Neurocomputing 150 (2015) 570–582

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Doubly supervised embedding based on class labels and intrinsic clusters for high-dimensional data visualization

Hannah Kim [a], Jaegul Choo [a,*], Chandan K. Reddy [b], Haesun Park [a]

[a] Georgia Institute of Technology, Atlanta, GA 30332, USA
[b] Wayne State University, Detroit, MI 48202, USA

ARTICLE INFO

Article history:
Received 2 December 2013
Received in revised form
20 September 2014
Accepted 25 September 2014
Available online 27 October 2014

Keywords:
Supervised dimension reduction
Multidimensional projection
T-distributed stochastic neighbor embedding
Visualization
Scatter plot
Clustering

ABSTRACT

Visualization of data can assist decision-making processes by presenting the underlying information in a perceptible manner. Many dimension reduction techniques have been proposed to generate faithful visualization snapshots given high-dimensional data. When class labels associated with the data are already provided, supervised dimension reduction methods, which utilize such pre-given label information as well as the data, have been effective in revealing the overall structure of data with respect to their pre-given class labels. However, the main principle of most of these supervised methods has been to enhance class separability, which generally leads to significant distortion of original relationships. To compensate for such distortion, we propose *a novel doubly supervised dimension reduction approach that highlights both natural groupings conforming to original relationships and classes determined by pre-given labels*. Our method imposes minimal supervision on the pre-given class information depending on their original distributions while imposing additional supervision on natural groupings to better preserve them in reduced feature space. Specifically, we apply the notion of doubly supervised dimension reduction to a state-of-the-art method called t-distributed stochastic neighbor embedding and present a new formulation and an algorithm. By performing both quantitative and qualitative analyses, we demonstrate the effectiveness of our method using various visualization examples on real-world data. Our results show that, compared to other existing methods, the proposed method better preserves the original high-dimensional relationships while simultaneously maintaining class separability and preserving cluster structures. In addition, due to the characteristics of preserving natural groupings, the visualization results generated by our method reveal interesting sub-groups that cohesively preserve the original relationships in the data.

1. Introduction

In the big data era, data are being collected easily in many settings in industry, science, and engineering. However, analysis on them is becoming more challenging than ever because of their complex nature and scale, often obfuscating the tasks to be solved. In these problematic situations, visualization can be helpful in facilitating decision-making processes by providing users with an overview of data.

Most real-world data are typically encoded as high-dimensional vectors as they can be effectively represented using a large number of features. One of the key methods for visualizing high-dimensional data in the form of a 2D/3D scatter plot is dimension reduction, and several well-known dimension reduction methods such as principal component analysis (PCA) [1] and multidimensional scaling (MDS) [2] have been widely applied in visualization applications. In general, the main idea behind most dimension reduction methods is to preserve the original high-dimensional relationships as much as possible in a lower-dimensional space. For example, PCA achieves this goal by maximizing the variance of the data in a low-dimensional space, and MDS tries to approximate all given pairwise similarity/distance values.

In many cases, however, additional information is available about the high-dimensional data. One of such additional information is class labels of the individual data points, indicating pre-given groupings of data. Unlike the previous unsupervised methods, which use only high-dimensional data as input, another type of methods called supervised dimension reduction utilizes such pre-given class labels when reducing the dimensions. Supervised dimension reduction, such as linear discriminant analysis (LDA) [3–5], has been successfully applied in numerous classification applications in machine learning and data mining (e.g., facial recognition [6]), and visual analytics [7,8]. Given these labels, supervised dimension reduction generally enhances class separation in lower-dimensional

* Corresponding author.
E-mail addresses: hannahkim@gatech.edu (H. Kim), jaegul.choo@cc.gatech.edu (J. Choo).

http://dx.doi.org/10.1016/j.neucom.2014.09.064
0925-2312/© 2014 Elsevier B.V. All rights reserved.

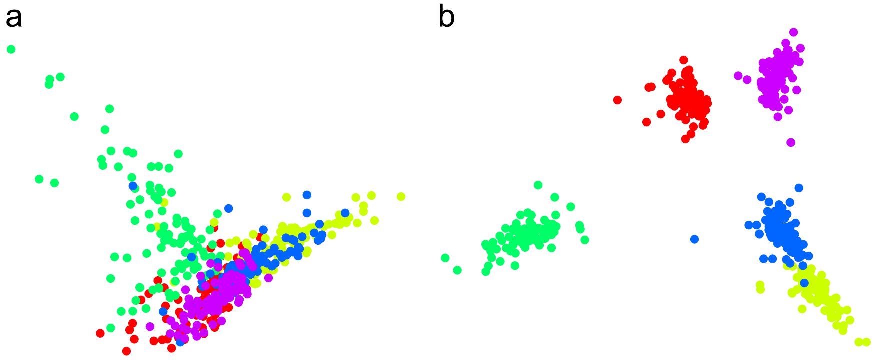


Fig. 1. Comparison of 2D visualization of Medline data described in Section 5.1. (a) Principal component analysis (PCA); (b) Linear discriminant analysis (LDA).

representations of data. Unlike unsupervised methods such as PCA, as shown in Fig. 1(a), supervised dimension reduction plays an important role in visualization by highlighting the class structure of data, as shown in Fig. 1(b). In this manner, supervised dimension reduction effectively performs the first step in a well-known visual information seeking mantra: Overview first, zoom and filter, then details-on-demand [9] by producing a visual overview with the class information highlighted.

Nonetheless, the two different criteria of separating classes and preserving original high-dimensional relationships could conflict with each other. For example, the widely used LDA aims at maximizing inter-class distances while minimizing intra-class ones, which could significantly distort the original relationships of the high-dimensional data. Although this issue might not be important in other applications such as classification, it could cause a problem in visualization tasks. That is, together with an overview at a class level, users would want the original relationships in data to be faithfully represented in visualization.

To effectively handle this trade-off problem in visualization applications, we propose a novel supervised dimension reduction approach. Basically, our proposed method incorporates a concept of *intrinsic clusters*, which takes into account natural groupings inherent in the data, to dimension reduction. As opposed to the classes that are externally formed by pre-given label information, intrinsic clusters are computed by a clustering algorithm purely based on the original high-dimensional relationships. Such intrinsic clusters provide a means to better preserve the original relationships in addition to the class separation capabilities available in the existing supervised dimension reduction methods.

The proposed method contains two important characteristics. First, we adaptively impose supervision on different classes depending on how clearly they are separated from the rest. In other words, we impose strong supervision on poorly separated classes so that they are visually distinct while imposing weak supervision on already well separated ones. In this manner, unnecessary distortion will be avoided. Second, we try to actively enhance the structure of intrinsic clusters by highlighting the separation between them. As a result, our method can properly capture original relationships while maintaining class separation in visualization. To realize our approach, we have chosen a state-of-the-art dimension reduction method, t-distributed stochastic neighbor embedding (t-SNE) [10], which has been applied successfully in various visualization applications, and we have extended it to what we call doubly supervised t-SNE.

The contribution of our work is summarized as follows:

- We introduce a novel concept of *double supervision* on dimension reduction based on pre-given class information as well as inherent clusters reflecting the natural groupings of data.
- We develop the formulation and algorithm of our novel dimension reduction method, doubly supervised t-SNE, which can separate pre-given classes as well as preserve the high-dimensional structure of the data.
- We evaluate the proposed method on various real-world data sets and demonstrate both quantitative and qualitative results.

The rest of the paper is organized as follows. Section 2 describes prior work related to dimension reduction. Section 3 introduces a widely used dimension reduction technique, t-SNE, as well as its basic extensions to supervised t-SNE. Next, Section 4 discusses the proposed methodology, and Section 5 presents our experiments. Finally, Section 6 concludes the paper.

2. Related work

Many dimension reduction techniques have been proposed in the past. The main goal of dimension reduction is to model high-dimensional data in a low-dimensional space such that the original information conveyed in a high-dimensional space is preserved as much as possible. Dimension reduction techniques attempt to achieve this goal by optimizing various objective and cost functions. For example, MDS [2] minimizes the sum of squared errors in terms of the pairwise distances of data items between high- and low-dimensional spaces. Isomap [11] works similar to MDS except that it uses geodesic pairwise distances approximated by the shortest path distances on k-nearest neighbor graphs instead of Euclidean pairwise distances of MDS. Another family of methods employs probabilistic formulation and objective functions. For example, stochastic neighbor embedding (SNE) [12] and t-distributed SNE (t-SNE) minimize the Kullback–Leibler divergence, a commonly used difference measure in probability, between the probability distributions derived from pairwise distance relationships in high- and low-dimensional spaces. However, these methods do not directly consider the original data grouping information, and thus they are called unsupervised methods.

Unlike the above-mentioned unsupervised methods, supervised methods (e.g., LDA [3]) assume that label information indicating the class structure in the data is already given and try to directly incorporate it into the dimension reduction process. In visualization, this grouping information has been actively used in various methods such as self-organizing maps (SOM) [13], where data clusters are naturally revealed during the dimension reduction process. To highlight the pre-given class structure, most supervised methods minimize distances within the same classes while maximizing those between different classes. In other supervised methods, a simple supervised extension of unsupervised methods via pre-given label information is to append the given label information as an additional dimension to the original high-dimensional representation of the data, which has been applied previously in SOM [13]. However, since the label information is generally represented as a numeric vector, if the scale of this label vector is too large compared to the scale of other data features, the low-dimensional structure could be dominated by the label information alone. On the other hand, if the scale of the label is too small, the label information is mostly ignored in the low-dimensional representation. In addition, even though the original class structure is not ordered, a numeric representation of the label information inevitably conveys ordered relationships between the classes (e.g., class 1 is considered to be closer to class 2 than to class 10). Thus, such an extension that treats the label information as just another feature could significantly distort the topology of the original relationships.

Other techniques modify the distance metrics to take advantage of the class information. For example, Fisher SOM [14] applies the Fisher information metric to SOM [13]. The Fisher information metric considers two data points that have similar class probability distribution close to each other. Recently, the Fisher information metric has been widely adopted by many supervised dimension reduction approaches such as supervised neighbor retrieval visualizer (S-NeRV) [15], discriminative t-SNE [16], and Fisher kernel t-SNE [17]. However, the Fisher information metric has several drawbacks. That is, to estimate class probability density, it requires an additional assumption such as a unimodal Gaussian distribution per class, which may not be the case in many complex real-world data sets.

