Confidence estimation for t-SNE embeddings using random forest

Busra Ozgode Yigin · Gorkem Saygili

Received: 16 November 2021 / Accepted: 17 August 2022 / Published online: 10 September 2022
© The Author(s) 2022

Abstract
Dimensionality reduction algorithms are commonly used for reducing the dimension of multi-dimensional data to visualize them on a standard display. Although many dimensionality reduction algorithms such as the t-distributed Stochastic Neighborhood Embedding aim to preserve close neighborhoods in low-dimensional space, they might not accomplish that for every sample of the data and eventually produce erroneous representations. In this study, we developed a supervised confidence estimation algorithm for detecting erroneous samples in embeddings. Our algorithm generates a confidence score for each sample in an embedding based on a distance-oriented score and a random forest regressor. We evaluate its performance on both intra- and inter-domain data and compare it with the neighborhood preservation ratio as our baseline. Our results showed that the resulting confidence score provides distinctive information about the correctness of any sample in an embedding compared to the baseline. The source code is available at https://github.com/gsaygili/dimred.

Keywords T-SNE · Confidence score · Embedding · Dimensionality reduction · Random forest

1 Introduction

There has been an upsurge in size, heterogeneity, complexity, and dimensionality of generated data as a result of developments in information technologies, data collection tools, and techniques. Visualization of data is not possible on an ordinary display considering the high dimensionality. Various dimensionality reduction (DR) algorithms have been developed in order to visualize high dimensional data, make them suitable for analysis, and produce simpler and more meaningful representations without losing the intrinsic pattern of the data. DR methods have continued to evolve, often independently in multiple fields including for genome-wide gene expression [1], single-cell RNA sequencing [2], genomics [3, 4], natural language processing [5, 6], computer security [7], and biomedical signal processing [8].

DR is a type of unsupervised machine learning that can be applied as a preprocessing step before clustering as well as solely for visualization. DR algorithms can be classified into two groups: linear and non-linear. Linear techniques such as Principal Component Analysis (PCA) [9], Singular Value Decomposition (SVD) [10], Linear Discriminant Analysis (LDA) [11] use linear functions to transform high dimensional data into lower dimension. Non-linear techniques such as Kernel PCA [12], Isomap [13], and t-Distributed Stochastic Neighbor Embedding (t-SNE) [14] are used for data having complex non-linear structures and intend to preserve non-linear geometric structures [15].

Currently, many studies comparing various DR algorithms are being addressed in the literature [15–18]. Recently, t-SNE has become one of the most popular DR algorithms especially for the visualization of image data containing many examples, such as MNIST [19], and genomics data such as Allen Brain Atlas Mouse Brain dataset [20]. Additionally, it is utilized to achieving better classification performance with the reduced dimensional space [21]. Hence, we use t-SNE in our experiments. While non-linear DR methods like t-SNE provide visual insight into the underlying pattern of high-dimensional data, they do not provide an explicit parametric mapping between the high-dimensional data space and its representation in the low-dimensional space (embedding). Hence, they are not
reconstructable which makes error detection a much harder task. It is critical to mitigate erroneous samples in an embedding especially while visualizing medical or genetic data.

In the literature, there are many studies aiming to generate confidence and detect errors for various domains such as medical image registration [22–25] and stereo matching [26–28]. Recently, DR becomes also the focus of error estimation research [29]. Evaluating and comparing the embeddings are typically done qualitatively, by placing projections side by side and letting human judgment to determine which projection is the best. There is also a study that proposes a quantitative way to evaluate embedding results similar to human perception [30]. Additionally, several techniques have been developed to detect erroneous samples, most of which rely on retaining local neighborhood rankings [18, 31, 32]. Ranking-based metrics focus on maintaining the neighborhood for each sample in both high dimensional space and embedding. Any change in the neighborhood, either an intrusion or extrusion of samples, is indicated as an error regardless of the class label of the extruder or intruder. However, we advocate that any change in the order of the neighborhood shouldn’t be considered as an error as long as the neighborhoods in the embedding consist of the samples that belong to the same class. On the contrary, if we visualize a sample with a different label than the majority of the samples in a cluster, we consider that sample as an error. Hence, the definition of error in an embedding should be related to class labels, not the order of local neighborhoods. To find a confidence score that is correlated with class-based error, we propose a supervised confidence estimation method with distance-based features extracted using neighborhoods from high- and low-dimensional spaces.

We outline here, the main contributions of our proposed approach:

1. Our algorithm generates confidence scores for embeddings that can distinguish erroneously embedded samples from the correctly embedded ones.
2. The proposed confidence estimation approach can be used on datasets from different domains (domain-independent).
3. The confidence score can be generated for each sample in the embedding that may lead to the removal of erroneous samples in the embedding to have more reliable representations of the data.

