Γ-SNE for feed-forward data visualization

Iain Rice,
Aston University,
Aston Triangle,
Birmingham,
B4 7ET, UK.
i.rice@aston.ac.uk

July 31, 2017

Abstract

t-distributed Stochastic Neighbour Embedding (t-SNE) is one of the most popular nonlinear dimension reduction techniques used in multiple application domains. In this paper we propose a variation on the embedding neighbourhood distribution, resulting in Γ-SNE, which can construct a feed-forward mapping using an RBF network. We compare the visualizations generated by Γ-SNE with those of t-SNE and provide empirical evidence suggesting the network is capable of robust interpolation and automatic weight regularization.

1 Introduction

Data which is high-dimensional in the observation space is naturally impossible to humanly interpret. The notion of information visualization transforms these observations, generating a low (2 or 3) dimensional representation of the data. This allows for valuable insight into complex data structures, for instance the financial applications of [23]. An argument for the benefits of 3-dimensional information visualization is presented in [14], however following the literature standard we restrict our experiments to 2-dimensional visualizations. There has been much interest in recent years in the creation of nonlinear mappings connecting the observation and visualization spaces, see [17],[28] for a review. One of the most popular mappings which has sparked interest is locally linear embedding [22] whereby local neighbourhoods are preserved by constructing a weighted neighbourhood. Stochastic Neighbour Embedding [15] (SNE) extends this concept by considering global neighbourhood structures, as opposed to local ones, attaching a probability distribution to the neighbourhoods. The mapping is learned by matching a

distribution over latent, visualized, neighbourhoods with that of the observation space. SNE differs from other global mappings such as NeuroScale [19] by placing emphasis on mapping local neighbourhoods at the expense of global reconstructions [20].

This methodology was extended using t-distributions over dissimilarities for the visualization space in t-SNE [29] which forms a part of the visualization framework in [3]. It was found that mismatching the neighbourhood distributions allowed for better local clustering, with examples given in the supplementary material of [29]. Further research into t-SNE has allowed for feed-forward mappings using both kernel methods [5, 13] and deep belief nets [27]. In addition to this, the impact of different measures of 'closeness' of the observed and visualized neighbourhood distributions is discussed in [6, 9].

In order to introduce the method proposed in this paper we provide a brief outline of the t-SNE mapping process. Given a set of observations, $\{\boldsymbol{x}_i\} \in \mathbb{R}^O$ we wish to create a 2-dimensional embedding of the data, $\{\boldsymbol{y}_i\} \in \mathbb{R}^2$, such that it can be visually interpreted using t-SNE. The mapping places a distribution over neighbourhoods:

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

where σ_i^2 is the perplexity corresponding to observation i [15]. These conditional distributions are then symmetrized resulting in an observation neighbourhood distribution:

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

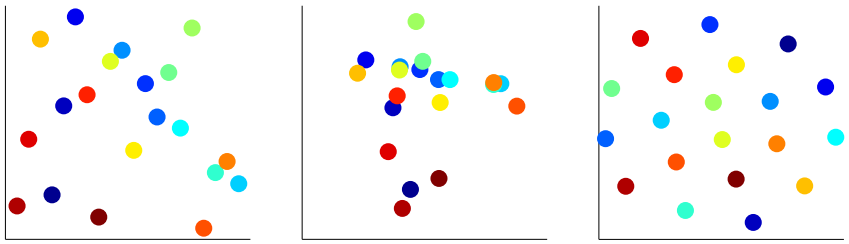
A distribution is then placed over neighbourhoods of the visualized points:

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

t-SNE then attempts to match q_{ij} and p_{ij} by minimising the Kullback-Leibler divergence:

$$C = KL\left(P\|Q\right) = \sum_{ij} p_{ij}\log\left(\frac{p_{ij}}{q_{ij}}\right). \tag{3}$$

There are two prominent issues with the setup of t-SNE in this form. Firstly, we note that q_{ij} is a t-distribution over the dissimilarity between $\{\boldsymbol{y}_i\}$ which we denote d_{ij}. This ensures there is a finite probability that d_{ij} can be negative, which for d_{ij} given by the Euclidean distance in t-SNE is not possible. This distribution is therefore a poor fit to the neighbourhood distances.



(a) (b) (c)

Figure 1: (a) Randomly generated data consisting 20 3-dimensional points, (b) NeuroScale mapping, (c) t-SNE mapping. The t-SNE visualization does not preserve any neighbourhood structure and instead places the data randomly on an approximately uniform grid, whereas the NeuroScale mapping preserves neighbourhood structure, enforcing a local reconstruction.

A further issue is the optimisation of the cost function in equation (3). In order to avoid achieving poor local-minima in the optimisation process a momentum-based gradient descent process is required. t-SNE has achieved impressive clustering results on popular datasets, however it can fail to properly preserve neighbourhoods when the t-distribution is a poor match to the true latent distributions. One simple way to show this is to attempt to cluster 20 3-dimensional randomly generated points where the values of the third dimension are fixed to zero such that this is already a 2-dimensional observation space, as in figure 1. The NeuroScale generated visualization preserves neighbourhood structures whereas t-SNE cannot preserve the neighbourhood structure since the latent distribution over visualized points is not t-distributed.

In order to rectify this issue this paper presents a different mapping to that of SNE or t-SNE, utilising the Gamma distribution for q_{ij}, allowing for improved neighbourhood preservation. This new approach is integrated with a feed-forward mapping to overcome some of the problems facing these algorithms, particularly the ease with which false neighbourhoods are created and the non-convexity of the optimisation process. We further demonstrate using four standard datasets that our approach outperforms t-SNE whilst retaining desirable projective properties allowing for reliable out-of-sample mapping not guaranteed with t-SNE.

