Info Vis

Article

Information Visualization
2018, Vol. 17(4) 282–305

DOI: 10.1177/1473871617713337
journals.sagepub.com/home/ivi


Projections as visual aids for classification system design

Paulo E Rauber[1,2], Alexandre X Falcão[2] and Alexandru C Telea[1]

Abstract

Dimensionality reduction is a compelling alternative for high-dimensional data visualization. This method provides insight into high-dimensional feature spaces by mapping relationships between observations (high-dimensional vectors) to low (two or three) dimensional spaces. These low-dimensional representations support tasks such as outlier and group detection based on direct visualization. Supervised learning, a subfield of machine learning, is also concerned with observations. A key task in supervised learning consists of assigning class labels to observations based on generalization from previous experience. Effective development of such classification systems depends on many choices, including features descriptors, learning algorithms, and hyperparameters. These choices are not trivial, and there is no simple recipe to improve classification systems that perform poorly. In this context, we first propose the use of visual representations based on dimensionality reduction (projections) for predictive feedback on classification efficacy. Second, we propose a projection-based visual analytics methodology, and supportive tooling, that can be used to improve classification systems through feature selection. We evaluate our proposal through experiments involving four datasets and three representative learning algorithms.

Keywords

High-dimensional data visualization, dimensionality reduction, pattern classification, visual analytics, graphical user interfaces

Introduction

In *supervised learning*, a subfield of machine learning, the important task of pattern classification consists of assigning a class label to a high-dimensional vector based on generalization from previous examples.[1] In broad terms, this task is typically solved by finding parameters for a classification model that maximizes a measure of efficacy. In this context, *efficacy* refers to desirable characteristics that a classification system should possess to be efficient and effective. These characteristics include quantitative metrics capturing the classification accuracy (which captures the effectiveness aspect), and also the use of a limited set of so-called features to describe the input space (which captures the efficiency aspect).

Pattern classification is a challenging task, partly due to its extremely large design space. For our purposes, this task can be divided into representation and learning, as follows.

Representation is concerned with how objects of interest are modeled as high-dimensional vectors. Elements of these vectors usually correspond to measurable characteristics (features) of the objects. Many different features can be considered, and it is generally unclear which of them are valuable for generalization. For example, in image classification, a wide variety of

[1]Department of Mathematics and Computing Science, University of Groningen, Groningen, The Netherlands
[2]University of Campinas, Campinas, Brazil

Corresponding author:
Paulo E Rauber, Department of Mathematics and Computing Science, University of Groningen, Nijenborgh 9, Groningen 9747 AG, The Netherlands.
Email: p.e.rauber@rug.nl

color, texture, shape, and local features can be extracted from images.[2] Using too few features can lead to poor generalization, thereby reducing classification effectiveness, and using too many features can be prohibitively expensive to obtain or compute, thereby reducing efficiency, or even introduce confounding information into the training data.[3,4] Deep neural networks recently became able to bypass feature design by dealing directly with raw images.[5,6] Yet, such networks require very large amounts of labeled (training) data, which are not always available, and pose additional design challenges of their own.[7] Hence, feature selection for classification system design still is a very important open problem.

Learning algorithms have to be selected, fine-tuned, and tested once a representation is available. A huge number of such algorithms exist, based on a wide variety of principles, and no single algorithm is the best for every situation.[8] Practitioners usually compare algorithms and hyperparameter choices using cross-validation.[1] However, this approach is bounded by the limited feedback that numerical (classification) accuracy measures can provide. As a consequence, when suboptimal results are obtained, designers are often left unaware of which aspects limit classification system accuracy, and what can be done to improve such systems. This and other issues have been referred to as the "black art" of machine learning[9] and motivate our interest in using interactive techniques to assist the design of classification systems.

Dimensionality reduction (*DR*) techniques are a highly scalable alternative for high-dimensional data visualization and exploration.[10] Given a dataset composed of high-dimensional vectors (also called observations or data points), DR techniques find corresponding low-dimensional vectors that attempt to preserve the so-called data structure. This structure is characterized by distances between observations, presence of clusters, and overall spatial data distribution.[11,12] In this text, we refer to the representation obtained by DR by the term *projection*. For visualization purposes, DR techniques typically reduce the number of dimensions to two or three. The resulting projections are typically depicted by scatterplots and enable insight into the structure of the original data.[13]

Visual exploration of high-dimensional datasets via projections has been widely applied to many data types, such as text documents,[14] multimedia collections,[15] gene expressions,[16] and networks.[17] However, projections are rarely used for the task of classification system design. Considering the aforementioned difficulties in designing such systems, we propose a visual analytic approach based on DR that supports two (highly interrelated) tasks:

T1: predicting classification system efficacy
T2: improving classification systems

With respect to task *T1*, we show how the presence of visual outliers, overall visual separation between observations in distinct classes, and visual distribution of observations of a given class are reflected in classification results. More specifically, we show that the structure of a projection is often a good predictor of the accuracy that a classifier can deliver on the original data, both in the case of using a predefined feature set, and in the case of performing feature selection; that confusion zones, containing misclassification results, can be often spotted using projections; and that projections can help the guided pruning of a complex dataset to increase classification accuracy.

Concerning task *T2*, we propose a combination between the aforementioned projections and visualizations called feature projections, which present correlations between features and information derived from traditional feature-scoring techniques to help designers select important features for classification systems. Overall, our contributions show that projections are valuable tools for various aspects of classification system design, especially in cases where traditional aggregate accuracy metrics do not provide sufficient insights.

We illustrate our approach through use cases involving both real and synthetic challenging datasets and representative learning algorithms.

