A Perception-Driven Approach to Supervised Dimensionality Reduction for Visualization

Yunhai Wang, Kang Feng, Xiaowei Chu, Jian Zhang, Chi-Wing Fu,
Michael Sedlmair, Xiaohui Yu, and Baoquan Chen

Abstract—Dimensionality reduction (DR) is a common strategy for visual analysis of labeled high-dimensional data. Low-dimensional representations of the data help, for instance, to explore the class separability and the spatial distribution of the data. Widely-used unsupervised DR methods like PCA do not aim to maximize the class separation, while supervised DR methods like LDA often assume certain spatial distributions and do not take perceptual capabilities of humans into account. These issues make them ineffective for complicated class structures. Towards filling this gap, we present a *perception-driven linear dimensionality reduction* approach that maximizes the perceived class separation in projections. Our approach builds on recent developments in perception-based separation measures that have achieved good results in imitating human perception. We extend these measures to be density-aware and incorporate them into a customized simulated annealing algorithm, which can rapidly generate a near optimal DR projection. We demonstrate the effectiveness of our approach by comparing it to state-of-the-art DR methods on 93 datasets, using both quantitative measure and human judgments. We also provide case studies with class-imbalanced and unlabeled data.

Index Terms—Dimensionality reduction, supervised, visual class separation, high-dimensional data

1 Introduction

High-dimensional data is common in many application domains such as information retrieval, computational biology, and text mining. To visualize such data, dimensionality reduction (DR) is a common strategy to reduce the data dimensions while maintaining the data features of interest (e.g., covariance and correlation between the data attributes). The dimensionality reduced data can then be visualized, for instance, as scatter plots [1].

We focus on *labeled data*, that is, when a class label is assigned to each data item. Labeled data, for instance, is common in classification problems. For such data, "supervised" DR methods [2] can be used to find good low-dimensional data representations that seek to maximize the separation among classes. Inspecting how well classes separate in the resulting low-dimensional representations—usually by means of color-coded scatterplots—is then a very typical visualization task [3]. Since supervised DR methods take the class label into account, they usually better capture the class structures [4] than unsupervised DR methods like Principal Component Analysis (PCA) [5] and Multi-Dimensional Scaling (MDS) [6] (see Figs. 1a and 1b).

Many different supervised DR methods have been proposed over the last decades. One of the most popular ones is Fischer's venerable Linear Discriminate Analysis (LDA) [7]. However, LDA often fails to characterize nonlinear class structures because it assumes that each class follows a Gaussian distribution. Other approaches such as Kernel Discriminant Analysis (KDA) [8] have been proposed to overcome this shortcoming. Yet, none of the existing supervised DR approaches takes into account the perceptual capabilities of humans, and hence class structures might still remain hidden for a human observer (see Figs. 1c and 1d).

To fill this gap, we propose a *perception-driven dimensionality reduction* approach, that seeks to generate faithful, linear 2D projections with maximal visual class separation. To do so, we leverage and extend recent work on visual separation measures that imitate the human perception of class separation, and use these measures to drive the DR process. Sedlmair and Aupetit [9] evaluated existing measures in terms of their ability to imitate human perception, and found that the *Distance Consistency* (DSC) by Sips et al. [10] performed best, considerably better than LDA's measure. In a followup work [11], they proposed a variety of new visual separation measures, of which many outperformed all existing ones. Among the new ones, the *GONG* measure turned out to be the best. As it is computationally very expensive, the authors recommend the *KNNG* measure as a computationally more efficient alternative with almost the same separation capabilities as GONG. What is still unclear, however, is in how far these measures can effectively guide the DR procedure. Following Anand et al. [12], we could

- Y. Wang, K. Feng, X. Chu, X. Yu, and B. Chen are with Shandong University, Jinan City 250000, China. E-mail: {cloudseawang, falcon6130, cuxiaoxie, xhyu05, baoquan.chen}@gmail.com.
- C.-W. Fu is with the Chinese University of Hong Kong, Sha Tin, Hong Kong. E-mail: philip.chiwing.fu@gmail.com.
- J. Zhang is with CNIC, CAS, Beijing 100190, China. E-mail: zhangjian@sccas.cn.
- M. Sedlmair is with the University of Vienna, Wien 1010, Austria. E-mail: michael.sedlmair@univie.ac.at.

Manuscript received 20 Jan. 2017; revised 2 May 2017; accepted 2 May 2017. Date of publication 5 May 2017; date of current version 28 Mar. 2018.
(Corresponding Author: Yunhai Wang)
Recommended for acceptance by C.D. Hansen.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2017.2701829

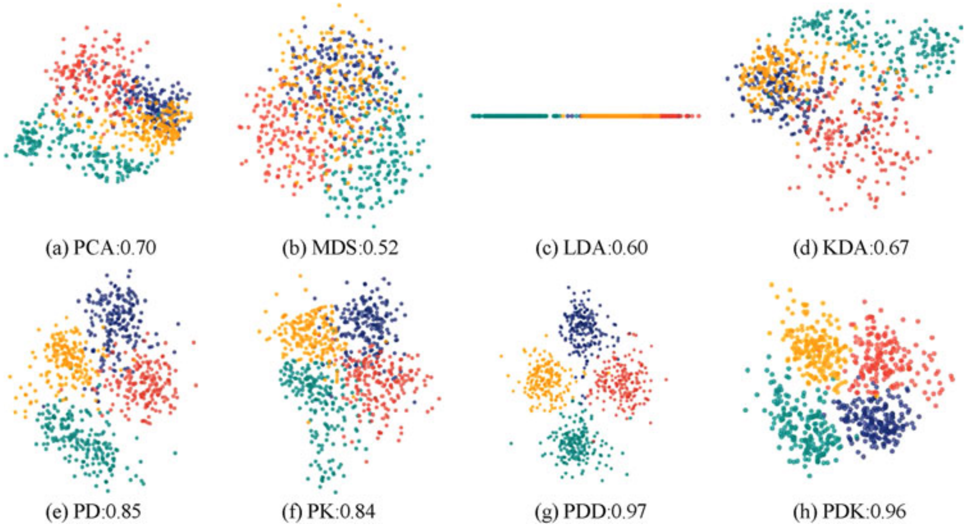


Fig. 1. Comparing the performance of different DR methods (first row) and our proposed methods (second row) in visualizing a 64-dimensional dataset with four classes shown in different colors: (a) PCA; (b) MDS; (c) LDA; (d) KDA; (e) our perception-driven DR with DSC (PD); (f) ours with KNNG (PK); (g) ours with density-aware DSC (PDD); (h) ours with density-aware KNNG (PDK). We can see that the four classes are mostly mixed together in the visualizations produced from the existing methods (first row), though KDA can roughly separate the red and cyan classes. Our methods (second row) produce visualizations with clearer visual class separation, specifically for PDD and PDK, which use the new density-aware measures. The GONG scores (see the numbers above) quantitatively confirm this result. Note that (a) and (b) are unsupervised DR methods, while (c)-(h) are supervised, that is, taking class labels into account.

find interesting projections by randomly generating many subspaces and ranking these subspaces by the scores of separation measures. However, such random exploration may not identify the optimal DR results.

We propose a different approach, *perception-driven DR*, which is based on customizing a simulated annealing optimization algorithm [13]. We formulate the perceptual measures into a linear DR approach and seek a $2 \times d$ projection matrix (d is the data dimensionality) that maximizes the visual class separation. We then use the simulated annealing approach to rapidly and efficiently find a projection that is close to the global maximum in terms of the separation. As for the separation measures, we use DSC and KNNG, which have been found to be among the best state-of-the-art measures, but are still computationally efficient enough for our purpose. The results of directly using these measures in our approach are shown in Figs. 1e and 1f. When designing our perception-driven DR process, however, we found that current measures lack the ability to properly model the density of classes. Yet, density has been found to be crucial for the formation of class structures in the DR process [14], which is one of our goals. To overcome these issues, we devised two new measures, density-aware DSC (dDSC) and density-aware KNNG (dKNNG), by incorporating the distances between points into the measures. Using these two new measures, we arrived at two new linear DR methods, which we call Perception-driven DR using Density-aware DSC (*PDD*), and Perception-driven DR using Density-aware KNNG (*PDK*). Their results are shown in Figs. 1g and 1h).

We evaluated our approach on 93 datasets [1], comparing it to six existing DR methods: PCA [5], LDA [7], KDA [8], NCA [15], t-SNE [16], and random projections [12]. We quantitatively compared our perception-driven DR methods to the others using the GONG and Silhouette Coefficient (SC) measures. We found that, for most datasets, our methods were capable of producing the best class separation scores, using less time than the other methods. We also ran a human judgment study similar to the one by Sedlmair et al. [1]. The results of this study confirmed the capability of our methods to produce results that better align with the perceptual judgments of humans.

We furthermore presented two extensions of our methods. First, we showed how they can support the exploration of *class-imbalanced data* by normalizing dDSC and dKNNG according to the number of points belonging to each class. Second, we illustrated how they can be used to explore *unlabeled data* as high-dimensional data might not necessarily come with labels.

In summary, the main contributions of this paper include:

- We propose a perception-driven linear dimensionality reduction approach, which is based on (i) extending state-of-the-art visual separation measures, (ii) formulating them into a DR pipeline, and (iii) devising a customized simulated annealing optimization to rapidly generate the DR projection (Section 4).
- We evaluate the resulting DR methods, PDD and PDK, by comparing them to six state-of-the-art DR methods on 93 datasets (Section 5).
- We present two extensions that show how our methods can be used to explore class-imbalanced and unlabeled data (Section 6).