In our study, we make a comparison between the $K$ neighbors in the high-dimensional space and those in the embedding for extracting the features using different similarity measures on the high-dimensional space. To mitigate the effect of intrusions and extrusions both distances from the neighbors in high- and low-dimensional spaces, distances are sorted in an increasing order. Then, we use these features to train our random forest regression algorithm to predict a confidence score that indicates how trustworthy a sample is in the embedding. To our knowledge, we are the first to propose such a confidence score in particular for non-linear dimensionality reduction algorithms such as t-SNE.

We structured the rest of this paper as follows: Sect. 2 contains a description of t-SNE, definitions of correct and erroneous samples, confidence and NPR scores and the formulation of our proposed confidence estimation algorithm, followed by an experimental setup and dataset description. Experimental results on seven different datasets from two different domains (image and gene expression) are presented in Sect. 3. We elaborate our results in Sect. 4 and draw our conclusions in Sect. 5.

2 Methodology and experimental setup

Our proposed confidence estimation algorithm consists of feature extraction using different distance measures on high-dimensional data and training a random forest regression algorithm to predict a confidence score that indicates how trustworthy a sample is in the embedding. All the sub-steps in our algorithm are described in the following sections in detail.

2.1 t-Distributed stochastic neighbor embedding (t-SNE)

t-SNE is an unsupervised machine learning algorithm for visualization developed by Laurens van der Maaten and Geoffrey Hinton [14]. t-SNE is a non-linear dimensionality reduction technique to keep close neighborhoods in the high-dimensional space also close together in low-dimensional space. The t-SNE algorithm calculates a similarity measure between pairs of instances in the high- and low-dimensional space. It converts the high-dimensional Euclidean distances between data points $x_i$ and $x_j$ into conditional probabilities $p_{ij}$ as:

$$
p_{ij} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k\neq i} \exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{1}
$$

where $\sigma_i$ is the variance of the Gaussian that is centered on data point $x_i$. It preserves the local structure of the data using heavy-tailed Student-t distribution with a single degree of freedom to compute the similarity between two points in low-dimensional space rather than a Gaussian distribution, which helps to address the crowding and optimization problems. $y_i$ and $y_j$ are the low-dimensional counterparts of the high-dimensional data points $x_i$ and $x_j$. This gives a second set of probabilities ($q_{ij}$) in the low-dimensional space:

$$
q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k\neq j}(1 + ||y_k - y_j||^2)^{-1}} \tag{2}
$$
Then t-SNE optimize these two joint probabilities using the sum of Kullback-Leibler cost function and a gradient descent optimization method. The cost function $C$ is given by:

$$C = \sum_i KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$  \hspace{1cm} (3)

We applied t-SNE to obtain embeddings using the manifold package of the Sklearn library. t-SNE has a parameter called perplexity which is a hyperparameter to be set by the user and can affect the resulting embeddings substantially. The default value for perplexity is 30 in the original implementation [14] and we also use this value in all of our experiments.

### 2.2 Definition of the erroneous and correct samples

To label samples in an embedding as erroneous or correct, we check whether each sample has the same label as the majority of its $K$ nearest neighbors. Algorithm for finding the erroneous samples according to majority of the closest neighbors is given in Algorithm 1. First, Euclidean distances are calculated and sorted to determine the $K$ closest neighbors in the embeddings. Then, we count the number of the closest neighbors with the same label as the sample and indicate the sample as correctly embedded if the count is not lower than $K/2$. Hence, for a neighborhood of 20 samples, a sample is considered as a correctly embedded sample (1) if at least 10 of its neighbors have the same class label as the sample otherwise it is considered as an erroneous sample (0). We plot the t-SNE embeddings of our datasets by indicating erroneous samples with red circles in Fig. 1.

### 2.3 Extraction of the distance-based features

In our confidence estimation algorithm, we obtain our features by calculating different distance measures in high-dimensional space, considering the $K$ neighborhood of each sample in high- and low-dimensional spaces. Studies investigating the effect of different distance measures have proven that no distance metric is optimal for all kinds of data from different domains [33, 34]. Each data type may require a different distance measurement for optimal performance in line with the “no free lunch” principle. As a result, the most extensively used distance measures ($\ell^p$) namely Euclidean, cosine, correlation, Chebyshev, Canberra, and Braycurtis were utilized to extract features. The features are extracted in a neighborhood of $K$ around each sample. In our experiments, we choose $K$ as 20. It is worth noting that we use the Euclidean distance to obtain the 20 closest samples in high-dimensional ($N_D$) and low-dimensional ($N_d$) spaces. Furthermore, after determining the closest neighbors in both spaces, all distance measurements are calculated only in high dimensional space. Since different distance measures yield values in different ranges, normalizing these distances, as in Eq. 4 is an important step before feeding them into the machine learning algorithm. Normalization prevents inducing possible bias that is caused by large scale differences between features into the machine learning algorithm and generally makes the training faster.