Another set of approaches based on simple, intuitive transformations on the distance metric has also been proposed. One of the most straightforward methods performs distance scaling among data points belonging to the same class by a fixed ratio [18]. In other words, within-class distances are linearly decreased so that each class is represented more compactly. While this method generally produces visualization with well-separated classes, it can potentially cause excessive distortion of the original data relationships. Another distance transformation [19] that has been applied in Isomap takes a more sophisticated approach for better class separation. This approach does not allow any pairwise distances between different classes to be closer than those within the same class while restricting the latter to always be smaller than a prespecified value. In this manner, the classes in the low-dimensional representation are often overly separated even when the two classes are significantly overlapped in the original space. In this case, the method fails to convey original relationship information other than class information. In addition, these two methods apply the same supervision to all classes. That is, they decrease intra-class distances in the same manner, ignoring the characteristics of each class (e.g., whether a class is well separated from the others or not). On the other hand, our method allows users to obtain a low-dimensional mapping that adaptively imposes supervision on different classes.

In the following, we propose a supervised dimension reduction approach called doubly supervised dimension reduction, which seeks a low-dimensional representation that preserves dissimilarities between data item pairs and maintains class separability for effective visual analysis by utilizing pre-given class labels as well as intrinsic clusters.

3. t-distributed stochastic neighbor embedding

In this section, we briefly describe t-distributed stochastic neighbor embedding (t-SNE). Afterwards, we propose two basic supervised versions of t-SNE by manipulating the input distance based on the class labels.

3.1. t-distributed stochastic neighbor embedding (t-SNE)

t-distributed stochastic neighbor embedding (t-SNE) [10] is one of the most popular dimensionality reduction techniques in visualization applications. t-SNE extends the main idea of stochastic neighbor embedding (SNE) in which a pairwise similarity/distance value is converted into the probability with which a particular data point will choose another data point as its neighbor. That is, a closely related point is more likely to be chosen than a remotely related one. Considering the two different probability distributions, one of which is derived from the original high-dimensional space and the other from the low-dimensional space, both SNE and t-SNE aim at minimizing the difference between the two distributions, formulated as the Kullback–Leibler divergence, a commonly adopted difference measure for probability distributions. The main difference between t-SNE and SNE is that t-SNE uses a student t-distribution for the low-dimensional space and a Gaussian distribution for the high-dimensional space while SNE uses Gaussian distributions for both spaces. Additionally, for simpler and faster computation of gradients, t-SNE uses joint probability distributions instead of conditional probability distributions, which is adopted in SNE.

Given high-dimensional data points $x_i \in \mathbb{R}^n$ for $i = 1, \ldots, N$, let us define $d(x_i, x_j)$ as the Euclidean distance between the two data points. We calculate the joint probability p_{ij} in the high-dimensional space as symmetrized conditional probability $p_{ij} = \frac{1}{2}(p_{i|j} + p_{j|i})$, where $p_{i|j}$ is defined by

$$p_{i|j} := \frac{\exp(-d^2(x_i, x_j)/2\sigma^2)}{\sum_{r \neq s} \exp(-d^2(x_r, x_s)/2\sigma^2)}, \tag{1}$$

which is based on a Gaussian distribution with a variance σ. Let us denote $y_i \in \mathbb{R}^d$ as the low-dimensional coordinate of x_i after dimension reduction is performed, where d is typically set to 2 or 3 in visualization applications. The joint probability q_{ij} for the low-dimensional space is defined by

$$q_{ij} := \frac{(1 + d^2(y_i,\ y_j))^{-1}}{\sum_{r \neq s}(1 + d^2(y_r,\ y_s))^{-1}},$$

which is based on a student t-distribution with one degree of freedom.

The goal of t-SNE is to find a low-dimensional embedding y_i that minimizes the difference between high-dimensional probability distribution p_{ij} and low-dimensional probability one q_{ij}. This is achieved by minimizing the Kullback–Leibler divergence between them, which is defined by

$$KL(P \parallel Q) := \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{2}$$

The minimization of Eq. (2) is done via a gradient descent method, where the gradient of Eq. (2) with respect to y_i is given by

$$\frac{\partial}{\partial y_i} KL(P \parallel Q) = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + d^2(y_i,\ y_j))^{-1}. \tag{3}$$

The gradient can be understood as the sum of forces between item i and its neighbor j. Term p_{ij} acts as a pulling force put on y_i towards y_j, and term q_{ij} acts as a pushing force put on y_i away from y_j. When the low-dimensional distance between the two items is too large (or too small) compared to their corresponding high-dimensional distance, the pulling force becomes greater than (or less than) the pushing force in the low-dimensional embedding process. Thus, the forces serve the purpose of decreasing the mismatch between the probabilities. Starting with a random initialization on y_i's, t-SNE iteratively updates them based on Eq. (3).

The main advantage of t-SNE for visualization is due to the heavy-tailed characteristic of a student t-distribution for the low-dimensional space, which plays a role of alleviating the crowding problem: there is not enough room in the low-dimensional space for embedding all the neighbors in the high-dimensional space. Since moderately distant pairs of data items can be flexibly mapped with relatively large distances in the low-dimensional space, t-SNE can model distances between closely related data points more accurately than SNE and other dimension reduction methods.

3.2. Linearly supervised distance transformation extension to t-SNE (LS t-SNE)

In many cases, visualization tasks come with auxiliary class label information. This additional information can be utilized in dimensionality reduction so that the class structure can be highlighted. In the following, we propose two supervised extensions to t-SNE. The main idea behind our supervised extension is to transform the input distances based on the class label information. To this end, we utilize a linearly supervised distance transformation (LS t-SNE). LS t-SNE scales the original distance between a point and its neighbor that belongs to the same class by a factor of λ_{LS}, where $\lambda_{LS} < 1$, which will then be used as an input to t-SNE. Previously, this linear-shrinking approach has been applied to other dimension reduction methods [18]. This approach is simple and easy to understand since it uses weighted distances to supervise dimension reduction. In the following, we describe this extension to t-SNE, namely LS t-SNE, which employs this scaling process.

Given a high-dimensional data point $x_i \in \mathbb{R}^n$ and its neighbor $x_j \in \mathbb{R}^n$ along with their (pre-**g**iven) class labels l_i^g and l_j^g, respectively, their linearly scaled distance $d_{LS}(x_i, x_j)$ with a scaling parameter λ_{LS} is calculated as

$$d_{LS}(x_i, x_j) := \begin{cases} \lambda_{LS} d(x_i, x_j) & \text{if } l_i^g = l_j^g \\ d(x_i, x_j) & \text{otherwise} \end{cases}. \quad (4)$$

This transformation shrinks intra-class distances while keeping inter-class distances unchanged. We use such weighted distances to calculate the probability distribution for the original high-dimensional space, $p_{i|j}$, by replacing $d(x_i, x_j)$ with $d_{LS}(x_i, x_j)$ in Eq. (1). The rest of the process is the same as t-SNE. One of the major limitations of this weighted approach is that every intra-class pair is scaled to the same degree, ignoring the structure and the distribution of each class. For instance, insufficiently separated classes may need more supervision than already well separated classes.

3.3. Exponentially supervised distance transformation extension to t-SNE (ES t-SNE)

We propose another supervised distance transformation technique, namely exponentially supervised distance transformation (ES). ES t-SNE, similar to the linearly supervised approach in the sense that both decrease the inner-class distances, uses an exponential transformation instead of linear scaling.

Given high-dimensional data point $x_i \in \mathbb{R}^n$ and its neighbor $x_j \in \mathbb{R}^n$ along with their (pre-**g**iven) class labels l_i^g and l_j^g, respectively, the exponentially transformed distance $d_{ES}(x_i, x_j)$ is computed as

$$d_{ES}(x_i, x_j) := \begin{cases} \sqrt{1 - \exp(-d^2(x_i, x_j)/\beta_{ES})} & \text{if } l_i^g = l_j^g \\ \sqrt{\exp(d^2(x_i, x_j)/\beta_{ES})} - \alpha_{ES} & \text{otherwise} \end{cases}, \quad (5)$$

where α_{ES} and β_{ES} are supervision parameters. This transformation was originally introduced in [19], which proposed a supervised version of Isomap [20]. According to [19], α_{ES} should be less than 0.65 to keep the transformed inter-class distances bigger than those of intra-class distances that originally have the same high-dimensional distance values. Usually, β_{ES} is set as the average pairwise Euclidean distance in the original high-dimensional space. This transformation has the following properties. First, intra-class distances have an upper bound of 1 while inter-class distances have a lower bound of $1 - \alpha_{ES}$. Second, as the original high-dimensional distances increase, their transformed inter-class distances increase rapidly while their transformed intra-class distances converge asymptotically to 1.

We extend t-SNE with this distance transformation scheme. In Eq. (1), we replace $d(x_i, x_j)$ with $d_{ES}(x_i, x_j)$ and then follow the rest of the procedure in t-SNE. The main drawback of this metric is that it imposes strong separation between classes regardless of their original relationships. In other words, it makes already large inter-class distances transformed to excessively large distances because of the second property (i.e., rapidly increasing inter-class distances), which would result in unnecessary distortion of the original relationships in the final results of t-SNE.

4. Doubly supervised t-distributed stochastic neighbor embedding (DS t-SNE)

In this section, we propose a novel supervised dimension reduction method by introducing the concept of intrinsic clusters, which represent natural groupings within the original high-dimensional data. As opposed to pre-given class labels, which do not necessarily reflect original relationships, intrinsic clusters provide a way of better capturing original relationships, still under a supervised dimension reduction setting. By simultaneously incorporating both kinds of supervision based on pre-given classes and intrinsic clusters, we propose the idea of doubly supervised dimension reduction and extend t-SNE based on this idea.

4.1. Intrinsic clusters

We define the concept of *intrinsic clusters* as natural groupings inherent in an original high-dimensional space. In order to compute intrinsic clusters, we apply clustering techniques such as *k*-means to original high-dimensional data. In doubly supervised dimension reduction we propose in this paper, the grouping information of intrinsic clusters works as another set of labels in addition to pre-given class labels so that original relationships at an intrinsic cluster level can be highlighted.