2 Method

The difference between Γ-SNE and alternative Neighbour-Embedding algorithms is the change in distribution over the latent distances, q_{ij}. The same distribution over neighbours in the observation space as in t-SNE

and standard SNE from equation (1) is retained. Neighbourhood distances in the visualization space are again determined by the Euclidean distance, $d_{ij} = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|$, but here we place a Gamma distribution, q, over the neighbourhood probabilities. This is a more realistic assumption than that of SNE or t-SNE since the support of $q(d_{ij}|a,b)$ is over $d_{ij} > 0$. The pdf of the Gamma distribution is given by:

$$q(d_{ij}|a,b) = \left(\frac{1}{\Gamma(a)}\right) b^a d_{ij}^{(a-1)} e^{-d_{ij}b}.$$

As with SNE and t-SNE we remove the normalisation terms, opting to perform normalisation over the embedding dissimilarities numerically:

$$q_{ij} = \frac{d_{ij}^{a-1} e^{-d_{ij}b}}{\sum_{k\neq l} d_{kl}^{a} e^{-d_{kl}b}}, \tag{4}$$

where the normalisation term ensures the distribution sums to 1. In the case that $d_{ij} = 0$ we fix $q_{ij} = (\sum_{k\neq l} d_{kl}^{a-1} e^{-d_{kl}b})^{-1}$.

Following the framework of Neighnour Embedding algorithms we wish to match the distributions, p and q, where the natural choice is the Kullback-Leibler divergence as in equation (3). This is one of a wide variety of distance measures over probability distribution functions and in particular is a special case of the α [7], β [1] and γ-divergence [12] families. In [6] it was shown that the performance of t-SNE can be improved when the cost C is replaced with one of these divergence measures. We adopt these developments resulting in the modified cost functions C^α, C^β and C^γ respectively:

$$C^\alpha = \frac{1}{\alpha(\alpha-1)} \sum_{ij} p_{ij}^{\alpha} q_{ij}^{(1-\alpha)} - \alpha p_{ij} + (\alpha-1) q_{ij},$$

$$C^\beta = \sum_{ij} \frac{1}{\beta-1} p_{ij} \left(p_{ij}^{(\beta-1)} - q_{ij}^{(\beta-1)}\right) - \frac{1}{\beta}\left(p_{ij}^{\beta} - q_{ij}^{\beta}\right),$$

$$C^\gamma = \log\left[\frac{\left(\sum_{ij} p_{ij}^{\gamma+1}\right)^{\frac{1}{\gamma(\gamma+1)}} \left(\sum_{ij} q_{ij}^{\gamma+1}\right)^{\frac{1}{\gamma+1}}}{\left(\sum_{ij} p_{ij} q_{ij}^{\gamma}\right)^{\frac{1}{\gamma}}}\right].$$

These cost functions are nonconvex by nature and require optimisation through a gradient-descent procedure. The gradients for the latent visualized points, $\boldsymbol{y}_i$, using the auxiliary variable V_{ij} are given by:

$$V_{ij} = 4\left(\frac{a-1}{d_{ij}} - b\right)(\boldsymbol{y}_i - \boldsymbol{y}_j),$$

$$\frac{\partial C}{\partial \boldsymbol{y}_i} = \sum_j (p_{ij} - q_{ij}) V_{ij},$$

$$\frac{\partial C^{\alpha}}{\partial \boldsymbol{y}_i} = \frac{1}{\alpha}\sum_j \left[p_{ij}^{\alpha} q_{ij}^{1-\alpha} - q_{ij} \sum_{kl} p_{kl}^{\alpha} q_{kl}^{1-\alpha} \right] V_{ij},$$

$$\frac{\partial C^{\beta}}{\partial \boldsymbol{y}_i} = \sum_j \left[q_{ij}^{\beta-1}(p_{ij} - q_{ij}) - q_{ij} \sum_{kl} q_{kl}^{\beta-1}(p_{kl} - q_{kl}) \right] V_{ij},$$

$$\frac{\partial C^{\gamma}}{\partial \boldsymbol{y}_i} = \sum_j \left[\frac{p_{ij} q_{ij}^{\gamma}}{\sum_{kl} p_{kl} q_{kl}^{\gamma}} - \frac{q_{ij}^{\gamma+1}}{\sum_{kl} q_{kl}^{\gamma+1}} \right] V_{ij}.$$

Unlike the learning procedure for SNE and t-SNE we now have a parametric embedding in which we must determine a and b (note the ν parameter used in one-dimensional t-distributions is fixed to unity in t-SNE and therefore requires no optimisation). By defining an auxiliary variable for the normalisation constant, $u = \sum_{k \neq l} d_{kl}^{a-1} e^{-d_{kl} b}$, we have the gradients for a:

$$\frac{\partial C}{\partial a} = \frac{1}{u}\sum_{ij} p_{ij} \log(d_{ij}) - \log(d_{ij}) d_{ij}^{(a-1)} e^{-d_{ij} b}, \tag{5}$$

and for b:

$$\frac{\partial C}{\partial b} = \frac{1}{u}\sum_{ij} -p_{ij} d_{ij} + d_{ij}^{a} e^{-d_{ij} b}. \tag{6}$$

In this form a fixed mapping of a set of data, $\{\boldsymbol{x}_i\}$ can now be constructed using the above gradients. In numerical experiments we have found that optimising the latent points, $\{\boldsymbol{y}_i\}$, in addition to the parameters, a, b, can potentially yield local minima and thus require a stochastic or momentum-based gradient approach to optimising these parameters. An alternative method is to fix a and b prior to optimising the latent points. Initial experiments have found that a maximum likelihood fit of a and b to a PCA-embedded mapping yields a performance drop of less than 1% relative error of the cost function. The non-convexity of the cost function in this form of Γ-SNE is an issue shared also with SNE and t-SNE. We propose to not only construct a visualization of a dataset with Γ-SNE, but also to create a mapping whereby new, unseen data can be projected to the latent space. This requires an alternative optimisation approach which in numerical experiments has not suffered from the same local minima problems.

