_{ij}^{\alpha} q_{ij}^{1-\alpha}$, we have

$$
\begin{aligned}
\frac{\partial C_{\alpha\text{-SNE}}}{\partial \mathbf{y}_i} &= \frac{2}{\alpha} \sum_{j \neq i} \Big[\big((p_{ji}^{\alpha} q_{ji}^{1-\alpha} (1 - q_{ji}) - (\theta_j - p_{ji}^{\alpha} q_{ji}^{1-\alpha}) q_{ji} \big) \, \mathbf{y}_i \\
&- \big(q_{ij} \theta_i - p_{ij}^{\alpha} q_{ij}^{1-\alpha} - p_{ji}^{\alpha} q_{ji}^{1-\alpha} (1 - q_{ji}) + q_{ji} (\theta_j - p_{ji}^{\alpha} q_{ji}^{1-\alpha}) \big) \, \mathbf{y}_j \Big] \, , \quad \alpha \neq 0 \, .
\end{aligned}
$$

Finally, we have the compact form (5.2). The gradient for the case $\alpha = 0$ can be obtained in the limit $\alpha \to 0$ where we have

$$
\frac{\partial C_{\alpha\text{-SNE}}}{\partial \mathbf{y}_i} = 2 \sum_{j \neq i} (\mathbf{y}_i - \mathbf{y}_j) \left(q_{ij} D_{\mathrm{KL}}(\mathbf{q}_i \| \mathbf{p}_i) - q_{ij} \log \frac{q_{ij}}{p_{ij}} + q_{ji} D_{\mathrm{KL}}(\mathbf{q}_j \| \mathbf{p}_j) - q_{ji} \log \frac{q_{ji}}{p_{ji}} \right) . \tag{A.6}
$$

For the symmetric case, after similar calculations or, alternatively, using the Lagrangian technique proposed in [64], we have

$$
\frac{\partial C_{\alpha\text{-SSNE}}}{\partial \mathbf{y}_i} = \begin{cases} \frac{4}{\alpha} \sum_{j \neq i} (\mathbf{y}_i - \mathbf{y}_j) \left(p_{ij}^{\alpha} q_{ij}^{1-\alpha} - \theta q_{ij} \right) & \alpha \neq 0 \\ 4 \sum_{j \neq i} (\mathbf{y}_i - \mathbf{y}_j) \left(q_{ij} D_{\mathrm{KL}}(Q \| P) - q_{ij} \log(q_{ij}/p_{ij}) \right) & \alpha = 0 \end{cases} . \tag{A.7}
$$

with $\theta = \sum_{j \neq i} p_{ij}^{\alpha} q_{ij}^{1-\alpha}$ and $D_{\mathrm{KL}}(Q \| P) = \sum_{j \neq i} q_{ij} \log(q_{ij}/p_{ij})$ is calculated over the joint distributions P and Q.