In general, clustering high-dimensional data is a challenging problem due to the curse of dimensionality [21]. Therefore, in many cases, choosing a clustering method and its parameters (e.g., the number of clusters) that are the most suitable for our data at hand is still an open question. To properly obtain the intrinsic clusters, we use *k*-means with the number of intrinsic clusters, K, chosen as follows: We set the value of K in the range from half the number of pre-given classes, $\lfloor M/2 \rfloor$, to 2 M, where M is the number of pre-given classes, and for each K, we run *k*-means five times and choose the best result with the smallest objective function value. We then measure the quality of the clustering results for a different value of K by computing the Davies–Bouldin index [22] and choose the one with the best quality measure, as will be shown later in Section 5.1.

4.2. Adaptive supervision by pre-given class labels

Most existing supervised dimension reduction methods impose supervision on different classes in the same manner, regardless of whether they are well separated or not. On the other hand, our method aims to adaptively put stronger supervision on poorly separated classes than on those classes that are well separated. For this purpose, let us first denote the pre-given class label and the intrinsic cluster label of x_i as l_i^g and l_i^c, respectively. Given M classes and K intrinsic clusters, we construct a class-by-cluster confusion matrix $C \in \mathbb{R}^{M \times K}$, of which the (m,k)-th element c_{mk} is defined as $c_{mk} := card\{x_i : \ l_i^g = m, \ l_i^c = k\}$,

where $card(A)$ denotes the cardinality of set A. That is, c_{mk} represents the number of data points that belong to the m-th pre-given class and the k-th intrinsic cluster.

Now, we want to supervise classes proportional to the degree of impurity for each class. In other words, the case in which a class is distributed over many intrinsic clusters indicates that the data points in this class are poorly separated, and thus it requires stronger supervision. On the other hand, the other case in which a class is distributed over a small number of intrinsic clusters indicates that the class is coherent and/or compact. To formulate this idea, we first adopt an impurity measure of a particular class as the entropy of the data distribution of the class over intrinsic clusters. The entropy of the m-th class, $H(m)$, is expressed as

$$H(m) := -\sum_k \frac{c_{mk}}{\sum_{k'} c_{mk'}} \log \frac{c_{mk}}{\sum_{k'} c_{mk'}}.$$

By utilizing this impurity measure, we modify the joint probability distribution generated from the original high-dimensional relationships, p_{ij}, in Eq. (1) and define a new joint probability distribution $\hat{p_{ij}}$ as

$$\hat{p_{ij}} := \begin{cases} \alpha_{DS} \exp(H(m)) p_{ij} & \text{if } l_i^g = l_j^g \\ p_{ij} & \text{otherwise} \end{cases}, \tag{6}$$

where α_{DS} is a scaling parameter. Basically, this equation plays a role of increasing intra-class probabilities corresponding to those classes with high impurity measures. We introduce the additional parameter α_{DS} to control the degree of this adaptive supervision relative to the degree of the secondary supervision that will be described in Section 4.3. As shown in Fig. 2, class separation is highlighted for a large value of α_{DS}. After this step, $\hat{p_{ij}}$ is normalized to sum to 1.

4.3. Secondary supervision by intrinsic clusters

Supervision based on pre-given class labels may distort the original structure embedded in a low-dimensional space. Thus, to mitigate this distortion, we present a secondary supervision step using intrinsic clusters. Similar to the supervision on pre-given classes described in Section 4.2, we perform the secondary supervision on intrinsic clusters by increasing the probability values corresponding to pairwise relationships within each intrinsic cluster (i.e., the intra-cluster probability).

Given probability $\hat{p_{ij}}$ obtained from Eq. (6), we perform secondary supervision on intrinsic clusters and generate a new probability value $\tilde{p_{ij}}$ as

$$\tilde{p_{ij}} := \begin{cases} (1-\beta_{DS1})\hat{p_{ij}} + \beta_{DS1} & \text{if } l_i^c = l_j^c \\ \gamma_{DS}\hat{p_{ij}} & \text{otherwise} \end{cases}, \tag{7}$$

where $\beta_{DS1} := \Delta_{DS} / \sum_{l_i^c = l_j^c} (1-\hat{p_{ij}})$, $\gamma_{DS} := 1 - \Delta_{DS} / \sum_{l_i^c \neq l_j^c} \hat{p_{ij}}$, and Δ_{DS} is the total probability mass added to the intra-cluster probabilities (out of the inter-cluster probabilities). Note that the inter-cluster probabilities are scaled by a factor $\gamma_{DS} < 1$. It can be shown that the modified probability $\tilde{p_{ij}}$ satisfy the following equations:

$$\sum_{l_i^c = l_j^c} \tilde{p_{ij}} = \sum_{l_i^c = l_j^c} \hat{p_{ij}} + \Delta_{DS}, \ \sum_{l_i^c \neq l_j^c} \tilde{p_{ij}} = \sum_{l_i^c \neq l_j^c} \hat{p_{ij}} - \Delta_{DS}. \tag{8}$$

Thus, the maximum value that Δ_{DS} can have is computed as $\sum_{l_i^c \neq l_j^c} \hat{p_{ij}}$, which is the sum of inter-cluster probabilities. As shown in Fig. 3, the grouping of intrinsic clusters (e.g., the ellipses on Fig. 3(b)) becomes clearer as Δ_{DS}, which corresponds to the degree of supervision on intrinsic clusters, increases. For example, in Fig. 3(a), data points in the top-left corner are initially visualized as a single cluster, but as Δ_{DS} increases, they are divided into two sub-clusters in Fig. 3(c), highlighting intrinsic clusters in visualization.

Alternatively, one can use another approach that defines the probability $\tilde{p_{ij}}$ as

$$\tilde{p_{ij}} := \begin{cases} \beta_{DS2}\hat{p_{ij}} & \text{if } l_i^c = l_j^c \\ \gamma_{DS}\hat{p_{ij}} & \text{otherwise} \end{cases}, \tag{9}$$

where $\beta_{DS2} := 1 + \Delta_{DS} / \sum_{l_i^c = l_j^c} \hat{p_{ij}}$, $\gamma_{DS} := 1 - \Delta_{DS} / \sum_{l_i^c \neq l_j^c} \hat{p_{ij}}$. In this case, Eq. (8) still holds. The two approaches differ in terms of how to allocate the total probability transferred from inter-cluster relationships to intra-cluster relationships. The first approach guarantees that any intra-cluster probability is equal to or greater than β_{DS1} as can be seen from Eq. (7), which would boost small intra-cluster probabilities to at least β_{DS1}. However, the second approach increases intra-cluster probabilities just linearly by a factor of β_{DS2} (Eq. (9)), which gives no such guarantees. Thus, a small intra-cluster probability may still remain relatively small even after the supervision.

4.4. Parameter selection

Our method has two key parameters, α_{DS} and Δ_{DS}, which control the degrees of supervision on pre-given classes and intrinsic clusters, respectively. By default, we set α_{DS} to 1 and Δ_{DS} to 0.1. When $\alpha_{DS} = 1$, we impose no supervision on the pre-given class that belongs entirely to a single intrinsic cluster (i.e., $\hat{p_{ij}} = p_{ij}$ in Eq. (6)). By having $\Delta_{DS} = 0.1$, our method transfers the probability mass of 0.1 from the inter- to intra-cluster relationships.

In visualization, it is relatively less important to find the best parameter value than in other applications since users can change parameter values and check visualized results in an iterative and interactive manner [23]. Other applications such as prediction tasks often seek to find the best parameter values as the nature of the task is to find the most accurate model that best performs on unseen data rather than to try multiple parameter values interactively. In visual analysis tasks, the capability of easily changing

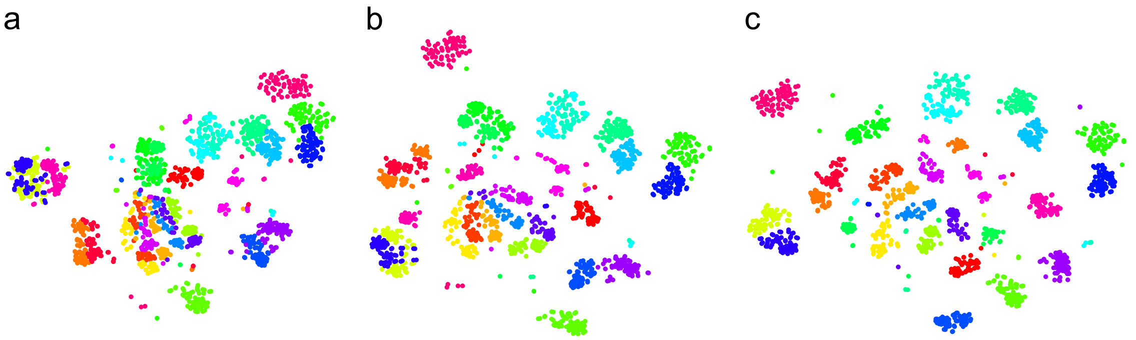


Fig. 2. Effect of α_{DB} on the 2D visualization of Isolet data described in Section 5.1. (a) $\alpha_{DB} = 0.5$; (b) $\alpha_{DB} = 1$; (c) $\alpha_{DB} = 2$.

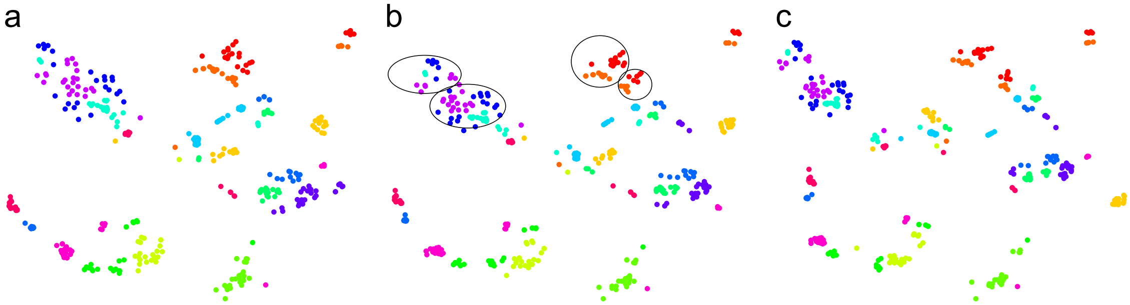


Fig. 3. Effect of Δ_{DS} on 2D visualization of Libras Movement data described in Section 5.1. (a) $\Delta_{DS}=0.1$; (b) $\Delta_{DS}=0.15$; (c) $\Delta_{DS}=0.2$.

parameters is more important in understanding various aspects of data from interactive visualization. Although not shown in this paper, we have developed a graphical user interface with which one can easily try out different parameter values. The visualization examples and the quantitative analysis shown in Section 5 are based on such processes.