The rest of the paper is organized as follows. We summarize related work in Section 2, before we briefly introduce formal definitions on linear DR and visual separation measures in Section 3. In Section 4, we present our perception-driven DR approaches, including density-aware visual separation measures and our DR-involved optimization via simulated annealing. We evaluate our approaches in Section 5 and describe two extensions in Section 6, followed by conclusion and limitations in Section 7.

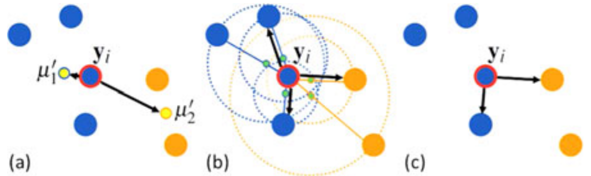


Fig. 2. We illustrate the construction of nearest neighbor graphs in a two-class data for (a) DSC, (b) GONG, and (c) KNNG by taking one of the data points (circled in red) as an example. The black arrows in the illustrations show how the data point connects with its neighbors in each nearest neighbor graph; however, the three measures define neighbors differently. For DSC, neighbors are class centers (see Eqs. (4) & (5)), see the two yellow dots in (a). For GONG and KNNG, neighbors are nearby data points; the small green dots in (b) are the intermediary points in-between the data point and its neighbors.

2 Related Work

Existing related work can be divided into two categories: visual class separation measures and dimensionality reduction techniques.

2.1 Visual Class Separation Measures

Since the projection pursuit indices proposed by Friedman and Tukey [17], many visual quality measures have been developed for different purposes in visualization. Bertini et al. [18] conducted a systematic analysis of various quality measures and found that it is necessary to consider models of human perception in the visualization design. This work follows the same principle, where we aim to model the human perception of class separability into supervised DR methods.

In terms of class separability, Sedlmair et al. [19] presented a taxonomy of factors that influence the human perception of visual class separation. By combining different factors together, many class separation measures have been proposed. The Class Density and Histogram Density measures [20], for instance, are both based on class density, while the Distribution Consistency [10] looks at the class labels of the nearest data points. Rather than considering the local structure, the afore-mentioned Distance Consistency [10] computes the separation degree of each point by comparing its own class center to the nearest class center. Aupetit and Sedlmair [11] unified all these measures into a framework by factorizing them into two components: neighborhood graphs and class purity functions. For example, the graph of DSC is created by connecting each data point with its own class center and the nearest class center (see Fig. 2a), and its class purity function is a binary comparison of the distances to the two centers. Under this framework, Aupetit and Sedlmair proposed 2002 new measures [11] by defining 143 different graphs and 14 different class purity functions.

Besides them, the machine learning community also proposed several different measures. Classical examples include Dunn's index [21], LDA's objective [7], and the Silhouette Index [22]. Their main difference is between- and within-class distances [14] defined on different graphs. The first two measures construct the neighborhood graph by connecting all pairwise nodes in the same class, while the last measure considers also the pairwise distances between all the nodes, which is a complete graph.

To learn how these measures model human perception, Sedlmair and Aupetit [9] proposed a machine learning framework to evaluate all the measures and found that DSC is the best one. Recently, they further evaluated their proposed new measures [11] and found that their proposed *0.35-Observable Neighbors of each point of the target class* (GONG) performs the best, much better as compared to DSC. Meanwhile, they found the *average Class-Proportion of the two-Nearest-Neighbors of each point in the target class* (KNNG) also performs much better than DSC, while having lower computational cost compared to GONG. In this work, we further extend KNNG and DSC with the ability to capture density information, so they can be properly used in an iterative DR process (see Section 4.1).

2.2 Dimensionality Reduction Techniques

Dimensionality reduction methods can be categorized into *linear* and *non-linear* methods. Linear DR methods project the data to a lower dimensional space by a linear transformation. The benefit of this approach is that it is relatively easy to interpret and understand, and fast to compute. A variety of linear methods [2] have been developed to preserve different data features of interest. Among them, PCA and LDA are two very popular methods due to their simplicity and computational efficiency. PCA maximizes the data variance captured by the low-dimensional projection, while LDA maximizes the separation of classes in the labeled data. The latter resembles our goal. To combine their advantages, Choo et al. [23], [24] proposed a two-stage framework for visualizing labeled data, where LDA is first used to obtain a cluster-preserved low-dimensional data, and PCA is then applied to further reduce the dimension to two for visualization. Many other linear methods exist: Locality Preserving Projection (LPP) [25], for instance, aims to preserve the neighborhood structure of the data; or, Neighborhood Components Analysis (NCA) [15], which has the same goal as LDA and out methods, that is, separating labeled classes. As opposed to LDA, NCA makes no assumptions on the shape of the class distributions though and thus overcomes some intrinsic shortcomings of LDA, which assumes that the classes follow the Gaussian distributions. Our methods PDD and PDK fall into the same category: linear, supervised DR methods that make no assumptions on the class shapes. However, we go beyond the state-of-the-art by modeling the human perception into the DR process to optimize the visualization efficiency.

While linear methods are computationally efficient and relatively easy to understand, they might miss non-linear structures in the data. Multi-dimensional scaling is a widely-used DR method that can be used in a non-linear fashion. It attempts to preserve the dissimilarities between high-dimensional data points in the low-dimensional space. Since it involves an $O(n^2)$ optimization, many methods have been proposed [26] to accelerate its computation. The Kernel Discriminant Analysis [8] is a non-linear variant of LDA that seeks to separate labeled classes by using kernel functions. However, it is computationally very expensive and may not always capture complicated class structures.

Other methods seek to better preserve local neighborhood structures, and to uncover the intrinsic structure in the data by building a model of manifold connectivity. Examples include Isomap [27], Locally Linear Embedding [28], Laplacian Eigenmaps [29], and many other variants. Unfortunately, these methods are computationally very intensive and heavily rely on the constructed neighborhood graph. By transforming distances between data points to probabilities, t-SNE [16] produces visualizations with well-separated classes of high-dimensional data. Like t-SNE, our metric also generalizes the simple binary separation degree into continuous measures, but our new measures achieve better results with less computation, as shown in Section 5.

The above methods are based on measures defined in the data space. Human perception is not considered as a first class citizen in their formulations. To investigate which projections are preferred by humans, Lewis et al. [30] conducted a user study and found that experts are reasonably consistent in their preferences, while novices generally seem not to have shared preferences on projections. Sedlmair et al. [1] and Etemadpour et al. [31] both investigated the effects of user tasks and data characteristics on the DR process, such as visual class separability. Aligned with their observations, interesting low-dimensional projections can also be found by using visual quality measures. Projection pursuit [17] identifies interesting projections by scoring each projection with the projection pursuit indices and presenting the top-ranked results to user. Anand et al. [12] extended this method to high-dimensional space by using binning and random projections. Both methods focus on unlabeled data and thus cannot ensure that the classes are visually well-separated under the projections they found. As an alternative, state-of-the-art separation measures, such as DSC and KNNG, could be used in these frameworks. However, such random projection cannot guarantee the identification of DR results with maximal class separation. Instead, we formulate an optimization problem that makes use of the class separation measure to guide the DR process, where we seek the projection that leads to the maximal visual separation through our customized simulated annealing algorithm.

3 Preliminaries: Formal Definitions

In this section, we provide formal definitions of the components used in the design and evaluation of our approach. We first formally introduce the approach of linear dimensionality reduction. We then describe three state-of-the-art visual separation measures: DSC, GONG, and KNNG.

3.1 Linear Dimensionality Reduction

Given n d-dimensional data points $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{d\times n}$, linear DR methods aim to optimize an objective function $f(\mathbf{X})$ to produce a linear transformation $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_d\} \in \mathbb{R}^{l\times d}$, which is a projection, to map $\mathbf{x}_i$ in d dimensions to $\mathbf{y}_i$ in l (lower) dimensions: $\mathbf{y}_i = \mathbf{P}\ \mathbf{x}_i$. We denote the set of resulting points as $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\} \in \mathbb{R}^{l\times n}$.

To capture different data features of interest, many different objectives $f(\mathbf{X})$ have been designed, e.g., PCA attempts to find $\mathbf{P}$ that minimizes the reconstruction error of the projected data

$$f(\mathbf{X}) \ = \ \|\ \mathbf{X} \ - \ \mathbf{P}^T\mathbf{P}\mathbf{X}\ \|_F^2 \ , \tag{1}$$

where F denotes the Frobenius norm.

Suppose $\mathbf{X}$ is partitioned into C classes; each $\mathbf{x}_i$ is assigned a label $l(\mathbf{x}_i)$, and the cth class (with n_c data points) is $\{\mathbf{x}_1^c, \ldots, \mathbf{x}_{n_c}^c\}$, $c \in \{1, \ldots, C\}$. LDA aims to maximize the class separation in $(C-1)$-dimensional subspace by the following objective:

$$\max \ \mathrm{tr}\ \frac{\mathbf{P}^T\mathbf{S}_b\mathbf{P}}{\mathbf{P}^T\mathbf{S}_w\mathbf{P}} \ , \tag{2}$$

where $\mathbf{S}_b$ and $\mathbf{S}_w$ are the between- and within-cluster scatter, resp.

$$\mathbf{S}_b = \sum_{c=1}^{C} n_c(\mu_c - \mu)(\mu_c - \mu)^T \text{and } \mathbf{S}_w = \sum_{c=1}^{C}\sum_{i}^{n_c}(\mathbf{x}_i^c - \mu_c)(\mathbf{x}_i^c - \mu_c)^T, \tag{3}$$

where $\mu_c = \sum \mathbf{x}_i^c / n_c$ is the mean (centroid of data points) of the cth class and μ is the mean (centroid) of the entire dataset $\mathbf{X}$.