Let $D_p(x_i, x_j)$ be the $\ell^p$ distance between two samples $x_i$ and $x_j$ in high-dimensional space. We normalize distances for each of the six distance measures as:

```python
Algorithm 1 Algorithm for finding the erroneous samples according to majority of closest neighbors

1: function FIND ERRORS MAJORITY($X_{emb}$, labels, $K = 20$
2:     $X_{d} = $ Euclidean_Distance ($X_{emb}$)
3:     sort_index = argsort ($X_{d}$)
4:     error_list = []
5:     for i ← 0 to $X_{emb}.shape$ [0] do
6:         K_neigh = sort_index [i]
7:         lab_neigh = labels [K_neigh]
8:         num-label = count_nonzero (lab_neigh == labels[i])
9:         if num-label < $K/2$ then
10:            error_list.append(i)
11:        end if
12:     end for
13:     err = array(error_list)
14:     return err
15: end function
```
Fig. 1 t-SNE embeddings and erroneous samples of seven datasets: a Test set of 10000 samples from the MNIST dataset [19], b Test set of 10000 samples from the Fashion-MNIST dataset [39], c Test set of 2555 samples from Allen Mouse Brain (AMB18) dataset [40], d Simulated dataset of 8839 samples [40], e Test set of 1714 samples from Brain Human dataset [41], f Muraro dataset of 2122 samples [41], g Segerstolpe dataset of 2133 samples [41]. The erroneous samples are indicated with red circles. The total number of erroneous samples of each dataset is given on top of each dataset.
Since we are not concerned with the changes in the order of the neighbors, the calculated distances are sorted in ascending order as follows:

\[ N = (x_i, k) \text{ and } N_d = (x_i, k) \]

denote the \( k \)th neighbor of \( x_i \) in the high- and low-dimensional spaces. Features are calculated as:

\[
\hat{D}_e(x_i, x_j) = \frac{D_e(x_i, x_j) - \min(D_e(x_i, x_j))}{\max(D_e(x_i, x_j)) - \min(D_e(x_i, x_j))}
\]  

(4)

Since we are not concerned with the changes in the order of the neighbors, the calculated distances are sorted in ascending order as follows:

\[ S(x) = \{x_0 < x_1 < \ldots < x_n\} \]

(5)

Let \( N_p(x_i, k) \) and \( N_d(x_i, k) \) denote the \( k \)th neighbor of \( x_i \) in the high- and low-dimensional spaces. Features are calculated as:

\[
F_e(x_i, k) = |S(\hat{D}_e(x_i, N_p(x_i, k))) - S(\hat{D}_e(x_i, N_d(x_i, k)))|
\]

(6)

Figure 2 depicts the normalized Euclidean distances in high-dimensional space for the 20 neighbors in both high- and low-dimensional spaces of two samples in the embedding of the MNIST dataset. The sample selected and marked with the green circle is correctly located in a neighborhood with similar samples of the same label, while the sample in the red circle is erroneously placed. Graphs in Fig. 2 illustrate their corresponding Euclidean distances of 20 closest neighbors in higher and lower dimensions. Euclidean distances for neighbors from both spaces largely overlap for the correctly embedded sample, while there is no overlap for the erroneously placed sample, indicating the amount of error. We exploit these differences to predict the confidence of each sample in an embedding using a random forest regressor.

2.4 Definition of the confidence score

Ground truth confidence scores are obtained by calculating the number of neighboring samples that have the same label with the sample in the embedding and we normalize this number by dividing it by 20 (as in Algorithm 2). We utilize the same approach which we applied for detecting erroneous samples to determine the nearest neighbors and compare the labels of the current sample with the closest neighbors. Instead of labeling the samples as erroneous or correct, we assigned a confidence score to reflect how correct they are. Hence, our target scores range between 0 and 1, denoting lowest confidence as 0 and highest possible confidence as 1.
2.5 Confidence estimation algorithm

After extracting the distance-based features, we use them as input of a Random Forest (RF) regressor in which target labels are the ground truth confidence scores from previous section. RF is one of the most popular machine learning algorithms because it produces reliable results without requiring extensive hyper-parameter adjustment and can be used to solve both regression and classification problems. RF is a decision tree-based ensemble learner that fits a number of decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and prevent overfitting.