4.5. Computational cost

DS t-SNE has three additional steps, (1) k-means clustering, (2) supervision on pre-given classes, and (3) supervision on intrinsic clusters, followed by the main dimension reduction step, which is the same as the original t-SNE. The original t-SNE has the computational complexity of $O(N^2d)$, where N is the total number of data items, and d is the reduced dimension. Among the three additional steps of DS t-SNE, a nontrivial amount of computations can take place in the k-means clustering step, which increases linearly in terms of both N and n, where n is the original dimension. This mainly happens when n is significantly larger than N. In this case, one can pre-reduce the original dimension to a reasonable number by applying a computationally efficient method such as PCA beforehand. In fact, this process is already included as part of the original implementation of t-SNE to efficiently compute all the pairwise distances $d(x_i, x_j)$'s in the original high-dimensional space in an approximate manner. By applying the same process before the k-means clustering step, the computational cost of this step becomes much smaller than the subsequent steps of the original t-SNE computation. Hence, DS t-SNE can be considered to have the equivalent computational complexity to that of t-SNE, i.e., $O(N^2d)$.

5. Experiments

To evaluate DS t-SNE, we compare it with other state-of-the-art methods such as t-SNE, LS t-SNE (Section 3.2), ES t-SNE (Section 3.3), LDA [3], and S-NeRV [15]. We first describe the data sets used in this paper as well as our experimental setup in Section 5.1. The quantitative results of our experiments are shown in Section 5.2. Finally, Section 5.3 analyze visualized results in depth.

5.1. Data sets and experimental setup

We used the following eight data sets: (1) Image Segmentation data set [24] (2310 data points, 19 dimensions, 7 classes), (2) Isolet data set [24] (1558 data points, 617 dimensions, 26 classes), (3) Libras Movement data set [24] (360 data points, 90 dimensions, 15 classes), (4) Medline data set[1] (550 data points, 22,095 dimensions, 5 classes), (5) MNIST data set[2] (5000 data points, 784 dimensions, 10 classes), (6) 20 Newsgroup data set[3] (770 data points, 16,702 dimensions, 11 classes), (7) Optical Recognition of Handwritten Digits data set [24] (5000 data points, 64 dimensions, 10 classes), and (8) Reuter data set[4] (880 data points, 3907 dimensions, 10 classes).

In the following experiments, the parameter α_{DB} (in Eq. (6)) is set to 1, and the parameter Δ_{DS} (in Eq. (8)) is chosen in the range of [0, 0.5] in DS t-SNE. The optimal number of intrinsic clusters determined based on the mean Davies-Bouldin index for each data set is shown in Table 1. In LS t-SNE, we tested the parameter λ_{LS} (in Eq. (4)) in the range of [0.1, 0.9]. In ES t-SNE, we tested the parameter α_{ES} (in Eq. (5)) in the range of $[-0.15, 0.65]$, and the parameter β_{ES} (in Eq. (5)) is set to the mean of all the pairwise Euclidean distances, unless mentioned otherwise. For other t-SNE parameters such as perplexity, momentum, and the number of iterations, we followed the default values suggested in [10]. In S-NeRV, we set λ to 0.1 and 0.3, following the original S-NeRV paper [15]. In all methods, we chose the parameters producing the best mean reciprocal rank and mean precision measures, which are described in Section 5.2.

5.2. Quantitative results

In this section, we evaluate the performance of our method and compare it with other methods based on various quantitative measures. We employ four different measures to capture the diverse characteristics of the results: (1) the mean precision, (2) the mean reciprocal rank, (3) the rank correlation, and (4) the k-NN classification accuracy. The first three are the measures to assess the preservation of the original high-dimensional relationships, and the last measure is about the preservation of the pre-given class structure in terms of classification accuracy. Among the three measures about the original relationships, the mean precision and the mean reciprocal rank evaluate local structure preservation while the rank correlation evaluates global structure preservation.

The mean precision is defined as

$$PR=\frac{1}{N}\sum_{i=1}^{N}\frac{card\{j\in P_i\cap Q_i,\ i\neq j\}}{card\{j\in Q_i,\ i\neq j\}},$$

where N is the total number of data items, P_i (or Q_i) is the index set of the neighborhood data points of data item i in the original high-dimensional space (or in the low-dimensional space), which consists of a fixed number of the nearest neighbors to data item i, and $card(A)$ is the cardinality of set A. As described in [15], we consider the 20 nearest neighbors of a data point as its relevant items.

[1] http://www.cc.gatech.edu/~hpark/othersoftware_data.php

[2] http://yann.lecun.com/exdb/mnist/index.html.

[3] http://qwone.com/~jason/20Newsgroups/

[4] http://www.cc.gatech.edu/~hpark/othersoftware_data.php

Table 1
Number of intrinsic clusters (K) and mean Davies–Bouldin index (DB) for each data set.

	Image Segmentation	Isolet	Libras movement	Medline	MNIST	20 News-group	Handwritten digits	Reuter
K	3	16	23	9	20	22	9	18
DB	0.76	2.28	1.20	7.11	2.61	2.31	1.78	4.04

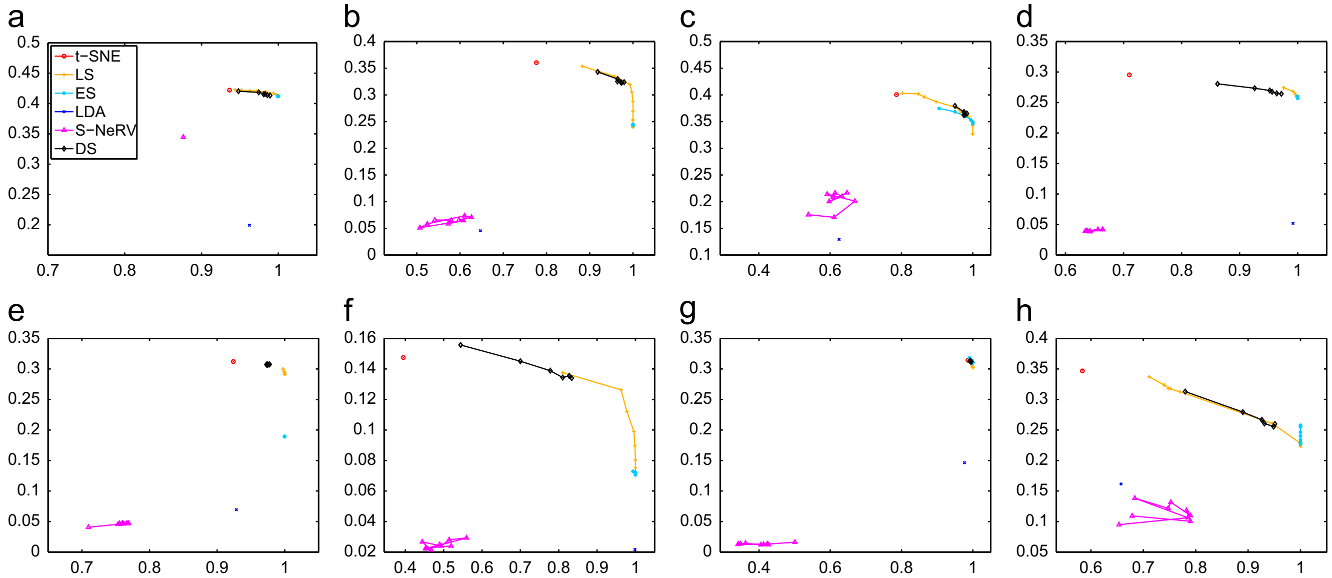


Fig. 4. k-NN classification accuracy (horizontal axis) – mean reciprocal rank (vertical axis) plotted for all data sets. The curves are parametrized by the supervision level of corresponding methods. (a) Image Segmentation data set; (b) Isolet data set; (c) Libras Movement data set; (d) Medline data set; (e) MNIST data set; (f) 20 Newsgroup data set; (g) Handwritten Digits data set; (h) Reuters data set.

Table 2
Comparison results of mean precision values. A higher value indicates better performance.

Data set	**t-SNE**	**LS t-SNE**	**ES t-SNE**	**LDA**	**S-NeRV**	**DS t-SNE**
Image Segmentation	0.7389	0.7056	0.6981	0.3359	0.6932	0.7364
Isolet	0.5688	0.5374	0.4437	0.1573	0.1647	0.5401
Libras Movement	0.6976	0.5143	0.4058	0.2624	0.4325	0.5667
Medline	0.3684	0.3185	0.3165	0.1379	0.1220	0.2813
MNIST	0.4693	0.4311	0.3137	0.1490	0.1238	0.4212
20 Newsgroup	0.2044	0.1387	0.1458	0.0398	0.0927	0.1374
Handwritten Digits	0.4957	0.4948	0.4919	0.2961	0.1819	0.4976
Reuters	0.4073	0.3777	0.3490	0.2277	0.2513	0.3962

Table 3
Comparison results of mean reciprocal rank values. A higher value indicates better performance.

Data set	**t-SNE**	**LS t-SNE**	**ES t-SNE**	**LDA**	**S-NeRV**	**DS t-SNE**
Image Segmentation	0.4221	0.4147	0.4121	0.1993	0.3446	0.4203
Isolet	0.3605	0.2531	0.2433	0.0456	0.0578	0.3431
Libras Movement	0.4005	0.3597	0.3469	0.1294	0.2137	0.3794
Medline	0.3266	0.2587	0.2585	0.0520	0.0404	0.2806
MNIST	0.3234	0.2910	0.1894	0.0694	0.0473	0.3076
20 Newsgroup	0.1475	0.0752	0.0717	0.0217	0.0283	0.1340
Handwritten Digits	0.3142	0.3031	0.3103	0.1464	0.0666	0.3045
Reuters	0.3468	0.2289	0.2339	0.1616	0.1383	0.3130

Table 4
Comparison results of distance rank correlation values. A higher value mean stronger correlation.