3.2 State-of-the-Art Visual Separation Measures

DSC [10] is a visual separation measure found to outperform a number of common measures [9], such as Dunn's Index [21], Distribution Consistency [10], and Class Density Measure [20]. The key idea of DSC is to compute the separation degree $\mathbf{s}(\mathbf{y}_i)$ of each point $\mathbf{y}_i$ by comparing the within-class distance $a(\mathbf{y}_i)$ and between-class distances $b(\mathbf{y}_i)$ in the 2D visual space after projection by $\mathbf{P}$

$$a(\mathbf{y}_i) \ = \ dist(\mathbf{y}_i, \mu'_c), \tag{4}$$

$$b(\mathbf{y}_i) \ = \min_{j\in\{1,\ldots,C\}, j\neq c} dist(\mathbf{y}_i, \mu'_j), \tag{5}$$

$$\mathbf{s}(\mathbf{y}_i) \ = \ \delta(a(\mathbf{y}_i) \ > \ b(\mathbf{y}_i)), \tag{6}$$

where $\mu'_j = \mathbf{P}\mu_j$, $a(\mathbf{y}_i)$ is the distance from $\mathbf{y}_i$ to its own class center (i.e., μ'_c), $b(\mathbf{y}_i)$ is the distance from $\mathbf{y}_i$ to another class center (which is the nearest), and $\delta(\cdot)$ is an indicator function: if $a(\mathbf{y}_i) \ > \ b(\mathbf{y}_i)$, $\mathbf{s}(\mathbf{y}_i)$ is one, else zero. Hence, each point $(\mathbf{y}_i)$ connects to two class centers according to Eqs. (4) and (5), and all these connections together form the nearest neighborhood graph, see Fig. 2a for an example. The final DSC value is the average of all $\mathbf{s}(\mathbf{y}_i)$, which in fact indicates the average classification error after the projection. Since the time complexity to find the nearest center for $b(\mathbf{y}_i)$ is $O(\log C)$, the overall time complexity to compute DSC is $O(n\log C)$.

GONG 0.35 DIR CPT (abbrev. as GONG). Aupetit and Sedlmair [11] proposed this measure and found it to be the best state-of-the-art measure: 11.7 percent better than DSC in terms of AUC, i.e., Area Under the Receiver Operating Characteristic curve. GONG is built upon the γ-observable neighbor graph: for each point $\mathbf{y}_i$, connect it to point $\mathbf{y}_j$ if the intermediary point in-between them (computed by $\gamma\mathbf{y}_j + (1-\gamma)\mathbf{y}_i$ with $\gamma = 0.35$) is closer to $\mathbf{y}_j$ than any point in $\mathbf{Y}\backslash\{\mathbf{y}_i\}$. In the example shown in Fig. 2b, three of the five intermediary points (the small green dots) satisfy this condition (see the black arrows), so their corresponding $\mathbf{y}_j$'s form a set denoted as $\Omega(\mathbf{y}_i)$, with which we can compute the separation degree of $\mathbf{y}_i$

$$\mathbf{s}(\mathbf{y}_i) \ = \frac{1}{|\Omega(\mathbf{y}_i)|} \sum_{\mathbf{y}_j\in\Omega(\mathbf{y}_i)} \delta(l(\mathbf{y}_i), l(\mathbf{y}_j)) \ , \tag{7}$$

where $\delta(l(\mathbf{y}_i), l(\mathbf{y}_j))$ is one if $\mathbf{y}_i$ and $\mathbf{y}_j$ have the same class label, else zero. For the example in Fig. 2b, $|\Omega(\mathbf{y}_i)| = 3$ and $\mathbf{s}(\mathbf{y}_i)$ is 2/3. The final GONG value is the average separation degree over all data points of the same class rather than over the entire dataset. Since finding $\Omega(\mathbf{y}_i)$ for a data point requires us to test its nearby data points in $\mathbf{Y}\backslash\{\mathbf{y}_i\}$, the overall time complexity is $O(n^2 \log n)$, which would be too high for DR-involved optimization.

KNNG 2 DIR CPT (abbrev. as KNNG) also uses Eq. (7) to compute the separation degree of each point in the dataset, but it simply takes the two nearest data points of $\mathbf{y}_i$ as $\Omega(\mathbf{y}_i)$, so $\mathbf{s}(\mathbf{y}_i)$ is always 0, 0.5, or 1. Hence, although the AUC score for KNNG is empirically slightly less than that of GONG, creating a two-nearest neighborhood graph only costs $O(2n\log(n))$, so its computation is far less expensive than creating the γ-observable neighbor graph for GONG. In the example shown in Fig. 2c, the separation degree of the point in red circle is 0.5.

4 Perception-Driven DR

The visual separation measures described above provide the foundation for evaluating the perceptual quality of 2D projections. In this work, one of our goals is to reformulate a linear DR optimization with such measures, so that we can seek a perception-driven DR projection that maximizes the visual class separation.

To achieve this goal, we first derive an energy function that maximizes the separation degree with importance parameters w_i

$$E(\mathbf{P}) = \arg\max_{\mathbf{P}} \frac{1}{n}\sum_{i=1}^{n} w_i\, \mathbf{s}(\mathbf{y}_i), \tag{8}$$

where $\mathbf{P}$ is a $2 \times d$ projection matrix, and w_i is a user-defined parameter for each data point with one as its default value.

Finding the global optimum for Eq. (8) is nontrivial because the objective function is non-linear and non-analytical. Moreover, gradient-descent-based optimization algorithms might deliver inferior results, since the indicator functions incorporated in Eqs. (6) and (7) are non-differentiable [32].

Even if we may solve the optimization using some complex algorithms such as genetic algorithms [33], directly using the state-of-the-art class separation measures might not generate desirable results due to their inherent drawbacks. First, current measures have not yet considered the class density distribution, which could affect the class structure [14] and impact the DR results. Second, evaluations of existing measures have been conducted only on scatter plots with binary classes [11]. It is not clear how well they work for multi-class scatter plots. Last, GONG has a very high computation cost in $O(n^2\log(n))$, which hampers the visual analysis process in many scenarios, e.g., interactive high-dimensional data exploration.

Due to the drawbacks mentioned above, DR methods directly driven by DSC or KNNG may not guarantee desirable results, as illustrated in Figs. 1e and 1f: the blue, yellow, and red classes are not well-separated. To overcome these issues, we propose two techniques: *density-aware separation measures* for multi-class data (Section 4.1) and a customized *simulated annealing* algorithm (Section 4.2). The density-aware formulations of DSC and KNNG are referred to as dDSC and dKNNG, respectively, and their corresponding perception-driven DR methods are pDR-dDSC and pDR-dKNNG (abbrev. as PDD and PDK).

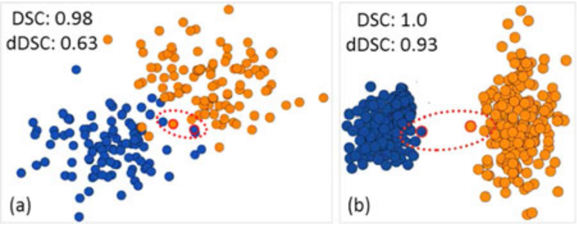


Fig. 3. The deficiency of DSC in characterizing the separation degree of points in two different scatterplots. The DSC values of the plots are similar, but our dDSC measure can catch their difference.

4.1 Density-Aware Visual Separation

4.1.1 Density-Aware DSC

To overcome the deficiency of DSC in characterizing class density and distribution, we formulate *density-aware DSC*, which defines the separation degree as the difference between within-class and between-class distances normalized by their maximum

$$\mathbf{s}(\mathbf{y}_i) = \frac{b(\mathbf{y}_i) - a(\mathbf{y}_i)}{\max\{a(\mathbf{y}_i), b(\mathbf{y}_i)\}}\ , \tag{9}$$

where $a(\mathbf{y}_i)$ and $b(\mathbf{y}_i)$ are defined as in Eqs. (4) and (5), so $\mathbf{s}(\mathbf{y}_i) \in [-1, 1]$. When $\mathbf{s}(\mathbf{y}_i)$ is close to 1, $\mathbf{y}_i$ is located close to the center of its class; when $\mathbf{s}(\mathbf{y}_i)$ is close to 0, $\mathbf{y}_i$ is located around the boundary of its class; and when $\mathbf{s}(\mathbf{y}_i)$ is close to -1, $\mathbf{y}_i$ is likely to be misclassified. Therefore, this new measure can take into account the class density and distribution, and provide a continuous separation degree. If data points are mostly concentrated around the associated class centers, the dDSC value is high; otherwise, it is lower. Fig. 3 shows an example that illustrates the capability of dDSC over DSC and shows that it is much more sensitive for different degrees of visual separability. The time complexity of computing dDSC is the same as DSC; both are $O(Cn)$, where C is the number of classes. This is very efficient and hence good for our purpose of using it in an iterative DR pipeline.

4.1.2 Density-Aware KNNG

Since KNNG only considers the class labels of two nearest points (see Section 3.2), for points near the class boundaries, their separation degrees are always 0.5 if one of their nearest points has the same label while the other one is different. This is true no matter how close or far the point is from the nearest point of the same label. Such points are often located near the class boundaries, where the labeling may not be reliable; see Fig. 4a for examples.