This article is organized as follows. Section "Preliminaries" presents our notation and definitions. Section "Related work" places our effort in the contexts of information visualization and machine learning. Section "Proposed approach" summarizes our approach and compares it to related work. Section "T1: predicting system efficacy" details our first contribution—showing how projections can be used as insightful predictors of classification system efficacy. Section "T2: improving system efficacy" details our second contribution—showing how the visual feedback given by projections can be integrated into an interactive and iterative workflow for improving system efficacy through qualitative and quantitative data exploration. This workflow is summarized in section "Proposed workflow." Section "Discussion" provides a critical analysis of the experiments, limitations, and weaknesses of our proposals. Importantly, it outlines cases where projections are known to fail as predictors of classification system efficacy, and why such cases do not contradict our proposal. Finally, section "Conclusion" summarizes the article and presents directions for future work.

Preliminaries

The following is a summary of the definitions and notation employed in this text.

A (supervised) dataset $\mathcal{D}$ is a sequence $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$. Every pair $(\mathbf{x}_i, y_i) \in \mathcal{D}$ is composed of an *observation* $\mathbf{x}_i \in \mathbb{R}^D$, and a *class label* $y_i \in \{1, \ldots, C\}$, where C is the number of *classes*. As an example, observations may correspond to images of animals and the classes to the C distinct species present in the images. The jth element of $\mathbf{x}_i$ corresponds to *feature j* and is typically measured from an object of interest. Considering the previous example, a feature may represent the *redness* of an image.

We denote the set of all features under consideration by $\mathcal{F} = \{1, \ldots, D\}$. For any $\mathcal{F}' \subseteq \mathcal{F}$, having $D' \leqslant D$ features, we denote by $\mathcal{D}_{\mathcal{F}'}$ the dataset corresponding to $\mathcal{D}$ with features restricted to $\mathcal{F}'$.

A *learning algorithm* finds a function, called *classifier*, that maps observations to classes based on generalization from a training (data)set $\mathcal{D}$. *Generalization* is usually evaluated by *cross-validation*, which consists of partitioning the available data into a set for model learning and a set for model evaluation. *Feature selection* aims at finding a small feature subset $\mathcal{F}' \subseteq \mathcal{F}$ such that the restricted training set $\mathcal{D}_{\mathcal{F}'}$ is sufficient for generalization.

DR finds a *projection* $\mathcal{P} = \mathbf{p}_1, \ldots, \mathbf{p}_N$, where $\mathbf{p}_i \in \mathbb{R}^d$, that attempts to preserve the *structure* of an original (unsupervised) dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, considering that each observation $\mathbf{x}_i$ corresponds to point $\mathbf{p}_i$. For the purposes of visualization, d is usually 2 or 3. DR is related to the feature selection task, discussed in the next section. However, there are important differences, especially in our context: first, feature selection can be seen as a specific type of DR, where the d dimensions of the resulting projection are chosen from the D dimensions (features) of the input data; in contrast, DR methods used in data visualization typically *synthesize* d new dimensions from the original D, as to better preserve the data structure. All state-of-the-art DR methods, such as the ones used in our work, are of this type. Second, DR (used for visualization) has a two-dimensional (2D) or three-dimensional (3D) target space, whereas feature selection typically yields higher dimensional spaces ($d > 3$). Third, and most importantly, feature selection, as used in our context, aims to reduce the dimensionality of an input space for increasing the efficacy of a classification system; in contrast, DR (again, as used in our context) aims to create visualizations that help designers understand this input space.

Related work

High-dimensional data visualization is a challenging and important task in many scientific and business applications. For an extensive overview of the field, we refer to the recent survey by Liu et al.[18] There are many alternatives for visual exploration of high-dimensional data, such as parallel coordinate plots,[19] radial plots,[20] star plots,[21] star coordinates,[22] table lenses,[23] and scatterplot matrices.[24] A common challenge for these methods is scalability to datasets with relatively modest numbers of observations *and* dimensions. DR techniques effectively address these scalability issues by finding a low-dimensional representation of the data that retains *structure*, which is defined by relationships between points, presence of clusters, or overall spatial data distribution.[11–13,18] The resulting projections can be represented as scatterplots, which allow reasoning about clusters, outliers, and trends by direct visual exploration. These and other tasks addressed by DR-based visualizations are detailed by Brehmer et al.[10]

DR techniques are typically divided into linear (e.g. principal component analysis (PCA), linear discriminant analysis (LDA), and multidimensional scaling (MDS)) and non-linear (e.g. Isomap, locally linear embedding (LLE), and t-distributed stochastic neighbor embedding (t-SNE)).[12,13] Although many traditional DR techniques are computationally expensive, highly scalable techniques have also been proposed (e.g. least square projection (LSP),[14] local affine multidimensional projection (LAMP)[15] and local convex hull (LoCH)[25]). These techniques are currently capable of dealing with hundreds of thousands of observations (or more)—although visual clutter eventually becomes a problem. Guidelines for choosing suitable DR methods for a particular task are outlined by Sedlmair et al.[26]

More related to our work, several visualization techniques have been proposed to help the interactive exploration of projections. Most notably, Tatu et al.[27] propose a process for finding *interesting* subsets of features, and displaying the results of DR restricted to these features, with the goal of aiding qualitative exploration. Yuan et al.[28] present an interactive tool to visualize projections of observations restricted to selected subsets of features. Additionally, in their tool, features are placed in a scatterplot based on pairwise similarities. This is analogous to the representation we propose in section "T2: improving system efficacy." However, differences exist—Yuan et al.[28] aim at subspace cluster exploration, while our goal is to provide support for classification system design. This difference is manifested by our additional mechanisms, which include feedback from automatic feature-scoring techniques and classification results. The work of Turkay et al.[29] also combines scatterplots of observations and features for high-dimensional data

exploration and is also concerned with tasks that are unrelated to classification system design.