Data set	**t-SNE**	**LS t-SNE**	**ES t-SNE**	**LDA**	**S-NeRV**	**DS t-SNE**
Image Segmentation	0.1685	0.2043	0.1164	0.4841	0.6498	0.1805
Isolet	0.6067	0.1425	0.1202	0.6403	0.0698	0.3453
Libras Movement	0.4366	0.3333	0.2641	0.3498	0.4995	0.2680
Medline	0.2397	0.2497	0.2416	0.3568	0.2193	0.2244
MNIST	0.3703	0.1072	0.1233	0.4713	0.4056	0.1122
20 Newsgroup	0.5062	0.0753	0.0681	0.0917	0.2459	0.2664
Handwritten Digits	0.2153	0.2104	0.2057	0.6375	0.0874	0.1745
Reuters	0.1706	0.3217	0.3135	0.4166	0.3090	0.1718

Table 5
Comparison results of k-NN classification accuracy values. A higher value indicates better performance.

Data set	**t-SNE**	**LS t-SNE**	**ES t-SNE**	**LDA**	**S-NeRV**	**DS t-SNE**
Image Segmentation	0.9364	0.9970	1.0000	0.9623	0.8762	0.9481
Isolet	0.7766	1.0000	1.0000	0.6470	0.5616	0.9185
Libras Movement	0.7861	0.9889	1.0000	0.6250	0.7056	0.9500
Medline	0.7620	1.0000	1.0000	0.9920	0.6440	0.8620
MNIST	0.9356	1.0000	0.9998	0.9284	0.7597	0.9784
20 Newsgroup	0.3948	1.0000	1.0000	0.9987	0.5000	0.8338
Handwritten Digits	0.9864	0.9996	0.9998	0.9766	0.6170	0.9878
Reuters	0.5838	0.9988	1.0000	0.6575	0.6838	0.7800

The mean reciprocal rank [25], which is a well-known information retrieval metric to evaluate top-k retrieved is computed as the reciprocal value of the harmonic mean of the ranks of the original k nearest neighbors of a data item in the low-dimensional space in terms of their distances from the item, which is defined as

$$Mean\ Reciprocal\ Rank = \frac{1}{N}\sum_i \left(\frac{1}{k} \sum_{x_j \in \mathcal{N}_k^x(x_i)} \frac{1}{rank_i(y_j)} \right),$$

where N is the total number of data items, and $\mathcal{N}_k^x(x_i)$ is the set of the k nearest neighbors of x_i (with $k=5$), and $rank_i(y_j)$ is the

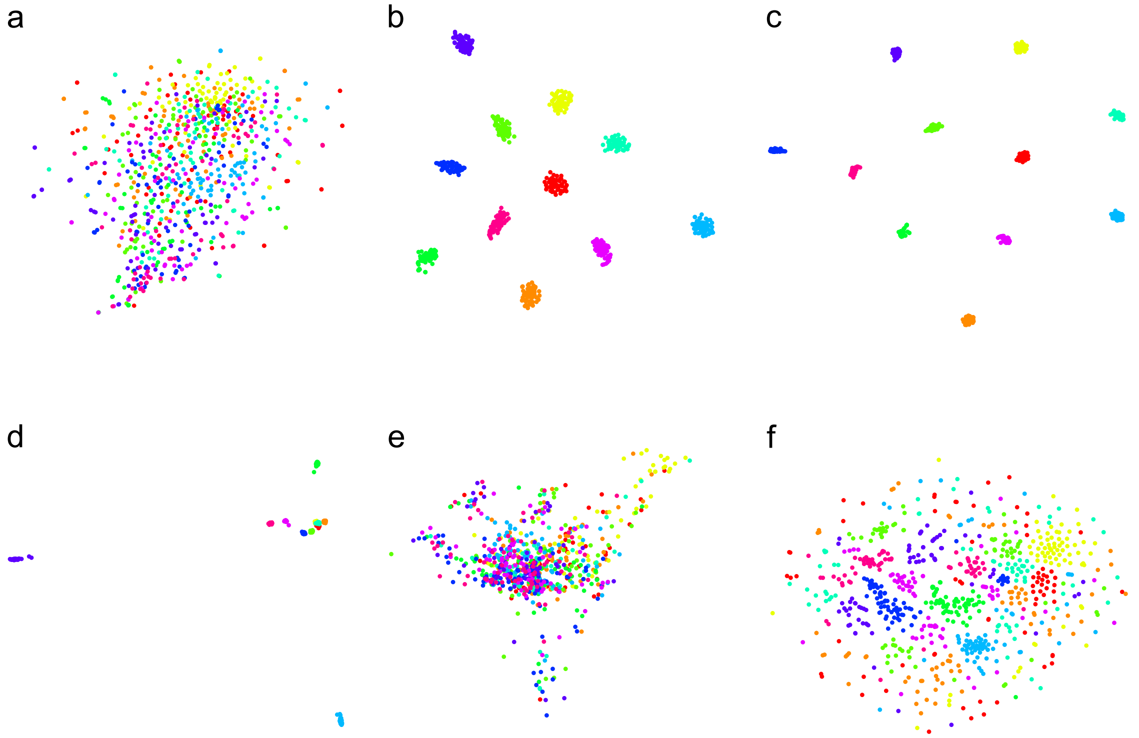


Fig. 5. Comparison of 2D visualizations of 20 Newsgroup data using different algorithms. (a) t-SNE; (b) LS t-SNE; (c) ES t-SNE; (d) LDA; (e) S-NeRV; (f) DS t-SNE.

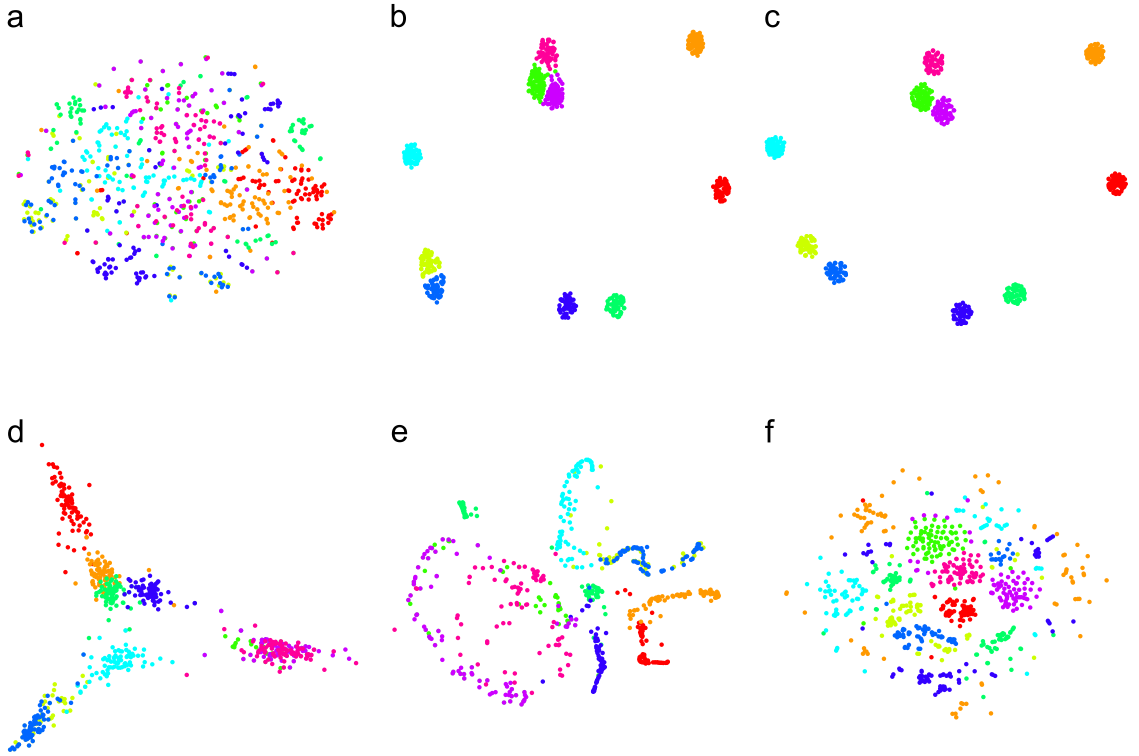


Fig. 6. Comparison of 2D visualizations of Reuters data using different algorithms. (a) t-SNE; (b) LS t-SNE; (c) ES t-SNE; (d) LDA; (e) S-NeRV; (f) DS t-SNE.

distance rank of data item j from data item i in the low-dimensional space. This measure indicates how closely the original k nearest neighbors are placed in the low-dimensional space.

The rank correlation computes Spearman's rank correlation coefficient ρ, which captures a monotonic association between two variables [26]. In this paper, we compute this measure by comparing the distance ranks from a data point to the other points in the original space with those in the embedded space. Unlike the previously mentioned measures based only on local neighbors, the rank correlation measure evaluates the preservation of a global structure by considering the entire data relationships.