In order to construct a feed-forward mapping from observation space to our visualization space we choose to interpolate over the data using a Radial Basis Function (RBF) network [4]. The k-th output dimension of the network takes the form:

$$\boldsymbol{y}_i^k = \sum_j \boldsymbol{W}_{jk} \phi(d(\boldsymbol{x}_i, \boldsymbol{c}_j)),$$

where $\boldsymbol{W}$ is a weight matrix, $\phi(\dots)$ is a nonlinear basis function, $d(\dots)$ is a dissimilarity measure comparing observation $\boldsymbol{x}_i$ to network centers, $\boldsymbol{c}_j$. In

this paper we consider the case where $\boldsymbol{x}_i$ is a data vector and $d(\dots)$ is the Euclidean distance, however this approach will work with any generalised observation X_i provided a dissimilarity measure d is specified. In the experiments of this paper we fix the basis functions to be thin plate splines such that $\phi(r) = r^2 \log(r)$. Typically the weights in neural networks and other interpolation models are optimised by gradient descent of a cost function, however here we can employ an alternative training approach. Denoting the matrix set of visualization vectors, $\boldsymbol{Y}$, we can compute the shadow targets [26] at each training iteration as in Neuroscale:

$$\boldsymbol{T} = \boldsymbol{Y} - \eta \frac{\partial C}{\partial \boldsymbol{Y}},$$

$$\hat{\boldsymbol{W}} = \boldsymbol{\Phi}^{\dagger} \boldsymbol{T}. \tag{7}$$

Where η is the gradient descent learning rate. It was shown in [25] that this approach has the useful properties of automatic weight regularization, a reduction of curvature and smoother optimisation when compared to alternative gradient descent methods in the NeuroScale mapping. We therefore seek to employ this optimisation procedure in Γ-SNE also and compare the empirical results with those of NeuroScale in the following section. In addition to $\boldsymbol{Y}$ the parameters a and b can also be learned in the Shadow Targets framework:

$$\hat{a} = a + \delta\eta \frac{\partial C}{\partial a},$$
$$\hat{b} = b + \delta\eta \frac{\partial C}{\partial b},$$

where δ is a smoothing parameter. This is required since the summation in $\frac{\partial C}{\partial b}$ can force the gradient contributions to be large, and the learning rate, η, which is used to learn $\boldsymbol{y}_i$ must be small for smooth gradient learning and to avoid poor minima, particularly in large datasets. We have found a suitable value for the smoothing parameter to be dependent on the Shadow Targets, namely $\delta = \sum_{ij}(\boldsymbol{T}\boldsymbol{T}^T)$. As is typical with nonlinear visualization algorithms we propose to initialise the latent variables with PCA, following which the visualization space will be centered at the origin. From this we see that our proposed δ weights a and b by the sample data covariance and that the relative scale of $\boldsymbol{T}$ will not adversely impact a and b.

The pseudocode required to compute an RBF-Γ-SNE mapping is shown in algorithm 1. This process is generic and as such the cost function, C can trivially be replaced with C^{α}, C^{β} or C^{γ} as above.

3 Results

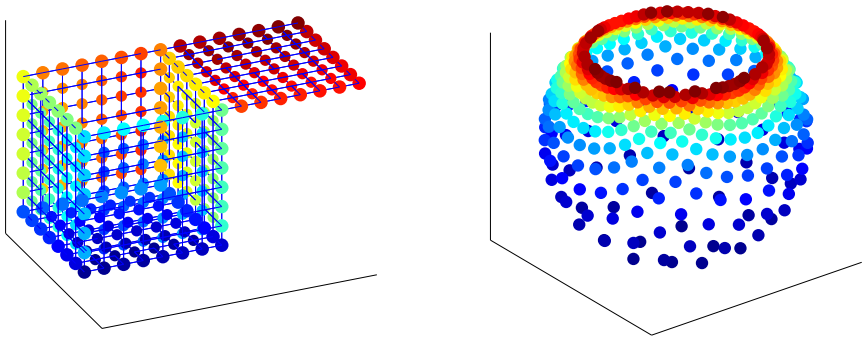
In this section we show the results of visualizations generated by Γ-SNE on four datasets. The first is the open box dataset used as a benchmark com-

Algorithm 1 Pseudocode for RBF-Γ-SNE for visualization

Require: Data $\{\boldsymbol{x}_i\}$ and perplexities σ_i, learning rate η, number of iterations, Niter

Calculate the neighbourhood probabilities, p_{ij} and basis function matrix, $\boldsymbol{\Phi}$,
Initialise latent points $\boldsymbol{Y}$ randomly or through PCA,
Initialise RBF weight matrix $\boldsymbol{W} = \boldsymbol{\Phi}^\dagger \boldsymbol{Y}$,
Calculate latent dissimilarities $d_{ij} = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|$,
Initialise Gamma distribution parameters $\{a, b\}$ through maximum likelihood of d_{ij},
Calculate latent neighbourhood probabilities, q_{ij},
Calculate initial cost, C,
for iter = 1:Niter **do**
    Calculate error gradients $\frac{\partial C}{\partial \boldsymbol{Y}}$, $\frac{\partial C}{\partial a}$, $\frac{\partial C}{\partial b}$,
    Update targets $\boldsymbol{T} = \boldsymbol{Y} - \eta \frac{\partial C}{\partial \boldsymbol{Y}}$,
    Calculate latent parameter gradient coefficient δ,
    Update RBF network weights $\boldsymbol{W} = \boldsymbol{\Phi}^\dagger \boldsymbol{T}$,
    Update parameters of the latent Gamma distribution: $\hat{a} = a + \delta\eta \frac{\partial C}{\partial a}$, $\hat{b} = b + \delta\eta \frac{\partial C}{\partial b}$,
    Calculate new latent points $\hat{\boldsymbol{Y}} = \boldsymbol{\Phi}\boldsymbol{W}$,
    Update latent dissimilarities, d_{ij},
    Update latent neighbourhood probabilities, q_{ij},
    Re-calculate cost, C,
    if cost reduced **then**
    $\boldsymbol{Y} \leftarrow \hat{\boldsymbol{Y}}, a \leftarrow \hat{a}, b \leftarrow \hat{b}$, increase η,
    else
    decrease η,
    end if
end for

parison for linear and nonlinear visualization algorithms in [17] containing an open top and uniformly sampled faces with 316 datapoints. The second is the punctured sphere dataset from the *mani* toolbox [30] used in [18] which poses the difficulties of an open top and sparse base in the 3-dimensional observation space with 500 datapoints. The third dataset used is a subset of the well known MNist dataset [16] containing 50 0's, 1's and 6's as in [8]. Finally we analyse the Caltec101 images dataset [11] following the creation of a 500-dimensional SURF bag-of-words feature descriptor vector [2] as in [21].