To better characterize the separation degree, we propose density-aware KNNG. Given $\mathbf{y}_j$ and $\mathbf{y}_k$ as the two nearest neighbors of $\mathbf{y}_i$, there are three cases in computing the separation degree of $\mathbf{y}_i$ (i.e., $\mathbf{s}(\mathbf{y}_i)$) by comparing labels $l(\mathbf{y}_j)$ and $l(\mathbf{y}_k)$ against $l(\mathbf{y}_i)$:

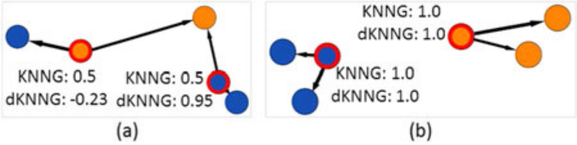


Fig. 4. The deficiency of KNNG in characterizing the separation degree. Here, we look into the class boundaries in Fig. 3 (see the red ellipses), and take as examples the four points circled in red above. (a) the KNNG values of the two points are the same, while our dKNNG can catch their difference, and (b) when a point and its neighbors have the same label, our dKNNG can behave consistently.

- If both labels $l(\mathbf{y}_j)$ and $l(\mathbf{y}_k)$ equal $l(\mathbf{y}_i)$, set $\mathbf{s}(\mathbf{y}_i)$ as 1.
- If both labels $l(\mathbf{y}_j)$ and $l(\mathbf{y}_k)$ differ from $l(\mathbf{y}_i)$, set $\mathbf{s}(\mathbf{y}_i)$ as -1.
- Without loss of generality, we consider $l(\mathbf{y}_i) = l(\mathbf{y}_j)$ and $l(\mathbf{y}_i) \neq l(\mathbf{y}_k)$. Then, we can compute $a(\mathbf{y}_i) = \mathrm{dist}(\mathbf{y}_i, \mathbf{y}_j)$ (same label) and $b(\mathbf{y}_i) = \mathrm{dist}(\mathbf{y}_i, \mathbf{y}_k)$ (different label), and compute $\mathbf{s}(\mathbf{y}_i)$ using Eq. (9).

Hence, $\mathbf{s}(\mathbf{y}_i) \in [-1, 1]$, which is consistent with dDSC, and the time complexity of computing dKNNG is the same as KNNG; both are $O(2n\log n)$.

4.1.3 Discussion

By incorporating distances into the class separation measures, both dDSC and dKNNG can better characterize the class density and distribution. However, they have slightly different characteristics: dDSC describes how dense the points around the class center are, such that we can effectively differentiate the two plots in Figs. 3a and 3b with different dDSC values. In contrast, dKNNG describes how dense the points with the same label are. Since the points around the class center often have the same label (see Fig. 4b), dKNNG focuses on the characterization of boundary points as shown in Fig. 4a. Since both class densities and class boundaries play important roles in determining the class structures [14], both dDSC and dKNNG can be used to guide the iterative DR procedure in the optimization, depending on the data separation pattern (see Section 5).

To explore whether our new measures are in accordance with human perception, we employ the evaluation framework proposed by Sedlmair and Aupetit [9]. This framework is based on predicting human judgements on 828 scatterplots using the AUC (area-under-the-curve) to score the measures: 50 percent indicates a random guess, while 100 percent indicates a perfect prediction of human judgements. The AUC scores of dDSC and DSC are 83.1 and 83.2 percent, respectively, while the scores of both dKNNG and KNNG are 92.1 percent. This suggests that our new measures dDSC and dKNNG are comparable to DSC and KNNG in terms of reflecting human perception. This result is not surprising as the framework [9] involves only clearly separable classes and clearly non-separable classes. While we want to ensure that our new measures are equally good in these clear-cut cases, our primary goal is to address cases that may not be that clear. This is crucial for the DR-involved optimization, where the class structures are unknown in the early iterations as shown in Fig. 7.

4.2 Optimization by Simulated Annealing

Simulated Annealing (SA) [13] is a stochastic optimization method inspired by the annealing process in metallurgy. In general, it solves an optimization problem by beginning with a high "temperature", gradually lowering the temperature with the Metropolis criterion [34] until a good solution is found. The "temperature" can be thought of as the probability of accepting intermediate results that are worse than the current iteration. The lower the temperature the lower the probability that worse results will be accepted. Despite the many local optimums, the global optimum is very likely to be found as long as the temperature cooling speed is sufficiently slow.

To optimize the non-linear objective function presented in Eq. (8), we customize a Simulated Annealing (SA) method [13] as outlined in Algorithm 1. It has two key components: i) initialization of $\mathbf{P}$; and ii) selection of neighbor solution $\mathbf{Q}$ from $\mathbf{P}$. Note that we follow the suggestion from Kirkpatrick et al. [35] and set the cooling coefficient α to be 0.95 in all our experiments. In addition, we set the initial T as $100 \times d$, where d is the number of dimensions in the input data.

Algorithm 1. Optimal Projection Matrix Approximation

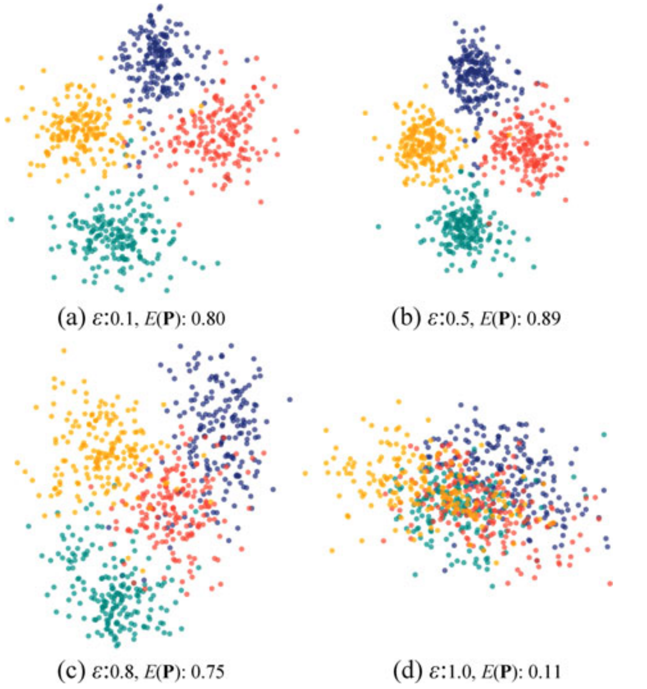
```
1: set an initial solution $\mathbf{P}$
2: set an initial temperature $T$
3: while $T \neq 0$ do
4:   randomly choose $\mathbf{Q}$ in the neighborhood of $\mathbf{P}$
5:   $\Delta E = E(\mathbf{Q}) - E(\mathbf{P})$
6:   if $\Delta E > 0$ then
7:     $\mathbf{P} = \mathbf{Q}$
8:   else if $prob(\exp(\Delta E/T)) > random(0,1)$ then
9:     $\mathbf{P} = \mathbf{Q}$
10:  end if
11:  reduce temperature by $T = \alpha T$
12: end while
```

Initialization of $\mathbf{P}$ *(line 1).* For most data, we find that using random initialization can quickly produce reasonable results. Alternatively, one can use any existing DR algorithm as an initialization. When comparing the SA-optimized results generated using random initialization, PCA, LDA, and LPP, respectively, to initialize $\mathbf{P}$, however, we found that the results are almost the same. As random initialization sometimes generated slightly better results, we opted for random initialization as the default option in our methods.

Choosing a Neighbor Solution $\mathbf{Q}$ *Near* $\mathbf{P}$ *(Line 4).* By linearizing $\mathbf{P}$ as a vector $\{p_1, p_2, \ldots, p_{2\times d}\}$, we may produce $\mathbf{Q}$ by adding a small random offset to each p_i in $\mathbf{P}$. This is a simple and fast approach to generate a random $\mathbf{Q}$, but we found that it does not necessarily achieve high-quality results with a fixed cooling coefficient α.

To rapidly produce better solutions, we pick some p_i with random probabilities and search for a local optimum. Specifically, we replace p_i with $p_i * (1 - \Delta r)$ or $p_i * (1 + \Delta r)$, and assign the one that can lead to a larger improvement of $E(\mathbf{P})$ to the corresponding element in $\mathbf{Q}$. To guarantee that the local optimum is close to $\mathbf{P}$, we set Δr to 0.05. Regarding the unselected variables, their new values are still generated by adding a random (positive or negative) offset with the magnitude 0.01.

The selection of which p_i to modify is determined by a random process, where we draw a random value uniformly in $[0, 1]$ and check if it is greater than threshold ϵ. If ϵ is small, many local optima need to be computed while the



(a) ε:0.1, $E(\mathbf{P})$: 0.80

(b) ε:0.5, $E(\mathbf{P})$: 0.89

(c) ε:0.8, $E(\mathbf{P})$: 0.75

(d) ε:1.0, $E(\mathbf{P})$: 0.11

Fig. 5. These four results (a-d) are produced using different values of ϵ: 0.1, 0.5, 0.8, and 1.0. Their final energies $E(\mathbf{P})$ are 0.80, 0.89, 0.75 and 0.11, respectively, while the running time are 9.94 s, 2.40 s, 0.81 s and 0.04 s, respectively.

final solution is not good enough due to insufficient randomness (see Fig. 5a), but if ϵ is too large, the solution is not good if the number of iterations is not sufficient. Note that when $\epsilon = 1$, it is equivalent to generating fully randomized neighborhoods (all p_i); see Fig. 5d. Thus, finding a proper ϵ is very important to the final result. To do so, we empirically tested the value of ϵ over the 93 datasets and found that a choice of 0.5 works well for most data.