The k-NN classification accuracy is computed as the prediction accuracy of class labels by performing k-NN classification in the low-dimensional space. That is, we compare the pre-given class

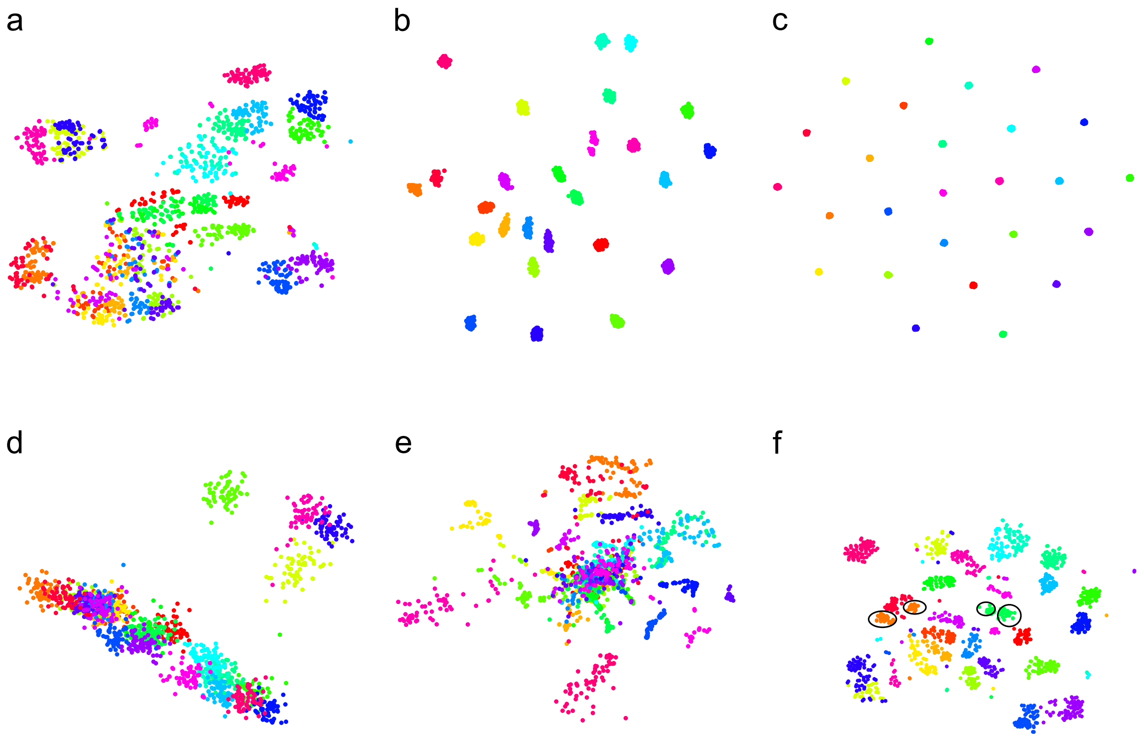


Fig. 7. Comparison of 2D visualizations of Isolet data using different algorithms. (a) t-SNE; (b) LS t-SNE; (c) ES t-SNE; (d) LDA; (e) S-NeRV; (f) DS t-SNE.

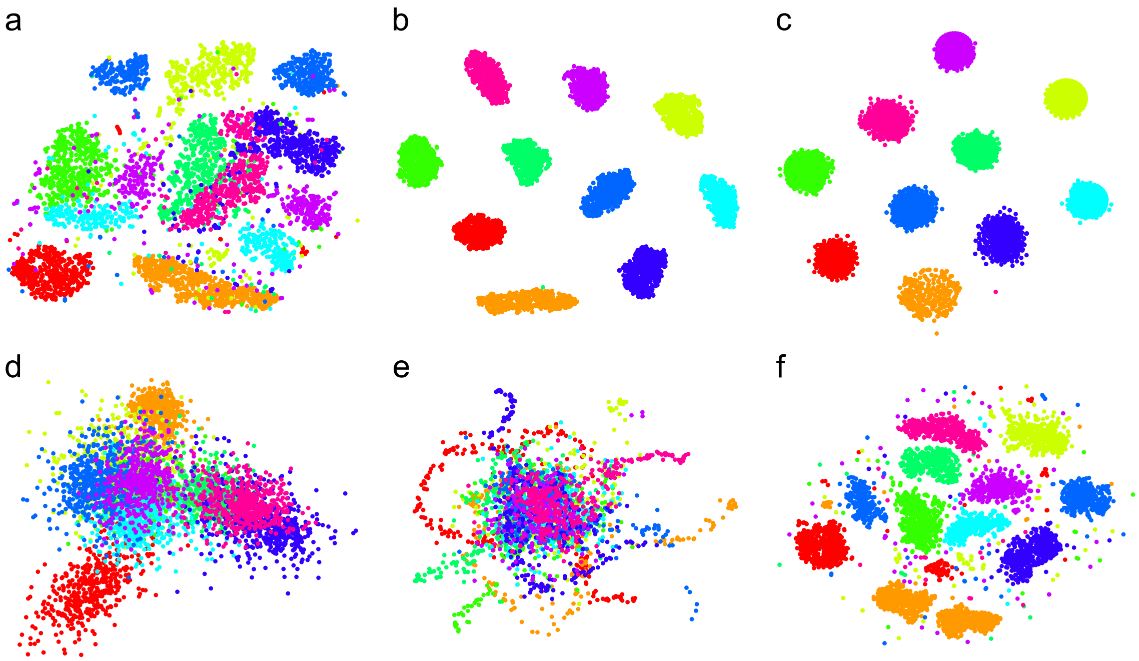


Fig. 8. Comparison of 2D visualizations of MNIST data using different algorithms. (a) t-SNE; (b) LS t-SNE; (c) ES t-SNE; (d) LDA; (e) S-NeRV; (f) DS t-SNE.

information of a data point and its k-nearest neighbor classifier output ($k=5$) in the low-dimensional space. This metric was previously used in the evaluation of S-NeRV [15].

The quantitative results are shown in Tables 2–5 as well as in Fig. 4. Tables 2 and 3 show the results of the mean precision and the mean reciprocal rank, respectively. These two measures indicate how well the original neighborhood structures in the high-dimensional space are preserved (regardless of class labels). In both sets of the results, t-SNE, the only unsupervised method among the compared methods, shows the highest performances for all the five data sets. This is expected since the supervised dimension reduction methods generally introduce the distortion of the given original relationships in return for enhanced class separability. Except for t-SNE, however, DS t-SNE is shown to perform the best for most of the data sets and shows the comparable performances to t-SNE (e.g., Medline, MNIST, and 20 Newsgroup data sets in Table 2 and Handwritten Digits data set in Table 3). Table 4 shows the Spearman's rank correlation

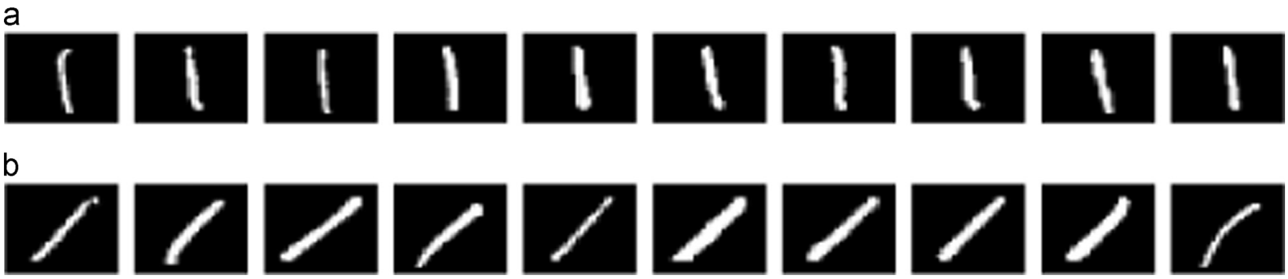


Fig. 9. Individual raw data examples of the sub-clusters shown in Fig. 8(f) of MNIST data. Each written digit is represented in a white color in a black background. (a) A sub-cluster in class ‘1’; (b) Another sub-cluster in class ‘1’.

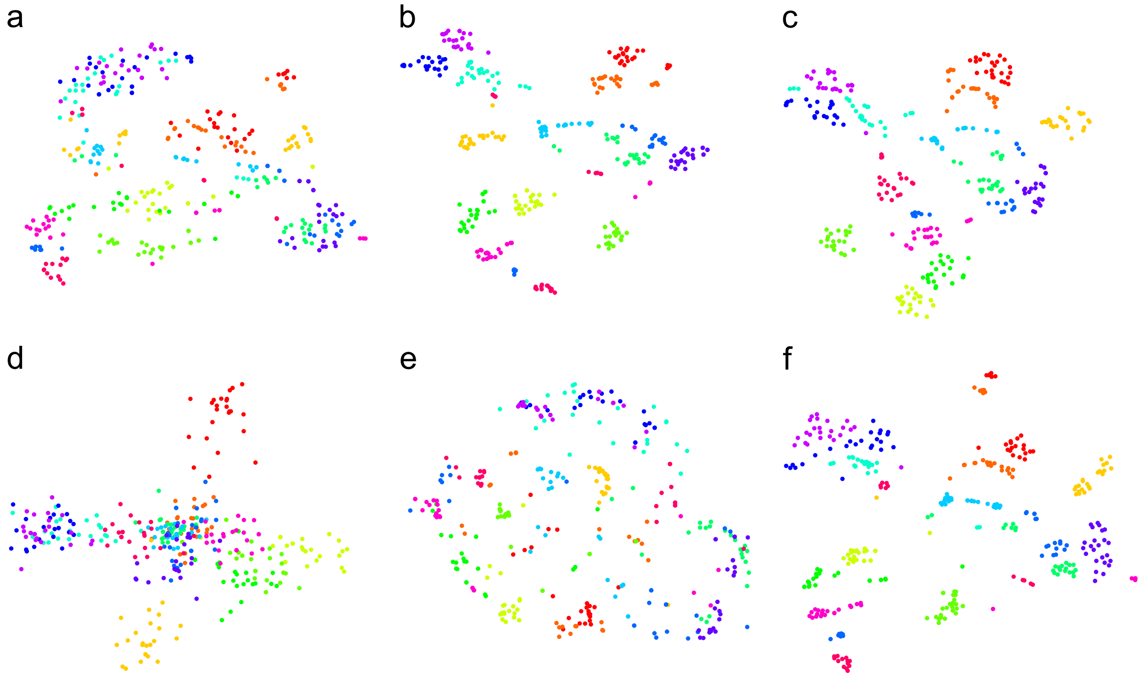


Fig. 10. Comparison of 2D visualizations of Libras Movement data using different algorithms. (a) t-SNE; (b) LS t-SNE; (c) ES t-SNE; (d) LDA; (e) S-NeRV; (f) DS t-SNE.

coefficients which evaluate the preservation of the global structure in the original high-dimensional space. Interestingly, DS t-SNE and the other t-SNE based methods are outperformed by LDA and S-NeRV in terms of global relationship preservation for most data sets. This is because the methods based on t-SNE, which adopts Student's t-distribution, emphasize local structure. However, users are usually more interested in the preservation of local relationships rather than the preservation of global relationships, and thus DS t-SNE can be viewed as one of the most competitive methods in terms of the original relationship preservation among the compared supervised methods.