(a) Open box (b) Punctured sphere

Figure 2: Open box and punctured sphere datasets. These structures both contain open tops which typically pose problems for visualization algorithms. The regular spacing between points in the box sides should be preserved in the visualizations, as well as the sparse base of the sphere.

3.1 Visualizations

Figure 2 shows the open box and punctured sphere observation spaces. For each dataset we compute four mappings with Γ-SNE based on the Kullback-Leibler divergence cost function C (a, b fixed, a, b learned, RBF with a, b fixed and RBF with a, b learned), a t-SNE generated visualization for comparison and α-Γ-SNE, β-Γ-SNE and γ-Γ-SNE visualizations learned using the α-, β- and γ-divergences respectively. The fixed mappings of Γ-SNE are optimised with scaled conjugate gradients and the RBF mappings are optimised using shadow targets. Due to the use of different cost functions, C, C^{α}, C^{β} and C^{γ}, we cannot compare mapping performance through minimum cost values. In order to numerically evaluate the performance of each of the algorithms we opt to assess the Trustworthiness and Continuity of mappings, following [24], through the area under the curves over all neighbourhoods, shown in table 1.

Figures 3, 4 and 5 show the mappings for the open box, punctured sphere and MNist datasets respectively. For the open box dataset the Γ-SNE mappings using the standard cost function, C, are very similar with slight changes in the curvature of the box sides (orange and light blue). The smoothest mapping is generated by RBF Γ-SNE with a, b learned. This mapping is more accurate than the fixed counterpart due to the better optimisation of the Shadow Targets algorithm. The t-SNE mapping tears the box faces from one-another in order to preserve local neighbourhoods as do the α-, β- and γ-Γ-SNE mappings. As expected the standard Γ-SNE

mapping achieves the best neighbourhood preservation from table 1.

The latent spaces corresponding to the punctured sphere dataset shown in figure 4 appear to be the same except for where the α and β-divergences are used. Here the mapping opts to tear the structure from the sparsely sampled base as opposed to the punctured hole as expected. Despite being less visually appealing this representation produces better neighbourhood preservation than all alternative mappings.

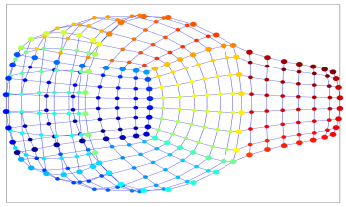
The t-SNE mapping of the MNist dataset appears to have performed the best class-separation of this dataset, but there are points which have been incorrectly removed from their local neighbourhoods in the center of the visualization space. The standard Γ-SNE mappings produce similar visualizations to α-Γ-SNE which has both the lowest class overlap and the highest level of trustworthiness. On the other hand, the visualizations learned using β- and γ-divergences have performed a latent clustering over small neighbourhoods, particularly obvious with β-Γ-SNE, whilst retaining some structure of the latent variables of curvature, angle and boldness for instance.

Figure 6 shows the visualizations of the Caltec dataset where all mappings similarly cluster the dollar bills, aeroplanes and bonsai trees mostly into distinct clusters. When the parameters of the Gamma distribution are learned the latent space is smoother than in the fixed $\{a, b\}$ and t-SNE cases creating a more intuitive representation. As for the MNist dataset, both the β- and γ-divergence mappings have created a set of micro-clusters spanning the space. For this dataset the optimal mapping in terms of neighbourhood preservation is found by Γ-SNE optimised using an RBF network and learning the distribution parameters.

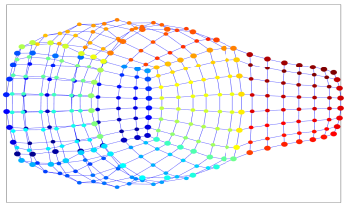
In order to demonstrate the suitability of the Gamma distribution over latent distributions, as opposed to the t-distribution, figure 7 shows the histograms for the Γ-SNE and t-SNE mappings of the Caltec dataset. Since the latent dissimilarities are Euclidean in both cases they are bounded from below by 0. The dissimilarities of the Γ-SNE mapping, with a, b learned, results in the distribution plotted as a red curve in figure 7a. This is clearly a tight fit between the probability density function and the actual learned distances. In contrast the t-SNE mapping, whose dissimilarities are shown in figure 7b, is not capable of such a tight fit with the t-distribution. A significant portion of the area under the curve is allocated to dissimilarities below 0 which will never be observed. This demonstrates why the developments in this paper are necessary.

3.2 Feed-forward properties

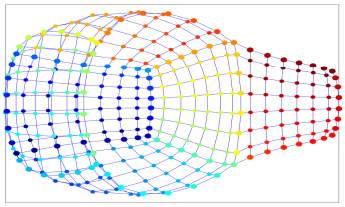
So far we have shown that the visualizations generated by Γ-SNE are superior to those of t-SNE in all four datasets. We now show some of the interesting properties found when performing the mapping with the feed-forward RBF network.



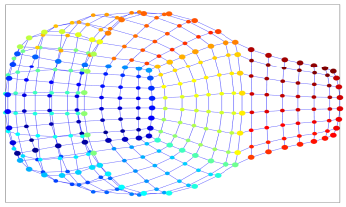
(a) Γ-SNE



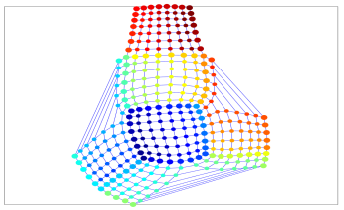
(b) Γ-SNE (a, b)



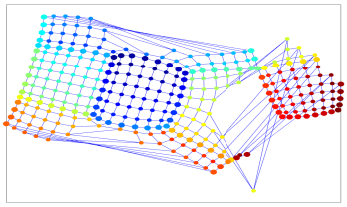
(c) RBF Γ-SNE



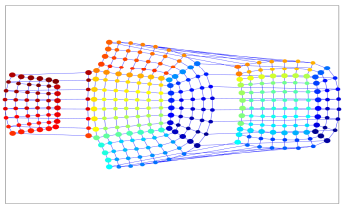
(d) RBF Γ-SNE (a, b)