Pattern classification is one of the most widely studied problems in machine learning. Learning algorithms, such as *k*-nearest neighbors (KNN), naive Bayes, support vector machines (SVMs), decision trees, artificial neural networks, and their ensembles, have been applied in a wide variety of practical problems.[1] Since the objective of pattern classifiers is to generalize from previous experience, hyperparameter search and efficacy estimation are usually performed using cross-validation.[30] Diagnosing the cause of poor generalization in classification systems is a hard problem. Options include using cross-validation to compute efficacy indicators (e.g. accuracy, precision and recall, and area under the receiver operating characteristic (ROC) curve) and learning curves, which show generalization performance for an increasing training set. In multi-class problems, confusion matrices can also be used to diagnose mistakes between classes.[31]

In the context of visualization, Talbot et al.[32] propose the visual comparison of confusion matrices to help users understand the relative merits of various classifiers, with the goal of combining them into better ensemble classifiers. In contrast to their work, we offer fine-grained insight into a single classification system using projections as a visualization technique. Other visualization systems also aim at integrating human knowledge into the classification system design process. *Decision trees* are particularly suitable for this goal, as they are one of the few easily interpretable classification models.[33] Schulz et al.[34] propose a framework that can be used to visualize (in a projection) the decision boundary of a SVM, a model which is usually hard to interpret. Projections have also been used specifically for visualizing internal activations of artificial neural networks.[35] More related to our work, other works also propose visualizations that consider classification systems as *black-boxes*. They usually study the behavior of such systems under different combinations of data and parameterizations. In this context, Paiva et al.[36] present a visualization methodology that supports tasks related to classification based on similarity trees. Similar to projections, similarity trees are a high-dimensional data visualization technique that maps observations to points in a 2D space, and connects them by edges to represent similarity relationships. In contrast to our methodology for system improvement, their methodology focuses on visualization of classification results and observation labeling. At a higher-level of abstraction, the use of visualization techniques to "open the black box" of general algorithm design, including (but not limited to) classification systems, is also advocated by Mühlbacher et al.[37]

Active learning refers to a process where the learning algorithm iteratively suggests informative observations for labeling. The objective of this process is to minimize the effort in labeling a dataset. Because this is an iterative and interactive process, visualization systems have been proposed to aid in the task, and sometimes include a representation of the data based on projections.[38,39] However, in these examples, projections do not have a role in improving classification system efficacy.

Feature selection is another widely researched problem in machine learning, because the success of supervised learning is highly dependent on the predictive power of features.[3,4] Feature selection techniques are usually divided into *wrappers*, which base their selection on learning algorithms, and *filters*, which rely on simpler metrics derived from the relationships between features and class labels.[4] The work of Krause et al.[40] is an example of visualization system that aids feature selection tasks by displaying aggregated feature relevance information, which is computed based on feature selection algorithms and classifiers. Their glyph-based visualizations are completely different from the projection-based integrated visualizations that implement our methodology, which are outlined in the next section.

Proposed approach

Our visualization approach aims to support two tasks (*T1* and *T2*), which we introduce in the following sections.

Predicting system efficacy (T1)

Consider the works presented in section "Related work" that use projections to represent observations in classification tasks,[38,39] or the projections of traditional pattern classification datasets.[13] If a projection shows good visual separation between the classes in the training data, and if this is expected to generalize to test data, it is natural to suppose that building a good classifier will be easier than when such separation is absent.

However, there is little evidence in the literature to defend the use of projections as predictors of classification system efficacy. As a consequence, it is unclear *whether* and, even more importantly, *how* insights given by projections complement existing methods of prognosticating and diagnosing issues in the classification pipeline. In section "T1: predicting system efficacy," we present a study that focuses precisely on these questions. It is important to emphasize the term *predictor*: we aim at obtaining insights on the ease of building a good classification system using projections *before* actually building the entire system.

In summary, the study presented in section "T1: predicting system efficacy" consists of the following. Considering a particular classification dataset split into training and test data, a projection of each of these sets is computed. Some claims are made about the classification problem based on the visual feedback provided by the training set projection and are followed by evidence that supports its predictive feedback. In many cases, some aspect of the problem is altered (e.g. features or observations under consideration), and the visual feedback is again evaluated.

We are aware of a single previous work that studies how projections relate to classifier efficacy,[41] which provides evidence that projections showing *well-separated* classes (as measured by the so-called silhouette coefficient) correlate with higher classification accuracies. However, that study has significant limitations. First, characterizing a projection by a single numerical value (the silhouette coefficient) is coarse and uninsightful. To support understanding *how* a classification system relates to what a projection shows on a finer scale, we perform and present our analyses at the *observation* level. Second, the silhouette coefficient used by Brandoli et al.[41] can be severely misleading, since it may be poor (low) even when good visual separation between classes exists. This happens, for instance, when the same class is spread over several compact groups in a projection. Third, we present a concrete projection-based methodology to improve classification system (*T2*), whereas Brandoli et al.[41] only conjecture this possibility.

Consider simple alternatives to visualize classification system issues, such as confusion matrices,[31] or listing misclassified observations together with their k-nearest neighbors. While simple to use, these mechanisms have significant limitations: confusion matrices become hard to inspect for a moderate number of classes, while listing does not scale well to hundreds (or even tens) of observations. Most importantly, these alternatives do not encode spatial information about observations in *confusion zones*, which we define in section "T1: predicting system efficacy."

Improving system efficacy (T2)

In section "T2: improving system efficacy," we propose a projection-based methodology for interactive feature space exploration that allows selecting features to improve the efficacy of a classification system (*T2*). This methodology is highly dependent on the use of projections as predictors of classification system efficacy (*T1*). As such, we describe next our methodology that jointly addresses the two tasks.

We implement this methodology in a visual analytics tool that links views of projections, representations of feature relationships, feature scoring, and classifier evaluation, in an attempt to provide a cost-effective and easy-to-use way to select features for arbitrary ("black-box") learning algorithms.