Table 5 shows the classification accuracy results, which indicate how well the classes are separated from each other. LS t-SNE and ES t-SNE show almost perfect accuracy values for all the data sets. This is mainly because of severe class separation imposed by the dimension reduction methods, as seen in Figs 5–8, 10, and 12. When the classes are already well separated from each other in the original space, such strong supervision would not introduce much distortion. However, when the classes are not well separated in the original space, class separation can bring unnecessarily excessive distortion against the faithful representation of the original relationships in visualization. On the other hand, DS t-SNE does not only generate compelling visualization without such excessive distortion, as seen in Figs 5–8, 10, and 12, but also shows reasonably good classification accuracy ranging from .78 (Reuters) to .99 (Handwritten Digits), as shown in Table 5.

In order to show the balance between original relationship preservation and class separation, we plot the mean reciprocal rank versus the k-NN classification accuracy by varying the parameters of the compared methods, which are described in Section 5.2. A different parameter value leads to a particular pair of the two performance measures, the mean reciprocal rank and the k-NN classification accuracy, and as we increase/decrease the parameter value, we obtain a trajectory curve of these measure pairs. The curve placed near the top right region indicates high performances in terms of both measures. The result of this experiment is reported in Fig. 4. This result shows that DS t-SNE and LS t-SNE generally perform better than the other methods such as S-NeRV and LDA. Compared to the naive way of supervision in LS t-SNE, however, our carefully designed supervision performed in DS t-SNE leads to more visually appealing results, as will be seen in the next section.

5.3. Visual analysis

Figs 5–8, 10, and 12 show our visualization results generated by various methods. First of all, in most cases, our proposed method, DS t-SNE, successfully shows both the class separation and the original relationships by leveraging intrinsic clusters as well as pre-given classes. In the case of text documents such as 20 Newsgroup and Reuters data sets (Figs. 5 and 6), an unsupervised method, t-SNE, does not properly reveal the clear class structure, partly because of the lack of coherence within each topic class,

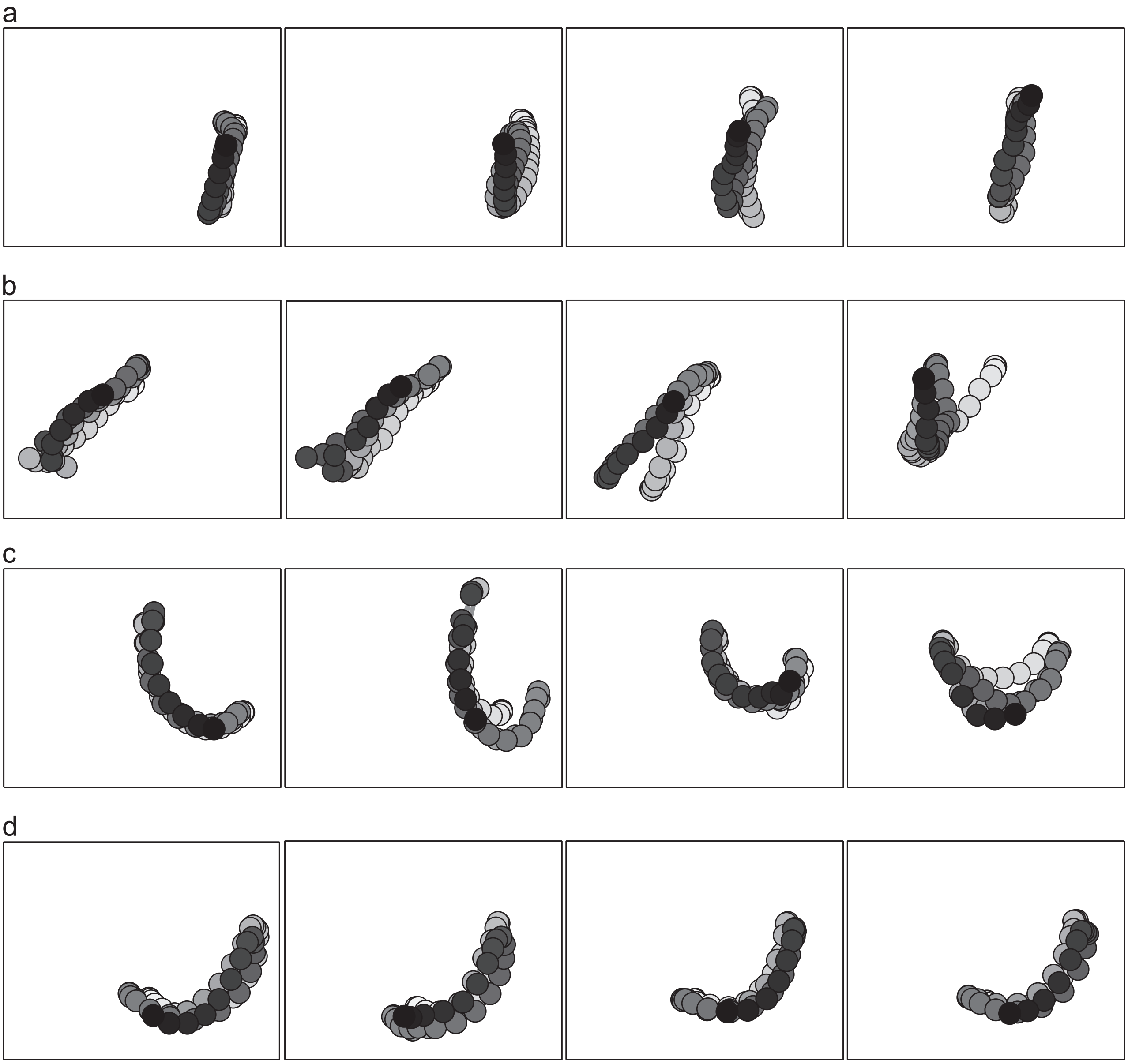


Fig. 11. Individual raw data examples of the sub-clusters in Fig. 10(f) of Libras Movement data. Each image illustrates hand movement traces, which start from a white color and end at a black color. (a) A sub-cluster in the yellow-colored class; (b) Another sub-cluster in the yellow-colored class; (c) A sub-cluster in the pink-colored class; (d) Another sub-cluster in the pink-colored class. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

which is often the case in document data. This aspect is also supported by our DB index results shown in Table 1. For instance, Reuters data set have a high mean DB value of 4.0 (Table 1), and although not shown here, two of its clusters have DB indices that are higher than 6.6. As a result, the result of t-SNE shows cluttered visualization without any distinguishable class structure. On the other hand, its supervised versions, LS t-SNE and ES t-SNE, show excessive separation between the classes, and thus fail to show the original high-dimensional relationships among individual documents. However, DS t-SNE generates balanced visualization between these two aspects, which allows users to properly understand the high-dimensional data structure.

Second, another major advantage of our method is the capability of revealing sub-clusters in the data, e.g., data groups belonging to a single class but distributed over multiple intrinsic clusters. For instance, two groups of sub-clusters (ellipses in Fig. 7(f)) are noticeably separated, while the corresponding classes in Fig. 7(a) are difficult to distinguish. Likewise, our method clearly reveals sub-clusters in other visualization examples. In Fig. 8, the orange-colored class does not show its sub-clusters in Fig. 8(a)–(e), but our method reveals its two groupings in Fig. 8(f). When we further examined the orange-colored class, which represents the written digit '1', these two sub-clusters indeed have clear distinction as one is vertically written (Fig. 9(a)) and the other is written with an angle (Fig. 9(b)).

Furthermore, we present an in-depth study on several data sets as follows. Fig. 10 shows visual results from Libras Movement data set, which is the Brazilian sign language data. In this data set, a class represents a particular hand gesture or movement. The visualization generated by DS t-SNE (Fig. 10(f)) reveals several sub-clusters in some of the classes. After further examination of the raw data, which correspond to movement trajectories, we found significant differences between various sub-clusters, as shown in Fig. 11. For example, Fig. 11(a, b) describes data points from the same class, but clearly the angles of the movements are shown to be different. Additionally, the samples shown in Fig. 11(c, d) also share the same class label, but the trajectories in one

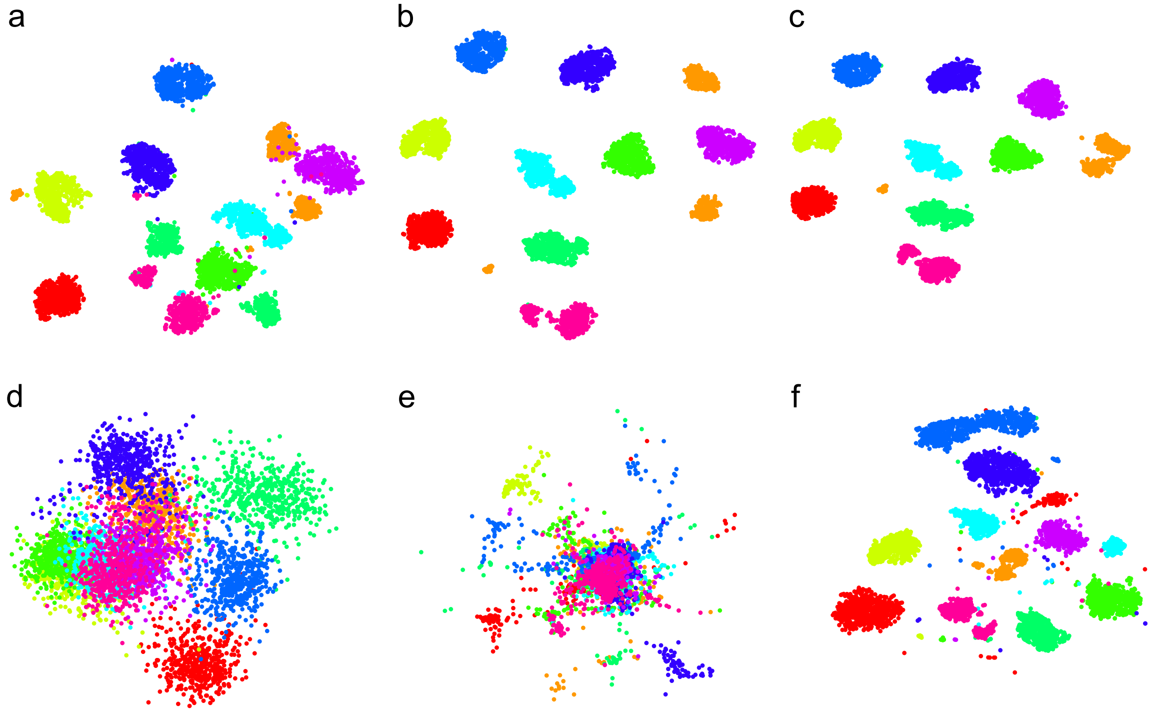


Fig. 12. Comparison of 2D visualizations of Handwritten Digits data using different algorithms. (a) t-SNE; (b) LS t-SNE; (c) ES t-SNE; (d) LDA; (e) S-NeRV; (f) DS t-SNE.