We train our RF regressor on the training sets of MNIST, AMB18 and Baron Human datasets separately and evaluate our algorithm on the test set of the same dataset (intra-dataset) and on the test sets of other datasets (inter-dataset). In order to optimize hyperparameters of our regressor, we incorporate the grid search method with 3-fold cross-validation and use the best parameters for the number of estimators (20, 50, 100, 200), maximum features (auto, sqrt), maximum depth (2, 5, 10, 20), and minimum samples per split (2, 5, 10, 20) in our tests, leaving all other parameters as default. Although we find optimum hyperparameters for each experiment, a hyperparameter setting of 10 for both maximum depth and minimum samples split, 200 for the number of estimators and ‘auto’ for the maximum features generally provides the best results.

2.6 Neighborhood preservation ratio

Neighborhood Preservation Ratio (NPR) has been used as a measure of correctness in embeddings and to quantitatively assess the performance of different dimensionality reduction algorithms [35–37]. There are different ways to calculate NPR, for instance Shen et al. [38] evaluated an NPR score that gives a general quality score for the whole embedding. In contrast, Maaten et al. [35] calculated an NPR score per sample in the embedding which uses the ratio of preserved neighbors of each sample in both high- and low-dimensional spaces. In this work, the NPR score of the latter is implemented as the baseline. The NPR assesses how well the distances in the high dimensional space are retained in the low-dimensional space based on the Euclidean distances. Consequently, it computes the intersection amount of the nearest K neighbors in high-dimensional and embedding space. We choose the k lowest Euclidean distance in the high-dimensional space \( N_D(x_i, 1 : k + 1) \) and the k lowest Euclidean distance in the embedding space \( N_d(x_i, 1 : k + 1) \) for each point \( i \). The NPR is the ratio of conserved neighborhood numbers:

\[
\text{NPR}(x_i) = \frac{N_D(x_i, 1 : k + 1) \cap N_d(x_i, 1 : k + 1)}{k}
\]

where \( N_D(x_i, 1 : k + 1) \cap N_d(x_i, 1 : k + 1) \) calculate the number of co-existing points in high- and low-dimensional spaces and \( k \) is the selected number of closest neighbors.

\[
\text{Algorithm 2} \text{ Algorithm for finding the confidence score according to the matching of the labels with closest neighbors}
1: \text{function} \ FIND \ CONFIDENCE \ SCORE(X_{\text{emb}}, \text{labels}, K = 20) \\
2: \hspace{10pt} X_d = \text{Euclidean Distance} (X_{\text{emb}}) \\
3: \hspace{10pt} \text{sort}.\text{index} = \text{argsort} (X_d) \\
4: \hspace{10pt} \text{Score} = [ ] \\
5: \hspace{10pt} \text{for} \ i \leftarrow 0 \ \text{to} \ X_{\text{emb}}.\text{shape}[0] \ \text{do} \\
6: \hspace{20pt} K\text{.neigh} = \text{sort}.\text{index}[i] \\
7: \hspace{20pt} \text{lab}.\text{neigh} = \text{labels}[K\text{.neigh}] \\
8: \hspace{20pt} \text{num}.\text{label} = \text{count}.\text{nonzero} (\text{lab}.\text{neigh} == \text{labels}[i]) \\
9: \hspace{20pt} \text{score} = \text{num}.\text{label}/K \\
10: \hspace{20pt} \text{Score}[i] = \text{score} \\
11: \hspace{10pt} \text{end} \ \text{for} \\
12: \hspace{10pt} \text{return} \ \text{Score} \\
13: \text{end \ function}
\]
Fig. 3 NPR and Confidence estimation results with box-plots for seven test sets. The NPR scores distribution for the correct (1) and erroneous (0) classes is shown in the first two boxes on each chart. The following boxes give the distribution of confidence scores produced by the RF regressor trained in AMB18, MNIST and Baron Human datasets for the correct (1) and erroneous (0) classes, respectively.
We use the NPR score as our baseline to compare the performance of proposed confidence estimation algorithm. As experimental results, we present box plots of NPR and RF results for correct and erroneous samples in the embedding for three different training sets (AMB18, MNIST and Baron Human) in Fig. 3. Additionally, we present the number of erroneous samples in the 100, 50, and 10 samples with the lowest NPR and RF scores.

### 2.7 Datasets

In our experiments, we employed seven different publicly available datasets from two different domains namely images and single cell RNA-seq. Two of those datasets consist of images and are widely used for supervised learning. The other five are single-cell RNA-seq from four real and simulated datasets.