Time Complexity. In the worst case ($\epsilon = 0$), where we need to compute the local optimum for every dimension, the number of times that we need to evaluate the objective function is $4dm$, where m is the total number of iterations in the SA process. For a dataset with C classes, the time complexities of computing dDSC and dKNNG are $O(Cn)$ and $O(2n\log(n))$, respectively. Hence, the time complexities of PDD and PDK (abbrev. of perception-driven DR methods with dDSC and dKNNG) are $O(4Cdnm)$ and $O(8dn\log(n)m)$, respectively.

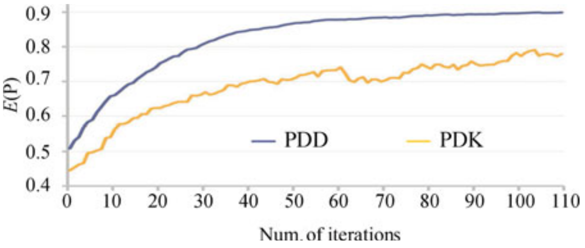


Fig. 6. The plots on $E(\mathbf{P})$ versus the number of iterations showing the convergence of the two proposed methods. PDD gradually reaches convergence, while PDK quickly reaches a reasonable solution but then has a slow rise with oscillations. Note that as the objectives of these methods are different, the difference in convergence is reasonable.

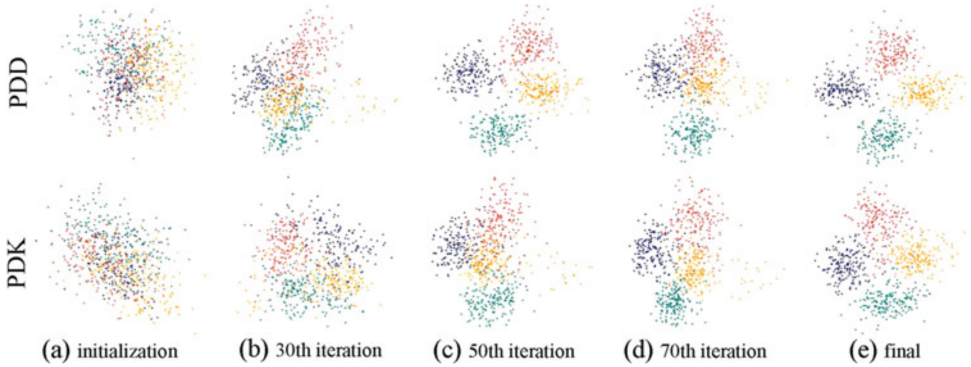
Since $C << n$, we can simplify the time complexities of PDD and PDK as $O(dmn)$ and $O(dmn\log(n))$, respectively. In our experiments, we found that 100 iterations are enough for most data, namely $m \leq 100$. This indicates that our methods (especially PDD) have advantages in computation cost even when compared with linear methods, like PCA and LDA (see Section 5 for details).

Fig. 6 shows the convergence curves, where we can see that PDD gradually converges, while PDK quickly arrives at a reasonable solution but then slowly oscillates until it converges. This difference is not surprising as the objective function of dDSC is smoother than that of dKNNG. Fig. 7 illustrates these difference by showing and comparing the intermediate results generated by these two methods.

5 Comparative Evaluation

We implemented our methods in C++ and tested them on a PC with an Intel Core i5-4590 3.3 GHz CPU and 8 GB memory. To confirm that our algorithms (PDD and PDK) can present the class structures of labeled high-dimensional data with maximal visual class separation, we compared their projection quality with six widely-used DR methods by performing two comparisons: (i) with numerical measures (Section 5.1) and (ii) based on human judgements (Section 5.2).

Existing DR Methods. The comparison includes six existing DR methods: PCA, t-SNE [16], LDA, KDA [8], neighborhood components analysis (NCA) [15], and the random projection method (RAND) [12]. We use GONG (see Section 3.2) as the score function in RAND method to choose the projection with the maximal class separation. Since GONG takes the class labels into account, we take RAND as a supervised method. PCA and t-SNE are the only unsupervised methods



(a) initialization

(b) 30th iteration

(c) 50th iteration

(d) 70th iteration

(e) final

Fig. 7. Illustration of convergence, where the top and bottom rows show the results of PDD and PDK, respectively. (a) results of the random initializations; (b) results after 30 iterations; (c) results after 50 iterations; (d) results after 70 iterations; and (e) final results.

TABLE 1
Datasets Used in Our Comparative Evaluation

Dataset Category	#sets	#points	#dim.	#classes
REAL	53	24 – 43,500	3 – 295	2 – 28
SYNTHETIC-GAUSSIANS	16	100 – 500	5 – 10	3 – 5
SYNTHETIC-ENTANGLED	24	185 – 2,318	3 – 15	3 – 15

in this set, that is, DR methods that do not directly take class labels into account. While this introduces a natural bias towards the supervised methods, we still deem a comparison interesting and relevant. PCA is a very well-known method and has been used widely in the visualization community. t-SNE—although not taking class labels into account—has been shown to still perform good on visual class separability tasks, specifically for high-dimensional class structures that live on clear, yet non-linear manifolds [1].

We also include the methods of perceptual-driven DR with the original DSC measrue (PD) and KNNG measure (PK). Together with our methods, we can categorize the resulting ten methods in two ways:

1) Two unsupervised methods (PCA, t-SNE), and eight supervised methods (LDA, KDA, NCA, RAND, PD, PK, PDD, PDK), and
2) Eight linear methods (PCA, LDA, NCA, RAND, PD, PK, PDD, PDK) and two non-linear methods (KDA, t-SNE).

We use the public C++ libraries [36], [37] to perform LDA and t-SNE, while the remaining methods are taken from the Matlab DR toolbox [16].

Datasets. For a comprehensive evaluation, we collected 93 labeled high-dimensional data of a wide variety of size, dimensionality, and number of classes. Among them, 71 datasets come from the 75 high-dimensional datasets used by Sedlmair et al. [19]. We removed the four synthetic *grid* datasets because of their high degree of artificialness, as well as the lack of separation patterns. We kept all the datasets from the other categories: *real*; *gaussian*, with synthetic gaussian blobs of classes; and *entangled*, datasets containing carefully crafted classes that are non-linearly interleaved in the high-dimensional space. Since DSC and KNNG were both already evaluated with these datasets [11], using them alone might lead to potential over-fitting issues. We thus gathered 22 additional *real* datasets from the UCI repository [38]. We opted for additional *real* datasets to further increase the ecological validity of our evaluation. Table 1 gives an overview over our final collection of 93 datasets and how they break down into the three categories *real*, *gaussian*, and *entangled*. These datasets have substantial variations in terms of data size, ranging from 24 to 43,500, and data dimensionality, from 3 to 295. By applying the ten DR methods to these 93 datasets, we receive 930 projections, in other words, 930 color-coded 2D scatterplots.

5.1 Comparison with Numerical Measures

We first compare the projections resulted from the ten chosen DR methods by using existing quality measures.

Measures. We chose two measures for this numerical comparison: GONG and silhouette coefficient [22]. As discussed in Section 3, GONG is the best class separation measure found by Aupetit and Sedlmair [11]. To apply it to multi-class data, we compute the final GONG value by averaging the separation degree over all data classes. Its value is still in the range [0,1] and the larger values indicate large class separation. In contrast to GONG, SC not only measures the class separation but also takes the cohesion between points into account. Similar to our dDSC, the SC value is in the range [-1,1], where zero means the point is located at the class boundary and a positive value near 1 indicates better cohesion and separability. Note that we do not use the best-known stress measure of MDS [39], because it has been shown unsuitable to assess the class separation [40]. We also do not use dDSC and dKNNG (or DSC and KNNG), as they are used in the design of our methods and hence might bias the study. For the curious reader, however, we provide this additional analysis in the supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2017.2701829.

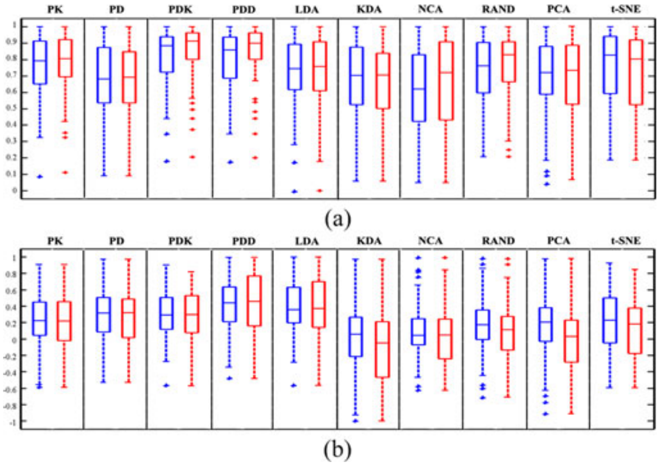


Fig. 8. These boxplots summarize the values of GONG (a) and SC (b) for the ten different DR methods, where the blue boxplots show the score distributions over *all* 93 datasets and red boxplots describe the score of the 53 *real datasets only*.

Results. All individual GONG and SC values generated by the ten different DR methods for each dataset, as well as screenshots of all the projections along with their GONG and SC values can be found in the supplemental material, available online. To facilitate the comparison between different DR methods, we here summarize the GONG and SC scores over *all 93 datasets*. The results are shown in the blue boxplot in Fig. 8. The ten methods are ordered as follows (from left to right): First PD, PK and our two methods (PDK and PDD), then supervised methods (LDA, KDA and NCA), and finally unsupervised methods (RAND, PCA and t-SNE). Since LDA is the closest to PDD, we put it right next to PDD, while t-SNE, as the only non-linear method, is arranged in the last column.