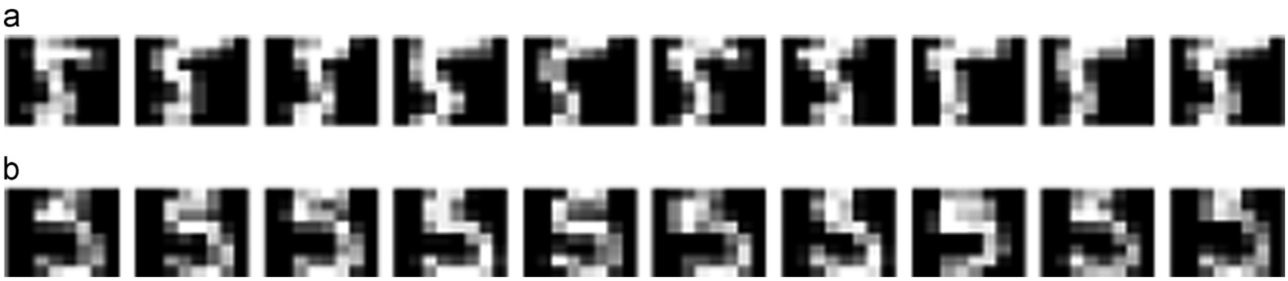


Fig. 13. Individual raw data examples of the sub-clusters in Fig. 12(f) of Handwritten Digits data. Each written digit is represented in a white color in a black background. (a) A sub-cluster in class '5'; (b) Another sub-cluster in the class '5'.

sub-cluster start movement from the right side (Fig. 11(c)) while those in another sub-cluster start from the bottom-left corner (Fig. 11(d)).

Fig. 12 shows visualization results from Optical Recognition of Handwritten Digits data set. In Fig. 12, the methods other than DS t-SNE represent class '5' (cyan-colored) as a single group, but our method shows it as roughly two separate sub-clusters. Fig. 13 shows individual raw data examples of the handwritten digit images corresponding to these cyan-colored sub-clusters in Fig. 12(f). The comparison between these examples from these sub-clusters indicates that the two sub-clusters differ in how people write the digit '5', e.g., whether the shapes of the bottom part are written big and round-shaped (Fig. 13(b)) or small and slim (Fig. 13(a)).

In many cases, LDA and S-NeRV fail to generate visually pleasing results in terms of class separability as well as the general point distribution in a 2D scatter plot. On the other hand, our method presents visually favorable low-dimensional embedding of data among the compared methods, facilitating both class-level or item-level analyses.

6. Conclusion

In this paper, we proposed a novel concept of double supervision in dimension reduction and presented its application to t-SNE, which we call doubly supervised t-SNE. Our double supervision is imposed on both the pre-given class information and the intrinsic clusters reflecting the natural grouping of the data. We demonstrated the advantage of the proposed method compared to other existing methods using both quantitative and qualitative analyses. As future work, we plan to apply our novel concept of double supervision to other popular dimension reduction methods. Furthermore, we plan to develop an interactive visualization system using the proposed method for high-dimensional data analysis [27,28].

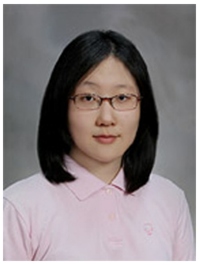
Acknowledgments

This work was supported in part by DARPA XDATA grant FA8750-12-2-0309 and NSF grants CCF-0808863, IIS-1242304, and IIS-1231742. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. We also thank anonymous reviewers for their insightful comments and suggestions.

References

[1] Jolliffe, Principal Component Analysis, Springer, New York, 2002.
[2] J. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1964) 1–27.

[3] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, San Diego, 1990.
[4] P. Howland, H. Park, Generalizing discriminant analysis using the generalized singular value decomposition, IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 26 (2004) 995–1006.
[5] C.H. Park, H. Park, A relationship between linear discriminant analysis and the generalized minimum squared error solution, SIAM J. Matrix Anal. Appl. (SIMAX) 27 (2005) 474–492.
[6] W. Zhao, R. Chellappa, A. Krishnaswamy, Discriminant analysis of principal components for face recognition, in: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 336–341.
[7] J. Choo, S. Bohn, H. Park, Two-stage framework for visualization of clustered high dimensional data, in: Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST), 2009, pp. 67–74.
[8] J. Choo, H. Lee, J. Kihm, H. Park, iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction, in: Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST), 2010, pp. 27–34.
[9] B. Shneiderman, The eyes have it: a task by data type taxonomy for information visualizations, in: Proceedings of the IEEE Symposium on Visual Languages, 1996, pp. 336–343.
[10] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. (JMLR) 9 (2008) 2579–2605.
[11] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[12] G.E. Hinton, S.T. Roweis, Stochastic neighbor embedding, in: Proceedings of the Advances in Neural Information Processing Systems, 2002, pp. 833–840.
[13] T. Kohonen, Self-Organizing Maps, Springer, Berlin, 2001.
[14] S. Kaski, J. Sinkkonen, J. Peltonen, Bankruptcy analysis with self-organizing maps in learning metrics, IEEE Trans. Neural Netw. (TNN) 12 (2001) 936–947.
[15] J. Venna, J. Peltonen, K. Nybo, H. Aidos, S. Kaski, Information retrieval perspective to nonlinear dimensionality reduction for data visualization, The J. Mach. Learn. Res. (JMLR) 11 (2010) 451–490.
[16] A. Gisbrecht, D. Hofmann, B. Hammer, Discriminative Dimensionality Reduction Mappings, Springer, Berlin, 2012.
[17] B. Hammer, A. Gisbrecht, A. Schulz, How to Visualize Large Data Sets? Springer, Berlin, 2013.
[18] M. Vlachos, C. Domeniconi, D. Gunopulos, G. Kollios, N. Koudas, Non-linear dimensionality reduction techniques for classification and visualization, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002, pp. 645–651.
[19] X. Geng, D.-C. Zhan, Z.-H. Zhou, Supervised nonlinear dimensionality reduction for visualization and classification, IEEE Trans. Syst. Man Cybern. B: Cybern. 35 (2005) 1098–1107.
[20] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[21] R. Bellman, Adaptive Control Processes: A Guided Tour, vol. 4, Princeton University Press, Princeton, 1961.
[22] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) PAMI 1 (1979) 224–227.
[23] J. Choo, C.K. Reddy, H. Lee, H. Park, p-ISOMAP: An efficient parametric update for isomap for visual analytics, in: Proceedings of the SIAM International Conference on Data Mining (SDM), 2010, pp. 502–513.
[24] A. Asuncion, D. Newman, University of California, Irvine, School of Information and Computer Sciences (2007). URL ⟨http://archive.ics.uci.edu/ml/⟩.
[25] E.M. Voorhees, et al., The trec-8 question answering track report., in: Proceedings of the Text REtrieval Conference (TREC), vol. 99, 1999, pp. 77–82.
[26] E. L. Lehmann, H. J. D'Abrera, Nonparametrics: Statistical Methods Based on Ranks, Springer, New York, 2006.
[27] J. Choo, H. Lee, Z. Liu, J. Stasko, H. Park, An interactive visual testbed system for dimension reduction and clustering of large-scale high-dimensional data, in: Proceedings of SPIE 8654, Visualization and Data Analysis (VDA), 2013, pp. 1–15.
[28] J. Choo, C. Lee, C.K. Reddy, H. Park, UTOPIAN: user-driven topic modeling based on interactive nonnegative matrix factorization, IEEE Trans. Vis. Comput. Graph. (TVCG) 19 (2013) 1992–2001.



Hannah Kim received the BS degree in mathematical sciences and the MS degree in accounting from the Seoul National University, Seoul, Korea. She is currently a Master's student in computer science at the Georgia Institute of Technology. Her research interests include data mining, machine learning, computational sustainability, and visualization.

Jaegul Choo is a Research Scientist at Georgia Institute of Technology. He received his Ph.D in Computational Science and Engineering from Georgia Institute of Technology in 2013. His research focuses on visual analytics for high-dimensional data, which leverages both data mining and interactive visualization. He was one of the four finalists in IEEE Visualization Pioneers Group dissertation award in 2013.

Chandan K. Reddy is an Associate Professor in the Department of Computer Science at Wayne State University. He received his Ph.D. from Cornell University and M.S. from Michigan State University. His primary research interests are data mining and machine learning with applications to healthcare, bioinformatics and social networks. His research is funded by NSF, NIH, DOT, and the Susan Komen for the Cure Foundation. He received the Best Application Paper Award at SIGKDD conference in 2010, and was finalist of the INFORMS Franz Edelman Award Competition in 2011. He is a senior member of IEEE and a member of ACM.

Haesun Park is a SIAM Fellow and professor in the School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Her research areas include numerical algorithms, data analysis, visual analytics, text mining, and parallel computing. She has played major leadership roles as the Executive Director of the Center for Data Analytics, Georgia Tech, general chair for the SIAM Conference on Data Mining, and editorial board member of SIAM and IEEE journals. She received a Ph.D. in Computer Science from Cornell University in 1987 and B.S. in Mathematics from Seoul National University with the University President's Medal.