The visual analytics workflow supported by our system, detailed in section "T2: improving system efficacy," is illustrated by Figure 1. This figure shows how our visual tools interact to support *T1* and *T2* for the overall goal of building better classification systems. The process can be summarized by a simplified 10-step flowchart. We start by partitioning a collection of objects of interest (images, in this example) into training and validation sets. Next, we extract a number of features from the training images, transforming them into observations (1). These observations are mapped into a projection (2). Optionally, to assure that the projection has a high quality, we may evaluate the various projection error metrics proposed in Martins et al.,[42,43] and fine-tune the DR algorithm parameters accordingly. Assuming the projection has sufficient quality, we study the visual separation between the classes using our proposed visual tools. If the separation is poor (4), we use our iterative feature exploration/selection tools (*T2*) to prune the feature set under consideration (5), and repeat the DR step until we obtain a good separation or decide that such separation is too difficult. If good separation is obtained (3), we proceed in building, training, and evaluating a classifier in the validation set, using the traditional machine learning protocol (6). If the evaluation shows good performance (7), the workflow ends with a good classification system that may be used in production. If the evaluation reveals poor performance (8), we use again our visual exploration tools to study what has gone wrong in the validation set. For instance, we may find that some types (*i.e.*, subsets of classes) of observations are consistently misclassified. In this case, and depending on the importance of these observations, we can choose to filter them out, simplifying the classification problem for the purposes of designing the system (9). Alternatively, we may find that such filtering is not possible, due to the relevance of the misclassified observations. In that case, we decide that we need to design new features, possibly using insights obtained through visual feedback (10).

The added value of our visual tools, which are represented in Figure 1, is twofold.

First, the tools provide evidence about potential flaws in a classifier before it is built (*T1*). This is supported by section "T1: predicting system efficacy," which shows how qualitative feedback obtained from projections relates to classification system efficacy in (unseen) test data.

Second, our tools provide a (partially guided) way to iteratively improve the overall classification system. This is supported by section "T2: improving system

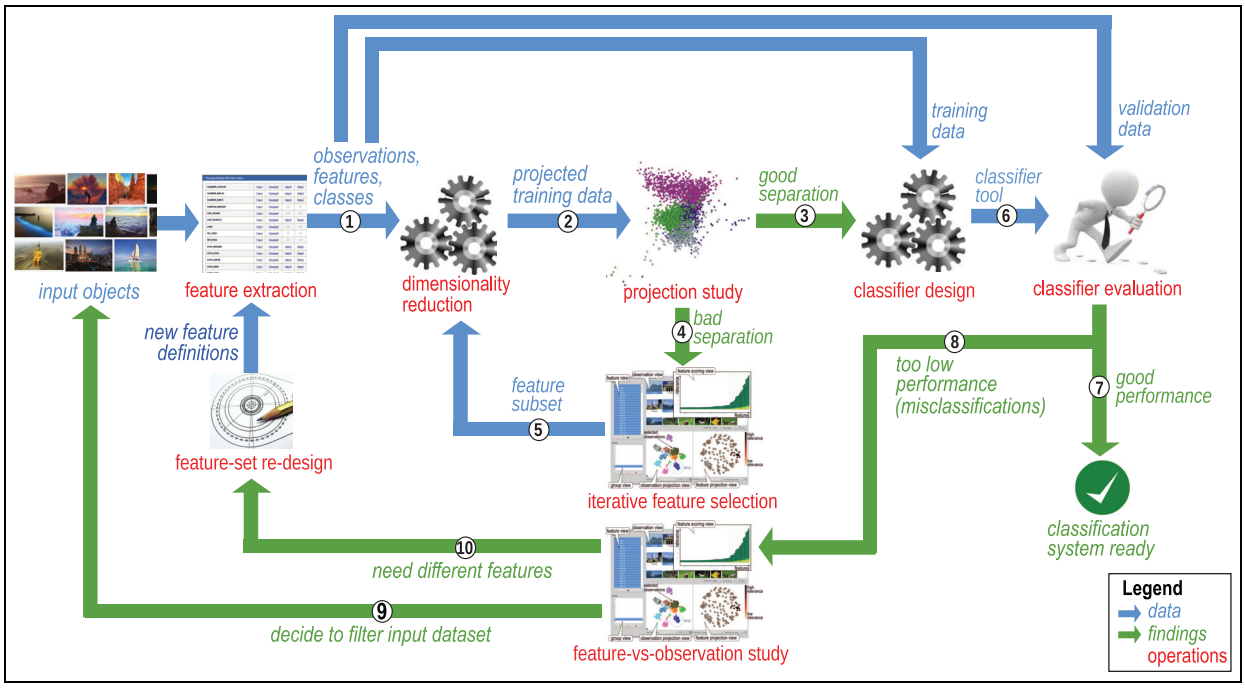

Figure 1. Visual analytic workflow for classification system design proposed in this article (see section "Proposed approach").

efficacy," which shows how their visual feedback can be used to improve classification system efficacy in (unseen) test data through feature selection.

T1: predicting system efficacy

As outlined in section "Proposed approach," this section is concerned with how projections can be used to predict classification system efficacy (*T1*). The main role of this section is to support the actual interactive projection-based system for classification system improvement presented in section "T2: improving system efficacy."

For this purpose, we conducted experiments on several datasets, which are presented in sections "Madelon dataset," "Melanoma dataset," "Corel dataset," and "Parasites dataset." Section "Experimental protocol" details the aspects of the experimental protocol that hold for every dataset under consideration.

Experimental protocol

The first step in our protocol is to randomly partition a dataset into training and test sets (one-third of the observations). Following good practice in machine learning, the partitioning is stratified,[44] that is, the ratio of observations belonging to each class is preserved in the test set.

Projections can be created independently for the training and for the test data. These projections can be represented by scatterplots, where each point is colored according to its class label. When displaying classification results for a test set in a scatterplot, we will use triangular glyphs to represent misclassified observations, colored based on their (incorrect) classifications, and rendered slightly darker (for emphasis).

In addition to showing these scatterplots, we also display a metric called *neighborhood hit* (NH).[14] For a given number of neighbors k (in our experiments, $k = 6$), the NH for a point $\mathbf{p}_i \in \mathcal{P}$ is defined as the ratio of its k-nearest neighbors (except $\mathbf{p}_i$ itself) that belong to the same class as the corresponding observation $\mathbf{x}_i$. The NH for a projection is defined as the average NH over all its points. Intuitively, a high NH corresponds to a projection where the real classes (ground truth) are visually well separated. Therefore, the NH metric is a good quantitative characterization of a projection for our purposes.