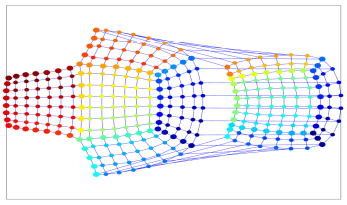
(e) t-SNE



(f) α – Γ-SNE

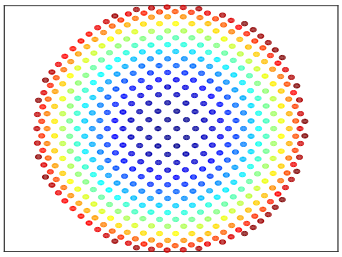


(g) β – Γ-SNE

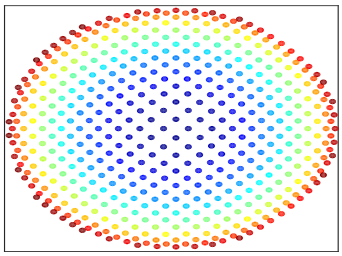


(h) γ – Γ-SNE

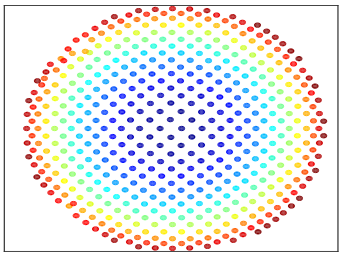
Figure 3: Open box mappings using (a) Γ-SNE with a, b fixed, (b) Γ-SNE with a, b learned, (c) RBF Γ-SNE with a, b fixed, (d) RBF Γ-SNE with a, b learned, (e) t-SNE (f) SNE.



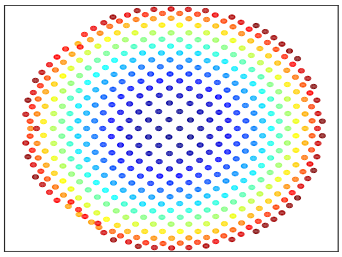
(a) Γ-SNE



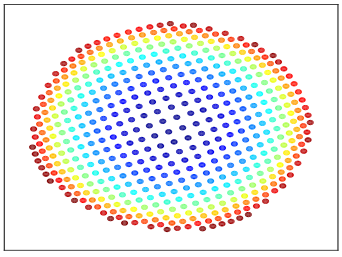
(b) Γ-SNE (a, b)



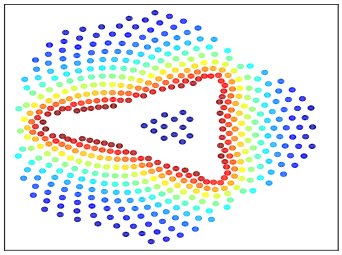
(c) RBF Γ-SNE



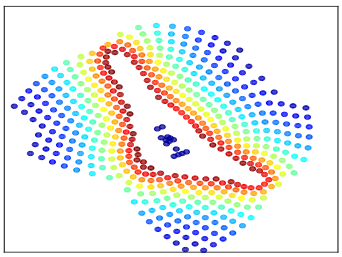
(d) RBF Γ-SNE (a, b)



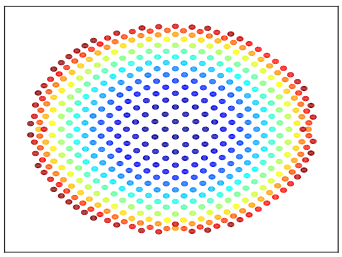
(e) t-SNE



(f) α – Γ-SNE

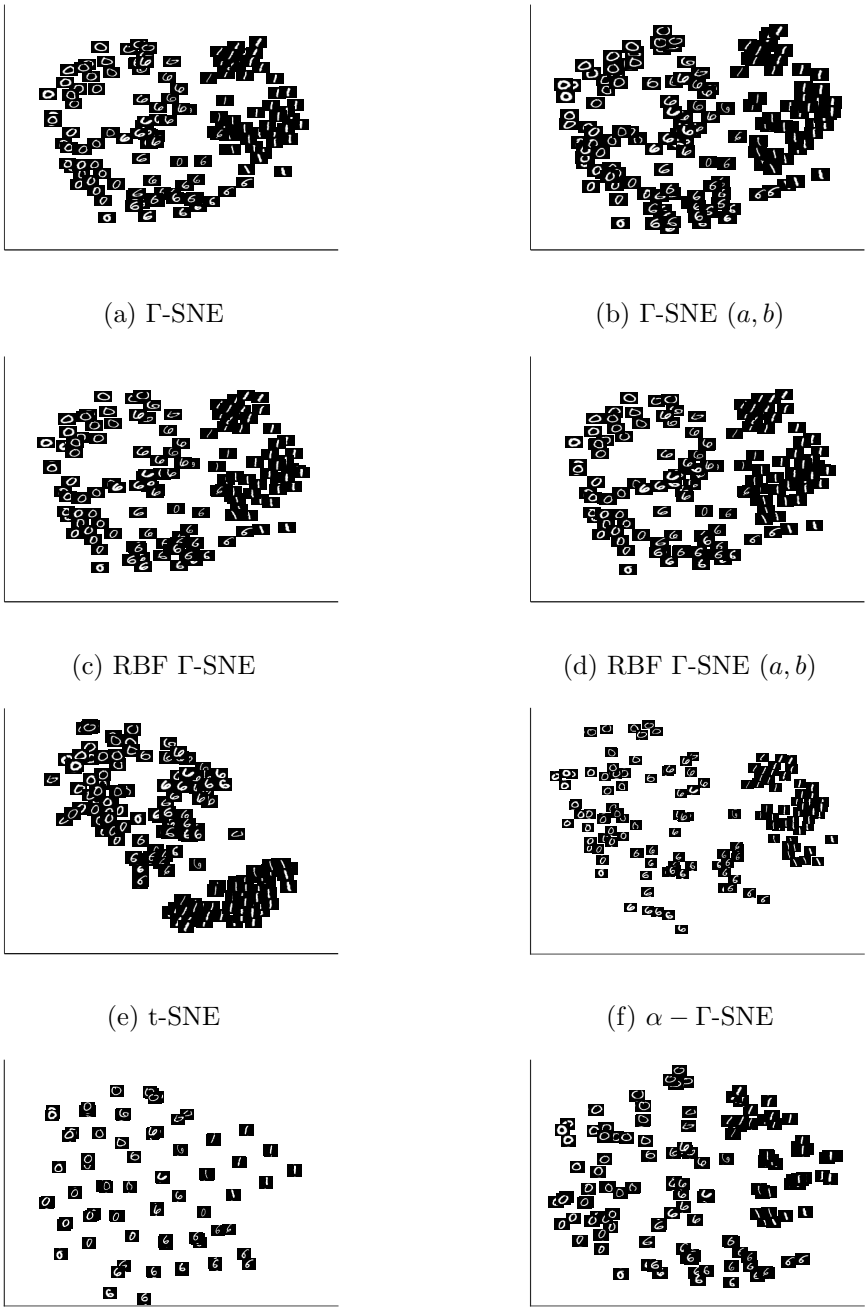


(g) β – Γ-SNE



(h) γ – Γ-SNE

Figure 4: Punctured sphere mappings using the above techniques.