The first dataset is MNIST [19] which consists of 60000 for training and 10000 for test images with sizes of $28 \times 28$, from 10 classes of handwritten digits. We create 120 embeddings consisting of 500 randomly selected non-overlapping samples from the MNIST training set and we use those embeddings for training. The reason for creating subsets is to obtain a higher number of erroneous samples in the embeddings which is crucial to have a more balanced training set. Test set of 10000 samples is used for testing our algorithm. The second image dataset is Fashion-MNIST [39] which consists of $28 \times 28$ gray scale images of 70000 fashion products divided into 10 categories, each with 7000 images. The training set has 60000 images, whereas the test set contains 10000. The Fashion-MNIST has the same image size, data format, and train and test split structure as the original MNIST. We use only the test set of this dataset for the evaluation of our model.

The next two datasets are a real and simulated single-cell RNA-seq datasets reported by [40]. AMB18 is the Allen Mouse Brain dataset from the primary visual cortex of the mouse brain used by Michielsen et al. [40]. AMB18 dataset consists of 15 different sub-populations, 12, 771 cells and 2000 genes for each cell. We use the sub-populations as labels. We split 20% of the dataset for testing and the rest for training. The simulated dataset contains 8839 cells, 9000 genes, and six different cell populations. We only utilize this set as a test set.

All other scRNA-seq datasets, Baron Human, Segerstolpe and Muraro are human pancreas datasets obtained from a benchmark study published by Abdelaal et al. [41] on Zenodo [42]. Baron Human consists of 14 different cell types, 8569 cells and 17499 genes for each cell generated by InDrop [43]. Segerstolpe dataset consists of 2133 cells 22757 genes of 14 different cell types generated by Smart-Seq2 [44]. Muraro dataset consists of 9 different cell types, 2122 cells and 18915 genes for each cell generated by CEL-Seq2 [45].

### 3 Results

Table 1 and Fig. 3 present the experimental results that we obtain by training RF regressor on training sets of the MNIST, AMB18 and Baron Human datasets. The tests are performed on test sets of seven datasets for NPR and our confidence score algorithm.

The number of erroneous samples in the lowest 100, 50, and 10 NPR and RF scored samples is calculated from each dataset and are shown in Table 1. It can be clearly seen that our algorithm reveals more erroneous samples in almost all datasets. The only two exceptions are the lowest 10 erroneous samples for AMB18 and Fashion MNIST datasets. Our confidence estimation algorithm produces slightly better results in intra-domain than inter-domain datasets. But even in the inter-domain datasets, it is able to detect substantial numbers of erroneous samples. For instance, our model trained on AMB18 is able to detect 73 erroneous samples for 100 lowest scored samples in Fashion-MNIST. The relatively low number of successfully-predicted erroneous samples in AMB18 and Simulated datasets is due to the low number of erroneous samples in these datasets. 40 out of 2555 samples for AMB18 is erroneous, whereas 125 out of 8839 samples are erroneous for the Simulated dataset. In contrast to these two, 601 out of 10000 and 2401 out of 10,000 samples are erroneous for the MNIST and Fashion-MNIST datasets, respectively. Furthermore, the inter-dataset results on Baron Human dataset shows that our algorithm trained on MNIST and Baron Human datasets find the same number of erroneous samples, 36, in the last 100 samples whereas NPR only finds 16 which is less than half of our confidence score.

In Fig. 3 each graph represents the box plots of NPR scores for each class of erroneous and correctly embedded samples, as well as the corresponding confidence scores generated from the RF algorithm on three different training sets, AMB18, MNIST, and Baron Human, respectively. Our algorithm generally achieves the most successful distinction between erroneous and correct samples than NPR as seen from the box plots for all datasets. For all intra-dataset experiments, the erroneous and correct samples are clearly separable with our confidence score whereas they are not separable with the NPR. This difference is specifically clear on the Baron Human intra-dataset experiment. Based on more challenging inter-dataset experiments, the erroneous and correct samples are clearly separable using our confidence score for the Segerstolpe dataset whereas they are not separable with the NPR since the two boxes of correct
and erroneous samples are highly overlapping. Similarly, the distinction is also visible on the Simulated dataset.

The number of erroneous samples for the test set of MNIST is 601 out of 10000 samples as depicted in Fig. 4a.