The DR technique that we use in this work is a fast implementation of t-SNE,[45] using default parameters and Euclidean distance. We chose t-SNE due to its widespread popularity and demonstrated capacity to preserve neighborhoods in projections.[13] However, our proposal does not depend on this particular technique, and other DR techniques can be used with no additional burden. For instance, we employed LSP[14] in our early work, but decided in favor of t-SNE due to its ability to preserve clusters in projections.

Our workflow requires a projection that preserves well *neighborhoods* from $\mathbb{R}^D$ in $\mathbb{R}^2$. This may be assessed

through the projection quality metrics described in Martins et al.[42,43] If a projection shows poor quality, it should be discarded (Figure 1, step 2) and not used further in the workflow. Instead, the measures outlined in Martins et al.[42,43] should be used to improve projection quality. Conversely, if a projection shows good quality, it becomes an excellent candidate for assessing the visual separation between groups, and can be used further in the workflow (steps 3 and 4).

Feature selection will be performed in many of our experiments. We will select a subset of features $\mathcal{F}' \subseteq \mathcal{F}$ to investigate the effect of restricting the input of the DR technique to these features—that is, we will compare the projections of both $\mathcal{D}$ and $\mathcal{D}_{\mathcal{F}'}$. We perform feature selection/scoring using extremely randomized trees,[46] with 1000 trees in the ensemble. Scores are assigned to features based on their power to discriminate between two given sets of observations. As will become clear in the next sections, the choice of feature selection technique does not affect our proposal. Feature selection is always performed considering only the training set, as this allows assessing the generalization of the selection to the test set.

Learning algorithms will be used to evaluate whether good projections (with respect to perceived class separation) correspond to good classification systems. We consider three distinct algorithms: KNN (using Euclidean distances), SVMs (using radial basis function kernel),[47] and random forest classifiers (RFC).[48] These techniques were chosen for being widely used in both machine learning and representative of distinct classes of algorithms. Note that *any* other classification technique can be used together with our approach, since the techniques are treated as black-boxes, that is, we assume no knowledge of their inner workings.

Hyperparameter search is conducted by grid search on a subset of the hyperparameter space for each learning algorithm. Concretely, we choose the hyperparameters with highest average accuracy on fivefold cross-validation on the training set. For KNNs, the hyperparameter is the number of neighbors k (from 1 to 21, in steps of 2). For SVMs, the hyperparameters are C and γ (both from 10^{-10} to 10^{10}, in multiplicative steps of 10). For RFCs, the hyperparameters are the number of estimators (10 to 500, in steps of 50) and maximum tree depth (from 1 to 21, in steps of 5). In the next sections, we use the term *classifier* to refer exclusively to a particular combination of learning algorithm and hyperparameters trained on the entire training set. The hyperparameters are always found by the procedure outlined in the previous paragraph. In summary, following good machine learning practice, the test set does not affect the choice of hyperparameters.

Classification results are always quantified, in this article, by the *accuracy* (AC, ratio between correct classifications and number of observations) on the test set.

Presentation of experiments is uniform across datasets. For each experiment, a high-level claim is first stated. This claim is followed by supportive images, showing projections and classification results. In several cases, some aspect of the problem is altered (e.g. features or observations under consideration), and we show how our projections reflect the expected outcome.

Limitations of our study are discussed in section "Discussion."

Madelon dataset

Data. *Madelon* is a synthetic dataset created by Guyon et al.,[49] which contains 500 features and 2 class labels. We split the Madelon training set into training (1332 observations) and test (668 observations) sets, following our experimental protocol. The number of observations in each class is balanced. This artificial dataset was created specifically for the NIPS 2003 feature selection challenge. Only 20 of the 500 features are informative, that is, useful for predicting the class label. According to its authors, this dataset was designed to evaluate feature selection techniques when features are informative only when considered in groups.[49]

Goal 1. Our first goal is to show that, for this dataset, poor separation between classes in the projection corresponds to poor classification accuracy. While this correspondence may appear obvious, it is easy to show that it does not always hold (see section "Discussion"). Therefore, analyzing the link between visual separation and classification accuracy is worthwhile.

Consider the projection of the training data shown in Figure 2(a). The two class labels, represented by distinct colors, are not visually separated in the projection, as also shown by the low neighborhood hit of 53.9%.

If our projection is representative of the distances in the high-dimensional space, it is natural to interpret Figure 2(a) as evidence that the classification problem is hard, at least if the learning algorithm being used is based on distances. We will show that, for this example, this observation holds even for learning algorithms that do not directly work with distances in the high-dimensional space. This characteristic is crucial if we want to use projections as visual feedback about the efficacy of classification systems that use such algorithms.