(a) Γ-SNE

(b) Γ-SNE (a, b)

(c) RBF Γ-SNE

(d) RBF Γ-SNE (a, b)

(e) t-SNE

(f) $\alpha -$ Γ-SNE

(g) $\beta -$ Γ-SNE

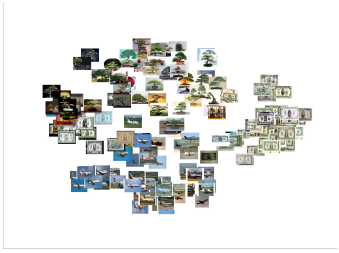
(h) $\gamma -$ Γ-SNE

Figure 5: MNist subset mappings using the above techniques where the 150 original images are centered on each latent point. Note that the 6's are correctly placed between the 0's and 1's as they share similar features to both characters.

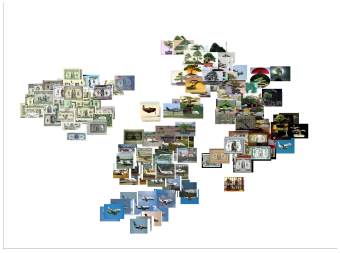
(a) Γ-SNE

(b) Γ-SNE (a, b)

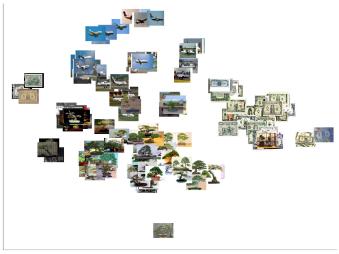
(c) RBF Γ-SNE



(d) RBF Γ-SNE (a, b)



(e) t-SNE

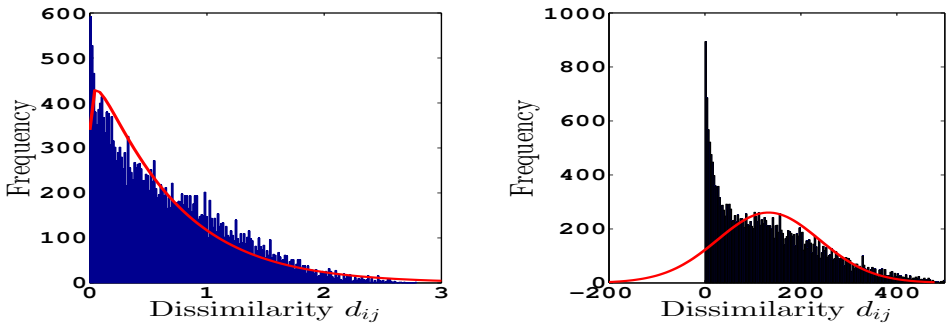


(f) $\alpha -$ Γ-SNE

(g) $\beta -$ Γ-SNE

(h) $\gamma -$ Γ-SNE

Figure 6: Caltec subset mappings using the above techniques.

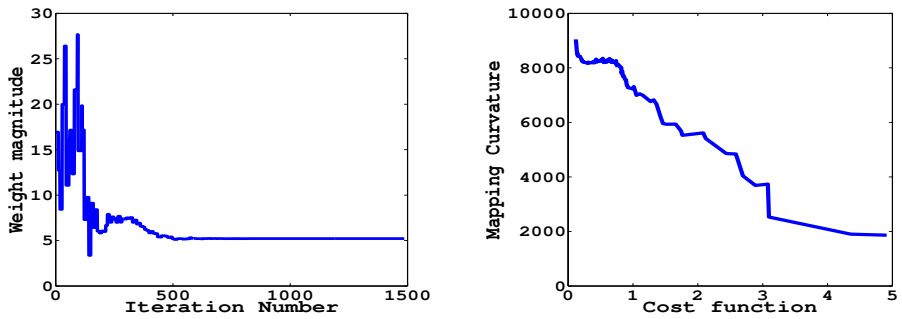


(a) (b)

Figure 7: (a) Fit of Gamma distibution from Γ-SNE and (b) t-distribution fit from t-SNE for the mappings of the Caltec dataset. The t-distribution offers a poor fit to the dissimilarities which ensures the latent distribution q_{ij} is a poor fit. This forces to optimisation of t-SNE to focus on matching local distances and not the tails, making visualizations appear clustered when the data may in fact not be.

	Box	Sphere	MNist	Caltec
Γ-SNE	**284.53**	338.69	106.59	111.66
Γ-SNE (a,b)	283.98	368.41	106.26	113.86
RBF Γ-SNE	283.19	338.54	106.30	111.95
RBF Γ-SNE (a,b)	283.05	338.57	105.82	**114.91**
t-SNE	270.98	342.89	101.83	106.79
$\alpha-$ Γ-SNE	239.29	**418.82**	**108.10**	108.13
$\beta-$ Γ-SNE	239.70	391.90	106.44	103.83
$\gamma-$ Γ-SNE	257.36	340.18	103.06	110.12

Table 1: Area under the Q_{TC} curves with higher values showing better neighbourhood preservation in the visualizations. Largest values highlighted in bold face.



(a) (b)

Figure 8: (a) Plot of weight magnitude, $\|\boldsymbol{W}\|_2$, against iteration number showing that the mapping automatically regularizes the weights during training of the MNist visualization. (b) Plot of mapping curvature against the cost function being optimised given in equation (3) for the punctured sphere dataset. We see the relative curvature increasing during training.