We eliminate erroneous samples using our confidence score with a threshold of 0.5 which is almost equal to eliminating low-scored 2500 samples from the MNIST as depicted in Fig. 4b. To compare their results, in line

Table 1 The number of erroneous samples in the 100, 50, and 10 lowest NPR and RF scored samples. RF scores are obtained from training on MNIST, AMB18 and Baron Human datasets and tested on all test sets. Bold numbers indicate the best results for each test set.

|                | RF Results Trained on MNIST | RF Results Trained on AMB18 | RF Results Trained on Baron Human | NPR Results |
|----------------|-----------------------------|-----------------------------|----------------------------------|-------------|
|                | 100  | 50  | 10  | 100  | 50  | 10  | 100  | 50  | 10  | 100  | 50  | 10  |
| AMB18          | 9    | 6   | 1   | 13   | 8   | 0   | 9    | 4   | 2   | 5    | 4   | 2   |
| Simulated      | 14   | 8   | 3   | 26   | 15  | 5   | 16   | 9   | 3   | 11   | 4   | 1   |
| MNIST          | **81** | **44** | **10** | **69** | **36** | **8** | **67** | **36** | **78** | **59** | **31** | **5** |
| Fashion-MNIST  | 50   | 28  | 6   | **73** | **42** | **8** | **67** | **35** | **7** | **57** | **32** | **9** |
| Baron Human    | 36   | 19  | 7   | 21   | 15  | 4   | **36** | **27** | **8** | **16** | **9** | **3** |
| Muraro         | 31   | 22  | 6   | 27   | 15  | 4   | 22   | 17  | 8   | 17   | 13  | 7   |
| Segerstolpe    | 37   | 25  | 7   | 37   | 25  | 6   | **46** | **30** | **7** | **29** | **19** | **6** |
with the confidence score, we also eliminated 2500 lowest NPR-scored samples from the MNIST as depicted in Fig. 4c. With the elimination according to the confidence score, the number of erroneous samples is reduced from 601 over 10000 samples (6%) to 163 (2.15%) over 7500 samples. On the other hand, the number of erroneous samples is reduced from 6% to 2.84% over 7500 samples by eliminating according to the NPR score. In Fig. 4d, it is seen that the borderline samples generally have lower confidence than the center samples.

4 Discussion

t-SNE has been a widely-used DR algorithm to visualize and explore high-dimensional data and sometimes it is used together with another machine learning algorithm to increase its classification performance. Especially in cancer research and gene analysis studies [1, 3, 4, 36, 40], being able to detect and remove the erroneous samples from an embedding may provide better analysis of the high-dimensional data. Although ranking-based algorithms [18, 31, 32] have been developed to evaluate the embedding success of t-SNE, inferring results from the rank may lead to false insights since the aim of t-SNE is not to preserve the ranking but to preserve the distribution of group of samples in both high- and low-dimensional spaces. Hence, we argue that considering intrusions and extrusions due to the ranking may be misleading in terms of evaluating the quality of an embedding. Furthermore, these algorithms are in general computationally costly for large datasets. As an alternative to ranking-based algorithms, neighborhood preservation-based techniques have been introduced. These techniques focus only to preserve the closest neighborhoods independent from the order. However, when we examine an embedding perceptually, what we consider as an erroneous sample is the one that does not belong to the class of the majority of its neighbors. In a previous study [29], an error detection algorithm based on classification was presented for dimensionality reduction. We advocate that a binary classifier would be inferior than a regressor, since there is no threshold value suitable for every dataset. Hence, the results with a constant threshold value may not demonstrate adequate success for every dataset. For example, if we consider box plots in Fig. 3, the correct and erroneous samples can be separated with a threshold value close to 0.5 after training on the MNIST, whereas the threshold should be 0.9 after training on the AMB18. As a result, a regression algorithm that can generate a confidence score rather than classification would be more useful.

In this study, we propose a confidence estimation approach via exploiting six different distance measures as features and training a random forest regressor to produce confidence scores per sample. In Fig. 2, we present the difference between a correctly and wrongly embedded sample using Euclidean distance of 20 closest neighbors in high- and low-dimensional spaces. For a correctly placed example, it can be understood from the overlap in the graph that the distances of the neighbors in both spaces are generally preserved. On the other hand, the distances in high dimensional space are entirely not preserved for the incorrectly placed example. Our algorithm exploits these differences using six different distance measures to generate a confidence score.

To compare the performance of our algorithm, we utilize the NPR score which is frequently used in estimating the quality of an embedding [35]. In Fig. 3, although there are some outliers, the scores produced by our regressor trained on our training sets differ more distinctively from each other for correct and erroneous samples on the all test sets than NPR, except for the Fashion MNIST and the Baron Human. We argue that this performance drop is due to the considerably-higher number of erroneous samples in the embedding of Fashion-MNIST and considerably-lower number of erroneous samples in the embedding of Baron Human. The NPR score, on the other hand, cannot clearly distinguish between correct and erroneous samples on all datasets. The fact that our algorithm’s performance between datasets (inter-dataset) yield comparable results indicates that our method can perform well regardless of which dataset or domain it has been trained on.