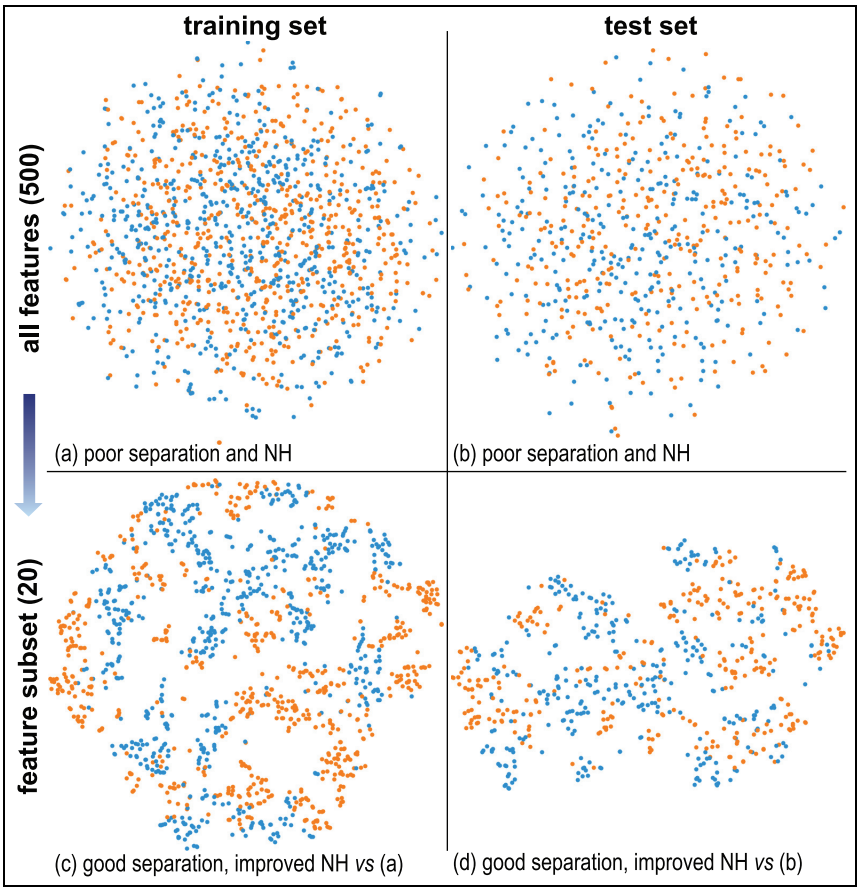

Figure 2. *Madelon* dataset: (a) training set (NH: 53.9%), (b) test set (NH: 50.97%), (c) training set, feature subset (NH: 83.56%), and (d) test set, feature subset (NH: 74.15%).

Figure 2(b) shows the projection of the test data, which also has a low neighborhood hit (NH) and poor separation. Following the experimental protocol outlined in the previous section for hyperparameter search, consider the best (in terms of average cross-validation accuracy) classifier for each learning algorithm. If the hypothesis about the difficulty of this classification task is true, the expected result would be a low accuracy on the test data.

Figure 3(a) and (b) shows the classification results for KNN (54.94% accuracy) and RFC (66.17%). The SVM classifier achieved 55.84% accuracy and is not shown due to space constraints. Triangles in the scatterplots show misclassified observations, colored based on their misclassification. The accuracies on the test set are considerably low, and both KNN and SVM perform close to random guessing.

Goal 2. Although these results show that the poor visual separation is correlated to a low classification accuracy, nothing we have shown so far tells that good separation relates to high accuracy. Let us investigate this next, specifically showing how we can select an appropriate subset of features to get a good class separation.

Using extremely randomized trees as a feature scoring technique, consider a subset containing 20 of the original 500 features, chosen based on their discriminative power in the training set. In other words, we chose the best features $\mathcal{F}' \subseteq \mathcal{F}$ to separate the two classes in the high-dimensional space. Figure 2(c) shows the projection of the training set restricted to these features. Compared to the previous projection of the training set (Figure 2(a)), the NH has improved considerably, and the visual separation has also improved. This visual feedback gives evidence that the classification task may become easier using a feature subset.

Figure 2(d) shows that feature selection also enhances the visual separation of the test set. Therefore, the visual separation after feature selection *generalizes* well to the test data.

The final question is whether the good visual separation corresponds to higher accuracy in the test

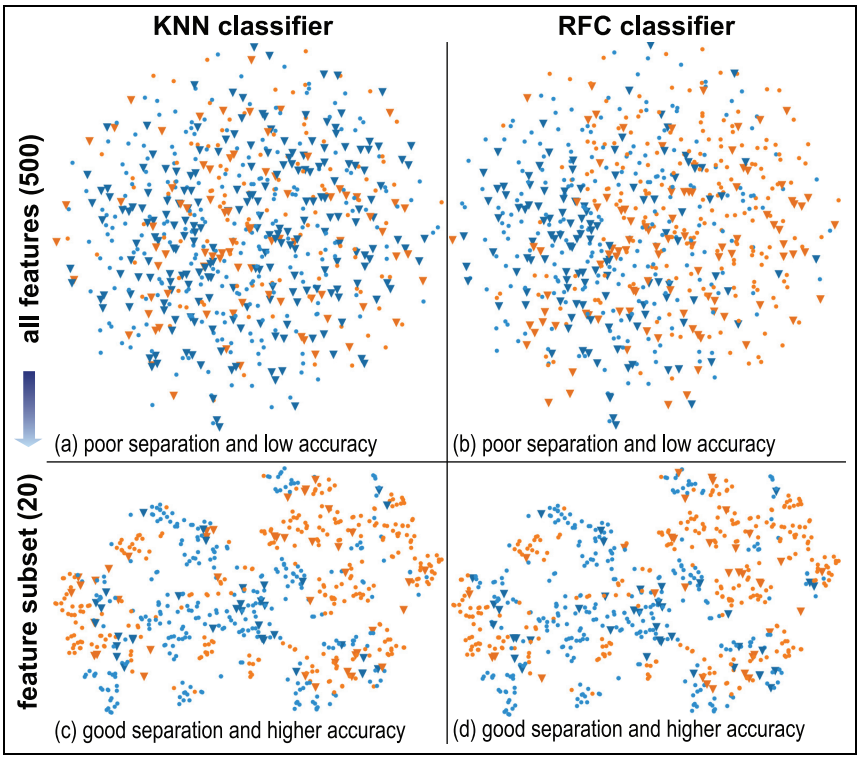

Figure 3. *Madelon* classification: (a) KNN (AC: 54.94%), (b) RFC (AC: 66.17%), (c) KNN, feature subset (AC: 88.62%), and (d) RFC, feature subset (AC: 88.92%).

set. Figure 3(c) and (d) confirms this hypothesis. Note that, after feature selection, both learning algorithms have greatly improved their results on the test set, with an increase of nearly 34% for KNN and 22% for RFC. In comparison, the neighborhood hit increased by almost 24% for the test set and by almost 30% for the training set. A similar increase happens in the case of the SVM, which goes from 55.84% to 86.68% test accuracy after feature selection. In other words, as could be expected, removing irrelevant features considerably enhances the generalization capacity of the learned model.