Fig. 8a shows that PDK has slightly better GONG scores than PDD, while PDD performs better than the rest of the methods. In contrast, PDD has larger SC scores than PDK, as shown in Fig. 8b. Furthermore, PD performs similarly to PDK and LDA works better than PDK (but not PDD) in this case. We believe the reason is that dKNNG might not characterize the class density well by modeling only the density of the boundary points. Note that the GONG and SC scores

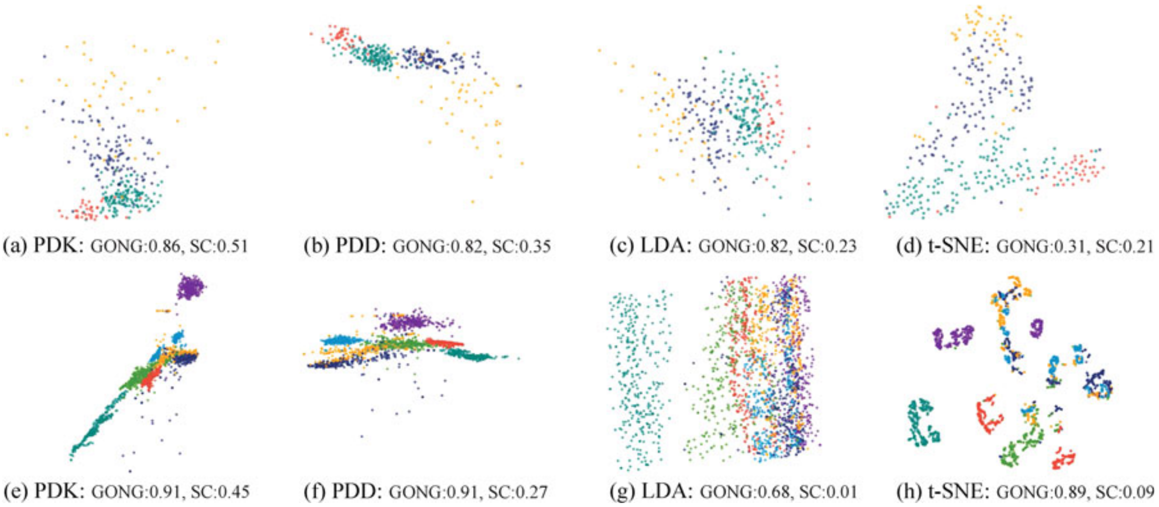

(a) PDK: GONG:0.86, SC:0.51
(b) PDD: GONG:0.82, SC:0.35
(c) LDA: GONG:0.82, SC:0.23
(d) t-SNE: GONG:0.31, SC:0.21
(e) PDK: GONG:0.91, SC:0.45
(f) PDD: GONG:0.91, SC:0.27
(g) LDA: GONG:0.68, SC:0.01
(h) t-SNE: GONG:0.89, SC:0.09

Fig. 9. Comparing the class separation quality of PDK, PDD, LDA, and t-SNE (left to right) on ForestType dataset (top) and Statlog (bottom) dataset, where the resultant GONG and SC scores for each data are shown next to each subfigure.

of NCA and KDA both have large variances. This is expected, because both methods impose class assumptions that might not be valid for some of the datasets.

The red boxplots in Fig. 8 focus on the distributions of GONG and SC scores over the 53 *real datasets*. We deem this differentiation important due to the applicability of our results to real world scenarios (ecological validity), as well as the potential bias introduced through the highly artificial *entangled* datasets. By comparing these (red) boxplots with the blue boxplots in Fig. 8, we can see that six of the supervised DR methods (PK, PD, PDK, PDD, LDA, and NCA) behave better on real data. The remaining four methods perform worse on real data. Meanwhile, PDD and PDK have almost the same GONG scores, while their SC scores differ with a similar amount as in the blue boxplot for all the data. PDD's SC values are higher in the red boxplots than the blue though, indicating that PDD performs better for *real* than *synthetic* data. By ranking the methods according to the GONG and SC scores, we found that the top four methods are: PDD, PDK, LDA, and t-SNE, where PDD does a better job than LDA and t-SNE, while PDK is a little worse than LDA in SC scores.

Qualitative Comparison. Next, we perform a more detailed qualitative comparison on the top four methods we found earlier using two randomly-picked datasets: ForestType [41] and Statlog [38]. Fig. 9 presents the result. We can see from the rightmost column of Fig. 9 that the class structures generated by t-SNE are scattered or split into several sub-classes, so their SC scores are rather small (0.21 and 0.09). PDK receives the highest SC scores with values of 0.51 (Fig. 9a) and 0.45 (Fig. 9e). In terms of GONG, both PDK and PDD score well (0.91) for the Statlog dataset. For the ForestType dataset on top, PDK performs slightly better than PDD by four percent (0.82 versus 0.86). LDA (Figs. 9c and 9g), on the other hand, performs consistently poorly on both datasets; all classes are spread in a major direction. We believe the reason is that LDA cannot identify the second discriminative direction to separate the classes.

5.2 Comparison with Human Judgments

On the other hand, we evaluate PDD and PDK in terms of human perception by conducting an empirical *user study* to collect human judgements. We followed the procedure in Sedlmair et al. [1], and recruited some trained *expert coders* who had experience in rating a large number of scatterplots in terms of how well the classes are visually separated. For details on the method and the justification of the underlying assumptions, we refer the reader to Sedlmair et al. [1]. We do not include the methods of PD and PK, because their results are constantly worse than our density-based PDD and PDK as shown in Fig. 8.

Data Collection. In detail, we recruited three expert coders: two visualization researchers who had more than five years of relevant experience, and a master student who focused on visualization research. Each of them separately rated 744 DR projections, which we got from projecting and presenting the 93 datasets with the eight DR methods (see previous section) as 2D scatterplots. A color scheme designed for categorical data (see http://colorbrewer2.org) was used to show the classes. Since some datasets have a large number of points, some classes might become invisible due to over-plotting, i.e., occlusion. We alleviate this issue by randomly shuffling the drawing order of data points in the scatterplots. Furthermore, we plot a convex hull of a class as a guidance for judging the separability of each class, once the coder clicks on the corresponding class name. For each class, coders rate the separability by using a Likert scale [42], which ranges from 1 ("not separable at all") to 5 ("nicely separable"). Fig. 10 shows the user interface used in the expert coding study. In this example, one expert rated the ten classes (class 0 to 9) as 3, 4, 2, 5, 1, 1, 2, 4, 1, and 1, respectively.

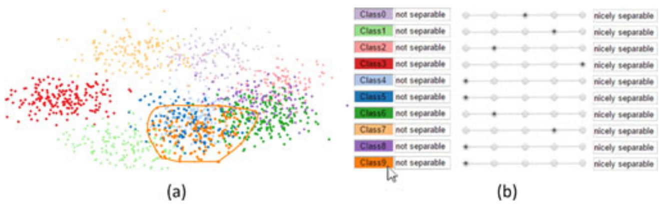

Class0 not separable nicely separable
Class1 not separable nicely separable
Class2 not separable nicely separable
Class3 not separable nicely separable
Class4 not separable nicely separable
Class5 not separable nicely separable
Class6 not separable nicely separable
Class7 not separable nicely separable
Class8 not separable nicely separable
Class9 not separable nicely separable
(a)
(b)

Fig. 10. The user interface in the user study: (a) the visualization results after the DR projection, and (b) the Likert scale for users to choose a rating for each class. By clicking the class button in (b), the corresponding convex hull of the class is revealed in the main visualization (a) for users to check the class separability.

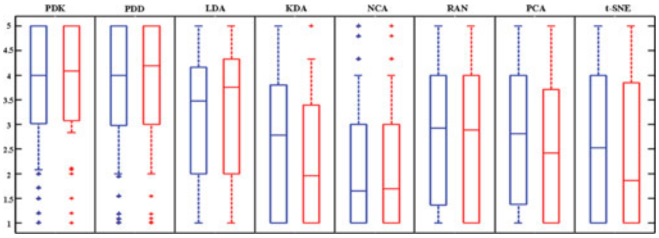


Fig. 11. The above boxplots show the averaged class-wise ratings of each method, where the red and blue boxplots present the ratings of all datasets and only real datasets, respectively.

Overall, each coder needed to judge the class separation of 4,104 color-coded classes across 744 scatterplots. The display order of the scatterplots was randomly chosen to avoid potential ordering bias. Since the judgement process took roughly 8 hours to complete (on average), each coder took a 15-minute break after every hour of work to prevent from fatigue. We used Krippendorff's alpha (for ordinal data) to measure inter-coder reliability [43]. For our collected ratings, Krippendorff's alpha is 0.816, which is larger than 0.8, so this number suggests that the ratings from the expert coders are reliable [43]. Heatmaps of individual class ratings can be found in the supplemental material, available online. *Results.* To investigate how PDD and PDK compare to existing DR methods, we compute the averaged separability rating of each method over all datasets and over only real datasets. The results are shown as blue and red box plots, respectively, in Fig. 11.

Fig. 11 reveals that PDK performs quite similar to PDD, followed by LDA and RAND, while KDA and NCA are the worst although they work quite well for some of the datasets. We think the reason for such idiosyncratic behaviors of KDA and NCA lay in their non-linear kernels, which might degrade the class separability. PCA and t-SNE are quite similar to RAND for all the datasets, but produce smaller separability ratings for the real data. In contrast, LDA generates better class separation for real data, while PDD and PDK perform well regardless of the data category. This suggests that our methods have stronger discrimination ability than LDA.