Our models trained on three datasets detect higher number of errors in the 100, 50, and 10 lowest-scored samples than those for NPR as presented in Table 1. Although our algorithm produces slightly better results in intra- than inter-domain datasets, this may also be due to the difference in the ratio of erroneous samples in the embeddings.

Our confidence score eliminates erroneous samples from embeddings substantially as depicted in Fig. 4b. As it can be seen from Fig. 4d, the borderline samples tend to have low confidence scores which can be more deceptive when determining cluster centers. The elimination of erroneous samples produces much clear and reliable embeddings. Besides detecting erroneous samples in an embedding, we foresee that our algorithm may provide feedback to DR algorithms to produce embeddings with a fewer number of erroneous samples. For example, in addition to Kullback–Leibler (KL) divergence, which t-SNE employs to evaluate the dissimilarity of distributions, our confidence score can be incorporated as a regularizer. As a result, it can contribute to the creation of better clusters. On the other hand, removing erroneous samples based on their confidence score may allow more accurate computation of cluster centers, especially when DR
methods are utilized as a preprocessing step before the clustering algorithm. Hence, we expect to have better clustering results if our confidence scores can be used in an adaptive clustering algorithm.

Although experimental findings on our datasets indicate a reasonably good error estimation performance compared to NPR, the overall performance depends on the quality of the overall embedding and the number of erroneous samples. For example, large number of errors in the Fashion-MNIST test set makes it difficult for the algorithm to distinguish between erroneous and correct samples since when incorrect samples gather together by chance, they negatively affect each other’s confidence score. On the other hand, having very few erroneous samples, as in the Baron Human dataset, makes the algorithm produce reasonably-high confidence scores for all samples including the erroneous ones.

5 Conclusion and future work

This paper describes a novel method for predicting the confidence score of the samples in an embedding. Our algorithm provides more reliable estimation of sample-wise quality of an embedding than the NPR score. Seven datasets from two different domains have been used to show the feasibility and efficiency of our algorithm. Experimental results confirm that our algorithm performs well in both intra- and inter-dataset experiments regardless of the domain of data. The results of our experiments illustrate that our algorithm would lead to better judgment about the distribution of the samples inside the data and can be used to build higher confidence clusters.

The performance of the proposed algorithm can be further enhanced using deep neural network-based regression models such as 1-D Convolutional Neural Networks or Long-Short Term Memory networks. We plan to utilize these algorithms as the next step to improve the performance.

In this paper, we use only t-SNE in our experiments since it is one of the most common non-linear dimensionality reduction algorithms. However, we do not expect an algorithm-related bias in our results since we calculate distance differences of high and low-dimensional neighbors using common distances as features without using any dimensionality reduction algorithm related information. Hence, we plan to extend our results to other dimensionality reduction algorithms as future work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Mahfouz A, van de Giessen M, van der Maaten L, Huisman S, Reinders M, Hawrylycz MJ, Lelieveldt BP (2015) Visualizing the spatial gene expression organization in the brain through nonlinear similarity embeddings. Methods 73:79–89. https://doi.org/10.1016/j.ymeth.2014.10.004
2. Townes FW, Hicks SC, Aryee MJ, Irizarry RA (2019) Feature selection and dimension reduction for single-cell rna-seq based on a multinomial model. Genom Biol 20(1):116. https://doi.org/10.1186/s13059-019-1861-6
3. Kobak D, Berens P (2019) The art of using t-sne for single-cell transcriptomics. Nat Commun 10(5416). https://doi.org/10.1038/s41467-019-13056-x
4. Meng C, Zeleznik OA, Thallinger GG, Kuster B, Gholami AM, Culhane AC (2016) Dimension reduction techniques for the integrative analysis of multi-omics data. Brief Bioinform 17(4):628–641. https://doi.org/10.1093/bib/bbv108
5. Warmerdam VD, Kober T, Tatman R (2020) Going beyond t-sne: Exposing what lies in text embeddings. In: Proceedings of second workshop for nlp open source software (NLP-OSS), pp 52–60. https://doi.org/10.18653/v1/2020.nlposs-1.8
6. Huang S, Ward MO, Rundensteiner EA (2005) Exploration of dimensionality reduction for text visualization. In: Coordinated and multiple views in exploratory visualization (CMV’05), pp 63–74. https://doi.org/10.1109/CMV.2005.8
7. Gashi I, Stankovi V, Leita C, Thonnard O (2009) An experimental study of diversity with off-the-shelf antivirus engines. In: Network computing and applications, pp 4–11. https://doi.org/10.1109/NCA.2009.14
8. Birjandtalab J, Poyuan MB, Nourani M (2016) Nonlinear dimension reduction for eeg-based epileptic seizure detection. In: 2016 IEEE-EMBS international conference on biomedical and health informatics (BHI), pp 595–598. https://doi.org/10.1109/BHI.2016.7455968
9. Hotelling H (1933) Analysis of a complex of statistical variables into principal components. J Educ Psychol 24(6):417–441. https://doi.org/10.1037/h0071325
10. Golub GH, Reinsch C (1970) Singular value decomposition and least squares solutions. Numer Math 14:403–420. https://doi.org/10.1007/BF02163027
11. Fisher R (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7(7):179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x
12. Schönkopf B, Smola A, Müller K-R (1997) Kernel principal component analysis. In: International conference on artificial neural networks - ICANN’97, pp 583–588. https://doi.org/10.1007/BFb0020217
13. Tenenbaum JB, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323. https://doi.org/10.1126/science.290.5500.2319
14. van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9:2579–2605
15. Ayesha S, Hanif MK, Talib R (2020) Overview and comparative study of dimensionality reduction techniques for high dimensional data. Inf Fusion 59:44–58. https://doi.org/10.1016/j.ifi sunk.2020.01.005