Even more interestingly, after feature selection, we see that the misclassified observations in the test set are often surrounded by points belonging to a different class (see triangular glyphs in Figure 3(c) and (d)). Thus, these observations could be interpreted as *outliers* according to the projection. Such feedback is *hard* to obtain from a traditional machine learning pipeline and is valuable for understanding classification system malfunction. Manually inspecting misclassified observations and their neighbors without the help of visualization would be very time-consuming and would not convey nearly as much insight about the *structure* of the data. Alternatives such as confusion matrices, for example, are difficult to interpret even for a modest number of classes (a confusion matrix for a 10-class problem has 45 independent values). The feedback presented by projections can, for example, prompt the users to consider special cases in their feature extraction pipeline.

Findings. In summary, the use case presented in this section shows how projections can predict classification system efficacy. In this use case, poor visual separation matches low classification accuracy, and good visual separation matches high classification accuracy. Furthermore, points that appear as outliers in a projection are often difficult to classify correctly. As we already mentioned in section "Proposed approach," previous studies showing these insights at an *observation level* are missing from the literature, making it unclear exactly whether and how insights provided by projections are useful. Such study is crucial to establish projections as an appropriate vehicle for visual feedback, which is basis of the interactive approach proposed in section "T2: improving system efficacy."

Melanoma dataset

Data. The *melanoma* dataset contains 369 features extracted from 753 skin lesion images, which are part of the EDRA atlas of dermoscopy,[50] using the feature

Figure 4. Melanoma dataset: (a) training set (NH: 64.87%), (b) test set (NH: 62.35%), (c) training set, feature subset (NH: 72.38%), and (d) test set, feature subset (NH: 62.55%).

extraction methods described in Feringa.[51] Class labels correspond to benign skin lesions (485 images) and malignant skin lesions (268 images). Note the considerable class imbalance in favor of the benign lesions.

Goals. The main goal of the experiments performed using this real-world dataset is to show the type of feedback that can be obtained through projections when the classification problem is difficult and the visual class separation is poor.

Figure 4(a) shows the projection of the training data. We see that the separation between classes is poor, which is confirmed by a low NH. Consider the set of 20 best features to discriminate between the two groups in the training set, according to extremely randomized trees. The corresponding projection of the training data restricted to these features is shown in Figure 4(c). Arguably, the separation is slightly improved, which is confirmed by a higher NH value.

Figure 4(b) and (d) shows the projections of the test data before and after feature selection, respectively. The poor separation is confirmed in the test data. More importantly, the separation does not seem to be better in the test set after feature selection. In other words, feature selection does not appear to have generalized particularly well to the unseen (test) data. From this evidence, we naturally suspect that classification accuracy is poor, and that feature selection will not enhance accuracy. Our next experiments confirm this suspicion.

Figure 5(a) displays the classification results on the test set obtained by the most effective learning algorithm (SVM, according to our protocol), using all the features. The class unbalance of the data places the expected accuracy of always guessing the most frequent class at 64%. Hence, an accuracy of 77.69% is not quite satisfactory. KNN also performs poorly, achieving only 73.71% accuracy (Figure 5(b)). This is evidence that the classification task is hard.

Figure 5(c) and (d) shows the classification results obtained after feature selection. As we see, feature selection improved the efficacy of the KNN classifier (from 73.71% to 77.69%) to the same level as an SVM using all features. However, the SVM results deteriorated after feature selection.

Furthermore, note the uniformity of blue classifications in the center of the projections shown in Figure 5(c) and (d). This confirms that distances in the projection are good indicators of classifier behavior in this case, even when the learning algorithm does not directly use distances in the high-dimensional feature space (Figure 5(c)).

As anticipated by the projection, feature selection did not improve generalization efficacy. Even so, reducing the number of features to approximately 5% of the original has benefits in computational efficiency and knowledge discovery. The reduced set of features

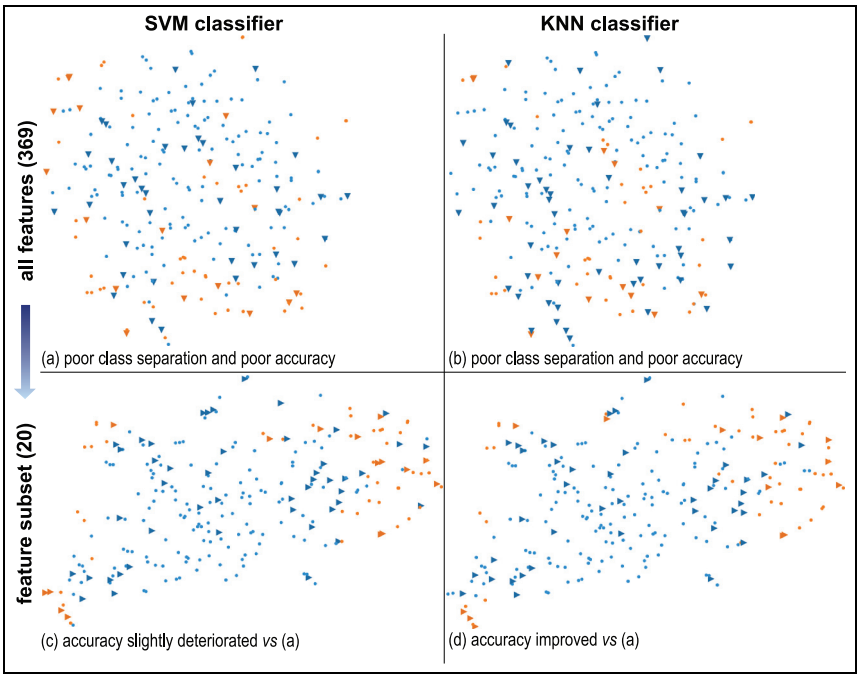

Figure 5. Melanoma classification: (a) SVM (AC: 77.69%), (b) KNN (AC: 73.71%), (c) SVM, feature subset (AC: 74.9%), and (d) KNN, feature subset (AC: 77.69%). The uniformity of blue classifications in the center of the projections shown in (c) and (d) confirms that distances in the projection are good indicators of classifier behavior.

contains valuable information to the system designer and indicates characteristics of the problem where designers may decide to focus their efforts. In other words, the use of feature selection, while not directly improving classification system accuracy, added value by reducing costs through data reduction.