Comparing the human judgments with the findings from the numerical study shown in Fig. 8, we obtain two interesting observations. First, both PDD and PDK receive larger ratings than LDA in terms of the human judgment results shown in Fig. 11, while PDD and PDK have similar and better performance (respectively) compared to LDA shown in Fig. 8 (numerical). This indicates that PDD and PDK have more advantages in producing results that are more consistent to human perception while being comparable to LDA in terms of GONG and SC scores. Second, t-SNE performs similar or better than RAND in Fig. 8 (numerical), whereas it is worse than RAND in Fig. 11 (human). This reveals that GONG might not be accurate enough to fully characterize human perception, resulting in high t-SNE GONG scores, while the human judgments were much lower. Such a conclusion is further confirmed if we look at the correlation coefficients between the user ratings and the GONG and SC scores, an evaluation approach that was used by Sips et al. [10]. Here, we see that human judgment's correlation with GONG is 0.61, and that with SC is 0.63. This indicates that the match is not bad, but still far from perfect, echoing the findings of previous studies on such measures [9], [10], [11], [19].

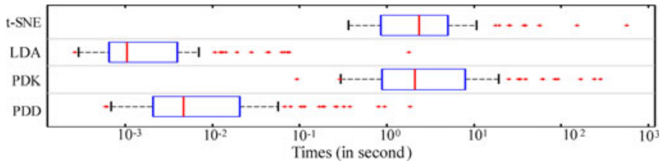


Fig. 12. The boxplot shows the log-scale computational times of the four DR methods with C++ implementations.

5.3 Comparison of Performance

Regarding time performance, all linear methods can quickly finish for most data, whereas non-linear methods are generally much slower. Since some methods are implemented in C++ and some taken from the Matlab DR toolbox, we only compare our method (which is in C++) against the C++ version of LDA and t-SNE.

Results. Fig. 12 presents the results using boxplots. It shows that our PDD is usually faster than t-SNE; our method finishes in less than 0.05 seconds even for some larger testing datasets. PDK, on the other hand, is close to t-SNE, but much slower than PDD. Since the time complexity of LDA is $O(nd^2)(d < n)$ or $O(d^3)(d \geq n)$ [44], LDA is often two or three times faster than PDD if $d < n$, otherwise PDD is faster or similar to LDA. It can be confirmed by the outliers of the boxplots corresponding to LDA and PDD. Since PDD takes less than 0.1 seconds for most data, it suggests that it can enable interactive visualization for most high-dimensional datasets.

6 Extensions

Besides visualizing the general labeled high-dimensional data, our methods can also be extended for the exploration of class-imbalanced data and classification of unlabeled data. Due to the higher computational cost of PDK, we only use PDD to demonstrate the effectiveness of these extensions.

6.1 Exploration of Class-Imbalanced Data

The class-imbalanced data is common in applications like medical disease prediction, fraud detection and risk management, where some classes have many samples, and others a few [45]. Such data is often characterized by the *imbalance ratio* (IR), the ratio of the number of samples in the majority class to the number of samples in the minority class. If we directly apply DR methods to them, the separation will bias towards the classes with more samples, because Eq. (8) assigns the same importance value to each sample. To alleviate this issue, we allow the user to normalize $\mathbf{s}(\mathbf{y}_i)$ by n_c, where the cth class, which includes point $\mathbf{y}_i$, has n_c points. We call this method the *weighted* PDD.

We tested PDD with a six-class *Dermatology* dataset, which consists of 358 records in 34 dim. with an IR value of 5.59. Figs. 13a and 13b show the results generated by PDD and weighted PDD, respectively. Their GONG and SC values are 0.97 versus 0.98 and 0.79 versus 0.83, respectively, showing that the class separations have been slightly enlarged in the weighted PDD. Inspecting Fig. 13b reveals that the minority class in red has been more clearly separated from the others. In contrast, the separations among

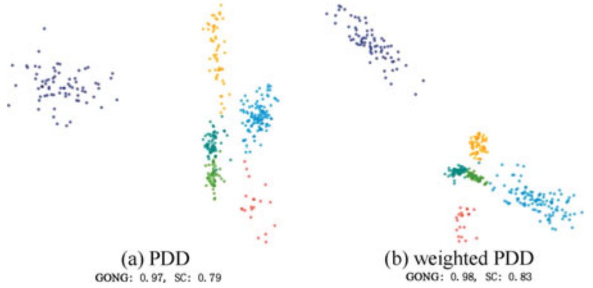


Fig. 13. The visualizations of the *Dermatology* dataset generated by unweighed PDD (a) and weighted PDD (b), respectively.

the green, cyan, yellow and blue classes are slightly weakened and these classes become more compact (except the blue one). The dark blue class remains clearly separated from the others, but its distribution becomes skewed. Investigating which data attributes result in such changes is part of our future work.

6.2 Classification of Unlabeled Data

Not all high-dimensional data necessarily comes with class labels. A user, however, might still be interested in learning how the data breaks down into groups of similar items. To do so, a user might first apply a set of clustering algorithms and evaluate which one separates the classes using DR to project and visualize the data [3]. This approach is inherently supported by our methods.

In other cases, however, automatic clustering might not lead to meaningful results, and a user would like to incorporate their prior knowledge by different means. To support this process, we allow the user to manually label certain data items based on their previous knowledge. The DR embedding is then iteratively updated whenever new labels come in. This scenario has been characterized by Sacha et al. [46], and resembles a typical approach in active learning of classification models.

Our methods can then take such partially labeled data as its input and create a lower-dimensional embedding of all the data points. This approach is similar to the idea of semantic interaction proposed by Endert et al. [47], in which users can move points in the lower-dimensional embedding and the changed distances are used to update the DR process.

More specifically, using PDD as an example, the following steps are iteratively pursued until the user reaches a desired result:

(1) randomly sample a small subset of items and present it to the user for labeling;
(2) apply PDD to the labeled samples;
(3) classify the remaining unlabeled data using the projection matrix obtained from PDD; and
(4) project all data points to the low-dimensional embedding and show the classification result to the user.

The classification is achieved with a nearest neighbour classifier [48], which seeks the nearest class center μ'_j for each point and takes j as the label. This is similar to the classification performed by LDA, which takes the same procedure in the low-dimensional embedding for the data points [7].

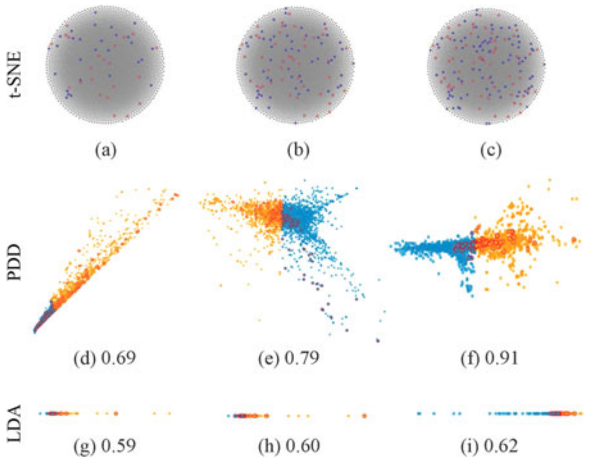


Fig. 14. The classification and visualization of the *Spambase* dataset. (a,b,c) The results generated by t-SNE with randomly selected 50, 100, and 200 labeled points; (d,e,f) the results of PDD classification with accuracies of 0.69, 0.79, and 0.91 for (a,b,c), respectively; (g,h,i) the results of LDA classification with accuracies of 0.59, 0.60 and 0.62 for (a,b,c), respectively. The mis-classified points in (d,e,f,g,h,i) are highlighted with red borders.

To see the effectiveness of this pipeline, we apply it to the 58-dimensional SPAMBASE dataset, which consists of 4,601 emails with known labels. Having this ground truth data on the labels allows us to assess the PDD classification accuracy (step 3) by simply computing the percentage of correctly classified points. Fig. 14 presents the progressive procedure, where 50, 100 and 200 labeled samples are gradually added for training within three iterations. To clearly see which samples are selected, we overlay them on their t-SNE embedding as colored points (see Figs. 14a, 14b, and 14c). Figs. 14d, 14e, and 14f presents the classification results by applying PDD to the selected samples shown in Figs. 14a, 14b, and 14c, where the mis-classified points are highlighted with red borders and the classification accuracies are 0.69, 0.79 and 0.91, respectively. We can see that the accuracy gradually improves as the number of labeled samples increases. We believe that the accuracy can still be improved if we use advanced active learning algorithms [49] to select samples instead of random selection.

For comparison, we also apply LDA to classify the whole data based on the selected labeled samples and obtain the results as shown in Figs. 14g, 14h, and 14i, where the accuracy is 0.59, 0.60 and 0.62, respectively. To avoid randomness, we run the PDD- and LDA-based progressive labeling multiple times. As the results are always similar, we can see that our PDD is also superior to LDA for interactive visual classification.

7 DISCUSSION AND FUTURE WORK

This paper presents a *perception-driven* linear DR approach, a novel dimensionality reduction technique for visualizing labeled high-dimensional data. The goal of our approach is to provide user with projections that optimize the visual separability of classes. We quantitatively and qualitatively compare our approach with different state-of-art DR methods, showing that our methods PDD and PDK outperform them with similar or better results in most cases. Lastly, we demonstrate the usefulness of our method PDD in exploring imbalanced data and unlabeled data.