Firstly, as with NeuroScale the network automatically regularizes the network weights during training. Figure 8a shows a plot of $\|\boldsymbol{W}\|_2$ against iterations for the mapping of the MNist dataset illustrating the drop in weight norm. High values of this norm indicate unregularized weights resulting in overtraining such that test and unseen data will not be projected reliably. When the weights in the feed-forward mapping are learned through the shadow targets updating rule of equation (7) the weight magnitude is given by:

$$\|\boldsymbol{W}\|_2 = \frac{\|\boldsymbol{Y}\|_2}{\|\boldsymbol{\Phi}\|_2},$$

where $\boldsymbol{\Phi}$ is typically fixed based upon the observed data and as such the weight magnitude is dependent on the norm of the visualized points, $\boldsymbol{Y}$. From a PCA initialisation, as is typically performed for many nonlinear dimension reduction algorithms, the visualized points will naturally be centered at the origin, however the spread will depend on the algorithm used. Table 2 shows the norm of the visualized points, $\boldsymbol{Y}$, for each of the experiments. In all cases except for the MNist mapping the magnitude for t-SNE far exceeds that of the methods proposed in this paper. The value of learning the Γ-SNE parameters, a and b, is clear here where the weight norm is lower than for the fixed Γ-SNE mappings in all experiments. This indicates that t-SNE does not perform automatic weight regularization as Γ-SNE does, a further benefit of our approach.

Secondly, unlike NeuroScale the curvature of the mapping increases as the error decreases towards zero. This is illustrated in figure 8b which plots the curvature of the network outputs against the cost function for the punctured sphere dataset. This finding is a likely contributor for the success of

	Box	Sphere	MNist	Caltec
Γ-SNE	16,180	18,537	15,400	136
Γ-SNE (a,b)	649	614	2,944	51
RBF Γ-SNE	15,388	18,244	15,700	137
RBF Γ-SNE (a,b)	13,538	18,166	15,659	100
t-SNE	303,190	745,990	12,057	10,522
α-Γ-SNE	25,144	14,501	31,186	290
β-Γ-SNE	23,362	17,499	63,497	473
γ-Γ-SNE	24,077	13,466	45,387	352

Table 2: Norm of visualized vectors, $\|\boldsymbol{Y}\|^2$, for each of the experiments.

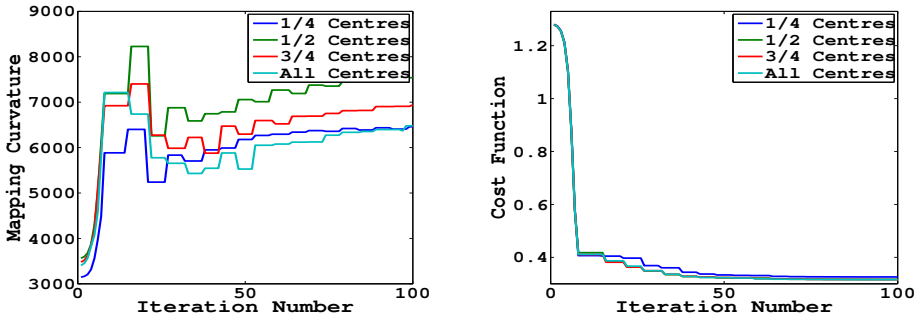
Number of centers	$\|\boldsymbol{W}\|_2$
79 (quarter)	2.6637
158 (half)	1.2671
237 (three quarters)	0.9691
316 (all)	**0.8472**

Table 3: Weight norms for networks trained with one quarter, half, three quarters and all of the data as network centers. This shows that, contrary to standard weight-based models, the weights regularize with a larger number of centers, and therefore a larger feature space.

SNE and its derivatives in unsupervised clustering tasks as it warps the input space to preserve local neighbourhood structures. This benefit over training data comes at the expense of out-of-sample neighbourhood preservation which in mappings with high curvature is not possible.

Finally, unlike typical neural network-based regression tasks the mapping improves as the number of network centers increases whilst regularizing the weights, as with NeuroScale. In order to test the interpolation ability of the RBF mapping proposed in this paper we computed the mapping for the open box dataset using a quarter, half, three quarters and all of the data as network centers. Figure 9a shows that the mapping curvature increases during training irrespective of the number of network centers. The cost function after training is equal for all configurations as shown in figure 9b, however when only one quarter of the dataset is used as centers achieving a minima requires more training iterations.

Table 3 shows the weight magnitudes for the four different configurations of the open box mapping. This shows that more network centers allows for more weight regularization as in NeuroScale.



(a) (b)

Figure 9: (a) Mapping curvature and (b) mapping error for RBF Γ-SNE with different center configurations. The curvature and cost function are relatively insensitive to the number of network centers and the curvature for all configurations increases as the cost function decreases.

4 Discussion and Future Work

This paper has introduced a variation on the popular Stochastic Neighbour Embedding algorithm. The new mapping is motivated by the desire to model neighbourhood dissimilarities using a parameterised Gamma distribution. It should be noted that the dissimilarity over inputs, $\{\boldsymbol{x}_i\}$, in this paper need not be positive-semidefinite following the successful results developed in [10], however integration of this work has not appeared widely in the literature. We have shown that using an RBF network we can optimise the parameters and visualized points of Γ-SNE using the Shadow Targets algorithm. This mapping was tested on two artifical datasets, the open box and punctured sphere, as well as the MNist and Caltec images datasets. Γ-SNE achieves results superior to t-SNE for these four cases. Further to this we have shown that the new algorithm automatically regularizes the weights and warps the space, increasing curvature whilst decreasing the mapping error. These results hold for a reduced number of network centers showing the network is capable of interpolating over the dataspace. This suggests that the mapping is incapable of overtraining in terms of weight magnitude, but the imposed curvature will prevent the mapping from being able to preserve all neighbourhoods in a large dataset, in agreement with the findings of [20]. We have also extended Γ-SNE beyond the standard Kullback-Leibler divergence cost function used as standard in SNE and t-SNE. In the experimental results we have shown that better performance is possible when α-, β- and γ-divergences are used, however the standard approach to Γ-SNE is more stable across all datasets.