16. Gracia A, González S, Robles V, Menasalvas E (2014) A methodology to compare dimensionality reduction algorithms in terms of loss of quality. Inf Sci 270(1):1–27. https://doi.org/10.1016/j.ins.2014.02.068

17. van der Maaten L, Postma E, van den Herik J (2009) Dimensionality reduction: a comparative review. J Mach Learn Res 10(3):66–71

18. Lee JA, Verleysen M (2010) Scale-independent quality criteria for dimensionality reduction. Pattern Recogn Lett 31(14):2248–2257. https://doi.org/10.1016/j.patrec.2010.04.013

19. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324. https://doi.org/10.1109/5.726791

20. Lein ES, Hawrylycz M et al (2007) N.A.: Genome-wide atlas of gene expression in the adult mouse brain. Nature 445:168–176. https://doi.org/10.1038/nature05453

21. Xu X, Xie Z, Yang Z, Li D, Xu X (2020) A t-sne based classification approach to compositional microbiome data. Front Genet 11:620143

22. Sokooti H, Saygili G, Glocker B, Lelieveldt BPF, Staring M (2016) Accuracy estimation for medical image registration using regression forests. In: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016, pp 107–115. https://doi.org/10.1007/978-3-319-46726-9_13

23. Eppenhof KAJ, Plum JPW (2018) Error estimation of deformable image registration of pulmonary ct scans using convolutional neural networks. J Med Imaging 5(2). https://doi.org/10.1117/1.JMI.5.2.024003

24. Sokooti H, Saygili G, Glocker B, Lelieveldt BPF, Staring M (2019) Quantitative error prediction of medical image registration using regression forests. Med Image Anal 56:110–121. https://doi.org/10.1016/j.media.2019.05.005

25. Saygili G (2018) Local-search based prediction of medical image registration error. In: SPIE Medical Imaging. https://doi.org/10.1117/12.2293740

26. Tosi F, Poggio M, Benincasa A, Mattoccia S (2018) Beyond local reasoning for stereo confidence estimation with deep learning. In: European conference on computer vision, pp 319–334. https://doi.org/10.1007/978-3-319-46726-9_13

27. Pfeiffer D, Gehrig S, Schneider N (2013) Exploiting the power of nonlinear stochastic embedding. J Immunol 196(2):924–932. https://doi.org/10.4049/jimmunol.1501928

28. Kim S, Min D, Kim S, Sohn K (2019) Unified confidence estimation networks for robust stereo matching. IEEE Trans Image Process 28(3):1299–1313. https://doi.org/10.1109/TIP.2018.2878325

29. Saygili G (2020) A supervised learning approach for detecting erroneous samples in embeddings. Turk J Electr Eng Comput Sci 28(5):2883–2894. https://doi.org/10.3906/elk-1909-162

30. Morarui C, Biba A, Cutura R, Frémy B, Sédmail M (2021) Dumbledr: Predicting user preferences of dimensionality reduction projection quality. arXiv e-prints

31. Lee AJ, Verleysen M (2009) Quality assessment of dimensionality reduction: Rank-based criteria. Neurocomputing 72(7–9):1431–1443. https://doi.org/10.1016/j.neucom.2008.12.017

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.