Our approach still has certain limitations, which we would like to address in the future. Although PDD outperforms the other methods for most data, for some datasets it results in inter-class distances that are smaller than those of PDK. To alleviate this issue, one possible direction would be to reformulate the dDSC by putting more emphasis on inter-class distance than intra-class distance. Second, PDK produces similar or slightly better results than PDD but its computational cost is more expensive due to the construction of nearest neighborhood graph. We have tried to bypass this burden by assuming that each point is connected to all remaining data points. Yet, directly applying dKNNG to such a fully-connected graph between all points does not well characterize the separation degree. Although both methods are derived from perceptual separation measures, they can be easily extended for classification tasks. We would like to investigate their performances in high-dimensional data classification. Last, we plan to investigate the possibility of semi-supervised methods for interactive classification of unlabeled data, where the class structures revealed by labeled data and inherent clusters within the data are consistently integrated together.

ACKNOWLEDGMENTS

This work is supported by the grants of NSFC-Guangdong Joint Fund (U1501255), NSFC (61379091, 91630204), the National Key Research & Development Plan of China (2016YFB1001404), Shandong Provincial Natural Science Foundation (2016ZRE27617) and the Fundamental Research Funds of Shandong University. Yunhai Wang and Kang Feng are joint first authors.

REFERENCES

[1] M. Sedlmair, T. Munzner, and M. Tory, "Empirical guidance on scatterplot and dimension reduction technique choices," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2634–2643, Dec. 2013.

[2] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *J. Mach. Learn. Res.*, vol. 16, pp. 2859–2900, 2015.

[3] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner, "Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences," in *Proc. 5th Workshop Beyond Time Errors: Novel Eval. Methods Vis.*, 2014, pp. 1–8.

[4] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[5] I. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2005.

[6] J. B. Kruskal and M. Wish, *Multidimensional Scaling*, vol. 11. Newbury Park, CA, USA: Sage, 1978.

[7] T. Cacoullos, *Discriminant Analysis and Applications*. New York, NY, USA: Academic, 2014.

[8] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput.*, vol. 12, no. 10, pp. 2385–2404, 2000.

[9] M. Sedlmair and M. Aupetit, "Data-driven evaluation of visual quality measures," *Comput. Graph. Forum*, vol. 34, no. 3, pp. 201–210, 2015.

[10] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan, "Selecting good views of high-dimensional data using class consistency," *Comput. Graph. Forum*, vol. 28, no. 3, pp. 831–838, 2009.

[11] M. Aupetit and M. Sedlmair, "SepMe: 2002 new visual separation measures," in *Proc. IEEE Pacific Vis. Symp.*, 2016, pp. 1–8.

[12] A. Anand, L. Wilkinson, and T. N. Dang, "Visual pattern discovery using random projections," in *Proc. IEEE Symp. Visual Anal. Sci. Technol.*, 2012, pp. 43–52.

[13] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. New York, NY, USA: Wiley, 1988.

[14] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. PéRez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognit.*, vol. 46, no. 1, pp. 243–256, 2013.

[15] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2004, pp. 513–520.

[16] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[17] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Comput.*, vol. C-100, no. 9, pp. 881–890, Sep. 1974.

[18] E. Bertini, A. Tatu, and D. Keim, "Quality metrics in high-dimensional data visualization: An overview and systematization," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2203–2212, Dec. 2011.

[19] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory, "A taxonomy of visual cluster separation factors," *Comput. Graph. Forum*, vol. 31, no. 3pt4, pp. 1335–1344, 2012.

[20] A. Tatu, et al., "Combining automated analysis and visualization techniques for effective exploration of high-dimensional data," in *Proc. IEEE Symp. Visual Anal. Sci. Technol.*, 2009, pp. 59–66.

[21] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *J. Cybern.*, vol. 4, no. 1, pp. 95–104, 1974.

[22] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.

[23] J. Choo, S. Bohn, and H. Park, "Two-stage framework for visualization of clustered high dimensional data," in *Proc. IEEE Symp. Visual Anal. Sci. Technol.*, 2009, pp. 67–74.

[24] J. Choo, H. Lee, J. Kihm, and H. Park, "iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction," in *Proc. IEEE Symp. Visual Anal. Sci. Technol.*, 2010, pp. 27–34.

[25] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Neural Inf. Process. Syst.*, 2004, vol. 16, Art. no. 153.

[26] S. Ingram, T. Munzner, and M. Olano, "Glimmer: Multilevel MDS on the GPU," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 2, pp. 249–261, Mar./Apr. 2009.

[27] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[28] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[29] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2001, vol. 14, pp. 585–591.

[30] J. M. Lewis, L. Van Der Maaten, and V. de Sa, "A behavioral investigation of dimensionality reduction," in *Proc. 34th Conf. Cognitive Sci. Soc.*, 2012, pp. 671–676.

[31] R. Etemadpour, R. Motta, J. G. de Souza Paiva, R. Minghim, M. C. F. de Oliveira, and L. Linsen, "Perception-based evaluation of projection methods for multidimensional data visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 1, pp. 81–94, Jan. 2015.

[32] C. M. Bender and S. A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers I*. Berlin, Germany: Springer, 1999.

[33] D. Pham and D. Karaboga, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. Berlin, Germany: Springer, 2012.

[34] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chemical Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.

[35] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[36] S. Bochkanov and V. Bystritsky, "ALGLIB-a cross-platform numerical analysis and data processing library," ALGLIB Project. Novgorod, Russia, (2011). [Online]. Available: http://www.alglib.net/

[37] L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, 2014.

[38] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[39] I. Borg and P. J. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Berlin, Germany: Springer, 2005.

[40] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz, "Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 3, pp. 564–575, May/Jun. 2008.

[41] B. Johnson, R. Tateishi, and Z. Xie, "Using geographically weighted variables for image classification," *Remote Sens. Lett.*, vol. 3, no. 6, pp. 491–499, 2012.

[42] R. Garland, "The mid-point on a rating scale: Is it desirable," *Marketing Bulletin*, vol. 2, no. 1, pp. 66–70, 1991.

[43] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*. Newbury Park, CA. USA: Sage, 2004.

[44] B. Scholkopft and K.-R. Mullert, "Fisher discriminant analysis with kernels," in *Proc. IEEE Workshop Neural Netw. Signal Process.*, 1999, vol. 1, no. 1, Art. no. 1.

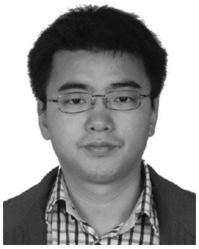
[45] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 1–6, 2004.

[46] D. Sacha, et al., "Visual interaction with dimensionality reduction: A structured literature analysis," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 241–250, Jan. 2017.

[47] A. Endert, P. Fiaux, and C. North, "Semantic interaction for sensemaking: Inferring analytical reasoning for model steering," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2879–2888, Dec. 2012.

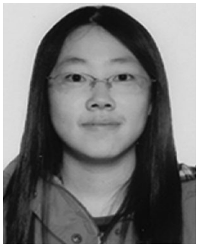
[48] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York, NY, USA: Academic, 1990.

[49] D. Sacha, et al., "Human-centered machine learning through interactive visualization: Review and open challenges," in *Proc. Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn.*, 2016, pp. 641–646.



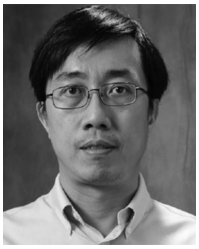
Yunhai Wang received the doctoral degree in computer science from Supercomputer Center, Chinese Academy of Sciences, China, in 2011. He is an associate professor in the School of Computer Science and Technology, Shandong University. His interests include scientific visualization, information visualization, and computer graphics.

Kang Feng received the BEng degree from the Department of Computer Science and Technology, Shandong University, in 2015. He is now working toward the MS degree in the Interdisciplinary Research Center (IRC), Shandong University, supervised by Prof. Baoquan Chen and Prof. Yunhai Wang. His research interests include visual analytics and machine learning.



Xiaowei Chu is a senior student in the School of Computer Science and Technology, Shandong University. Her research interests include information visualization and visual analytics.

Jian Zhang received the PhD degree in applied mathematics from the University of Minnesota, in 2005. After a postdoc with Pennsylvania State University, he is now a research scientist in the Supercomputing Center of Computer Network Information Center, Chinese Academy of Sciences (CAS). His current research interests include scientific computing, high performance computing, and scientific visualization.



Chi-Wing Fu received the PhD degree in computer science from Indiana University Bloomington. He joined the Chinese University of Hong Kong as an associate professor from 2016. He served as the program co-chair of SIGGRAPH ASIA 2016 technical brief and poster, associate editor of the *Computer Graphics Forum*, and program committee member in IEEE Visualization. His research interests include computer graphics, visualization, and user interaction.

Michael Sedlmair received the PhD degree in computer science from Ludwig Maximilians University of Munich. After a postdoc with the University of British Columbia, he is now an assistant professor with the University of Vienna. His research interests include visualization design and evaluation, human factors in visualization and data analysis, and human-computer interaction.

Xiaohui Yu received the BSc degree from Nanjing University, China, the MPhil degree from the Chinese University of Hong Kong, and the PhD degree from the University of Toronto, Canada. His research interests include the areas of database systems and data mining. He is a professor in the School of Computer Science and Technology, Shandong University, China, and an associate professor in the School of Information Technology, York University, Canada (on leave).

Baoquan Chen received the MS degree from Tsinghua University, Beijing and the PhD degree from the State University of New York at Stony Brook. He is now a professor of School of Computer Science and Technology, Shandong University. He received the NSF CAREER award 2003, IEEE Visualization Best Paper Award 2005, and NSFC Outstanding Young Researcher program in 2010. His research interests generally lie in computer graphics, visualization, and human-computer interaction.