This mapping requires further research into the effect of the imposed curvature in order to test whether the mapping can overtrain on data. The

analysis in this paper assumes that data is deterministic, however data is often uncertain and this method requires modification to deal with these uncertainties. One approach is to map each observed datapoint and its relative uncertainty matrix into the visualization space, as in [20] for Normally-distributed NeuroScale.

References

[1] Basu A, Harris IR, Hjort NL and Jones MC (1998) Robust and efficient estimation by minimising a density power divergence. *Biometrika* 3(85): 549–559. DOI:10.1093/biomet/85.3.549. URL http://oro.open.ac.uk/24027/.

[2] Bay H, Tuytelaars T and Gool L (2006) *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I*, chapter SURF: Speeded Up Robust Features. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 404–417.

[3] Boudjeloud-Assala L, Pinheiro P, Blansché A, Tamisier T and Otjacques B (2016) Interactive and iterative visual clustering. *Information Visualization* 15(3): 181–197. DOI:10.1177/1473871615571951. URL http://ivi.sagepub.com/content/15/3/181.abstract.

[4] Broomhead DS and Lowe D (1988) Radial basis functions, multivariable functional interpolation and adaptive networks. Technical Report 4148, RSRE.

[5] Bunte K, Biehl M and Hammer B (2012) A General Framework for Dimensionality-Reducing Data Visualization Mapping. *Neural Computation* 24(3): 771–804.

[6] Bunte K, Haase S, Biehl M and Villmann T (2012) Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing* 90: 23–45. DOI:10.1016/j.neucom.2012.02.034. URL http://dx.doi.org/10.1016/j.neucom.2012.02.034.

[7] Chernoff H (1952) A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics* 23(4): 493–507. URL http://www.jstor.org/stable/2236576.

[8] Damianou AC and Lawrence ND (2013) Deep gaussian processes. In: *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April*

29 - May 1, 2013. pp. 207–215. URL http://jmlr.org/proceedings/papers/v31/damianou13a.html.

[9] Dikmen O, Yang Z and Oja E (2015) Learning the information divergence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 37(7): 1442–1454. DOI:10.1109/TPAMI.2014.2366144.

[10] Elzbieta P and Duin RPW (2005) *The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence)*. River Edge, NJ, USA: World Scientific Publishing Co., Inc. ISBN 9812565302.

[11] Fei-Fei L, Fergus R and Perona P (2004) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision.*

[12] Fujisawa H and Eguchi S (2008) Robust parameter estimation with a small bias against heavy contamination. *Journal of Multivariate Analysis* 99(9): 2053–2081. DOI:10.1016/j.jmva.2008.02.004.

[13] Gisbrecht A, Schulz A and Hammer B (2015) Parametric nonlinear dimensionality reduction using kernel t-SNE. *Neurocomputing* 147: 71–82. DOI:10.1016/j.neucom.2013.11.045. URL http://linkinghub.elsevier.com/retrieve/pii/S0925231214007036.

[14] Gracia A, González S, Robles V, Menasalvas E and von Landesberger T (2016) New insights into the suitability of the third dimension for visualizing multivariate/multidimensional data: A study based on loss of quality quantification. *Information Visualization* 15(1): 3–30. DOI:10.1177/1473871614556393. URL http://ivi.sagepub.com/content/15/1/3.abstract.

[15] Hinton GE and Roweis ST (2002) Stochastic neighbor embedding. *Advances in neural information processing systems* : 833–840DOI: http://books.nips.cc/papers/files/nips15/AA45.pdf.

[16] LeCun Y, Cortes C and Burges CJC (1998) The mnist database. Http://yann.lecun.com/exdb/mnist/.

[17] Lee JA and Verleysen M (2007) *Nonlinear Dimensionality Reduction*. 1st edition. Springer Publishing Company, Incorporated. ISBN 0387393501, 9780387393506.

[18] Lin T, Zha H and Lee S (2006) Riemannian manifold learning for nonlinear dimensionality reduction. In: *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06. Berlin,

Heidelberg: Springer-Verlag. ISBN 3-540-33832-2, 978-3-540-33832-1, pp. 44–55.

[19] Lowe D and Tipping ME (1997) Neuroscale: Novel topographic feature extraction using rbf networks. In: Mozer M, Jordan M and Petsche T (eds.) *Advances in Neural Information Processing Systems 9*. MIT Press, pp. 543–549.

[20] Rice I (2015) *Probabilistic Topographic Information Visualisation*. PhD Thesis, Aston University.

[21] Rice I (2016) Improved data visualisation through multiple dissimilarity modelling. *Information Sciences* 370-371: 288–302.

[22] Roweis ST and Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290: 2323–2326.

[23] Sarlin P (2015) Data and dimension reduction for visual financial performance analysis. *Information Visualization* 14(2): 148–167. DOI:10.1177/1473871613504102. URL http://ivi.sagepub.com/content/14/2/148.abstract.

[24] Sun J, Crowe M and Fyfe C (2013) Incorporating visualisation quality measures to curvilinear component analysis. *Information Sciences* 223: 75 – 101. DOI:http://dx.doi.org/10.1016/j.ins.2012.09.047. URL http://www.sciencedirect.com/science/article/pii/S0020025512006408.

[25] Tipping ME (1996) *Topographic mappings and feed-forward neural networks*. PhD Thesis, Aston University, Aston Street, Birmingham B4 7ET, UK.

[26] Tipping ME and Lowe D (1997) Shadow targets: A novel algorithm for topographic projections by radial basis functions. *NeuroComputing* 19: 211–222.

[27] van der Maaten L (2009) Learning a parametric embedding by preserving local structure. In: Dyk DAV and Welling M (eds.) *AISTATS*, *JMLR Proceedings*, volume 5. JMLR, pp. 384–391.

[28] van der Maaten L, Postma E and van den Herik J (2009) Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research* 10: 1–41. DOI:10.1080/13506280444000102.

[29] van der Maaten LJP and Hinton GE (2008) Visualizing high-dimensional data using t-sne. *JLMR* 9: 2579–2605.

[30] Wittman T (2005) Mani fold learning matlab demo. Http://www.math.ucla.edu/ wittman/mani/.