Corel dataset

Data. The *Corel* dataset contains 150 scale-invariant feature transform (SIFT) features extracted from 1000 images by Li and Wang.[52] Class labels correspond to 10 image types: Africa, beach, buildings, buses, dinosaurs, elephants, flowers, horses, mountains, and food. The dataset is perfectly balanced between classes.

Goals. This experiment shows that projections can give insight into class-specific behavior, and also provides more evidence that projections can predict classification accuracy.

Figure 6(a) and (b) shows projections of the training and test data, respectively. Except for a *confusion zone* between the classes marked as green, orange, yellow, and brown, both projections show well-separated clusters. This separation is confirmed by a high NH value in both cases.

These projections can be interpreted as evidence that the classification task is easy. Confirming this hypothesis, Figure 7(a) shows the classification results for the best classifier (RFC). As expected, the accuracy obtained is very high (91.81%), considering that this is a balanced 10-class problem. More interesting, however, is the fact that many classification errors occur in the confusion zone observed in the projection of the test set. Thus, conclusions drawn from the visual feedback about confusion zones in this training set do generalize to unseen (test) data. Note that the concept of confusion zone is only possible because the data are *spatially* represented. It is, to our knowledge, not possible to depict a confusion zone otherwise. This is another valuable characteristic of our proposed projection-based representation.

We also use this dataset to consider an alternative scenario for predicting system efficacy. This scenario shows, again, that projections may be reliable predictors of classification system behavior. Consider the best 10 features to discriminate class 4 (purple) from other classes according to extremely randomized trees. The projection of the data restricted to this set of features is shown in Figure 6(c). As expected, note how class 4 is very well separated (center left), while observations in the other classes are poorly separated from each other. This is confirmed by low NH values (28.68%) and perfect *binary* NH values, when class 4 is considered against the rest. Figure 6(d) confirms that this characterization generalizes to the test data.

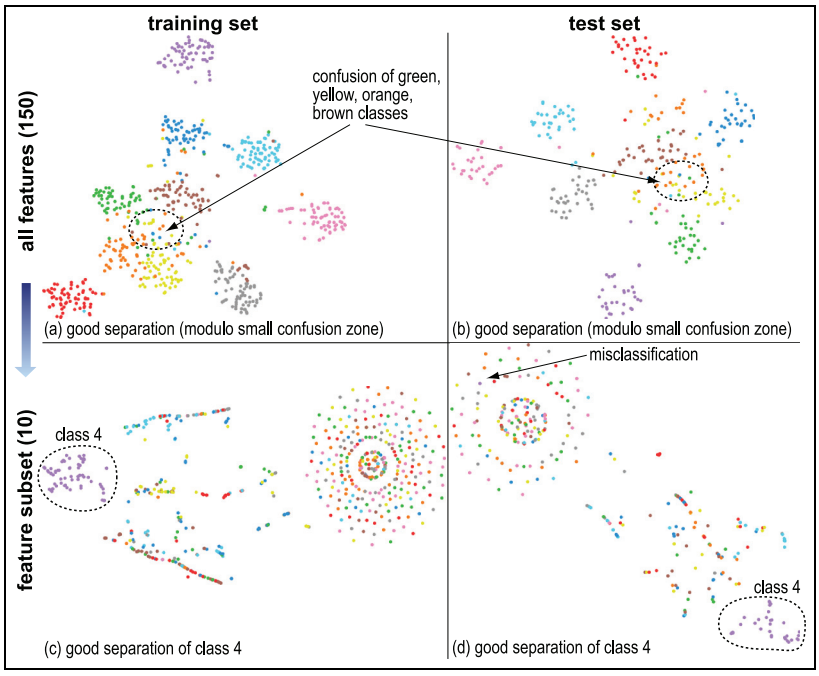

Figure 6. Corel dataset: (a) training set (NH: 85.7%), (b) test set (NH: 82.73%), (c) training set, feature subset (NH: 28.68%, 4 vs rest NH: 100%), (d) test set, feature subset (NH: 22.18%, 4 vs rest NH: 99.34%). Consult text on Figure 6(d) misclassification.

The poor separation between classes other than 4 leads us to expect poor accuracy results. Figure 7(b) shows the classification results using the features selected to separate class 4 from the rest, in the multi-class problem, which confirms this expectation. In contrast, the binary classification accuracy is *almost perfect* (99.7%, image omitted for brevity). There is a single mistake in the binary classification, which is placed in the top-left corner of the projection (top left of Figure 6(d)). The projection was also able to predict the existence of this outlier.

Parasites dataset

Data. The *parasites* dataset contains 9568 observations and 260 traditional image features extracted from (pre-segmented) objects in microscopy images of fecal samples.[53] We restricted ourselves to a subset of the original data that contains only the protozoan parasites (divided into six classes) and impurities (objects that should be ignored during analysis). Almost 60% of the observations correspond to impurities, which gives a significant class imbalance.

Goal. We present here one last example of the predictive power of projections, using a medium-sized realistic dataset. In this case, the projection reveals the presence of a large number of confounding observations that, when removed, increase classification accuracy.

Figure 8(a) displays the projection of the training set. We immediately see that impurities (marked pink) spread over almost the entire projection space. This is also seen in the projection of the test set (Figure 8(b)). In other words, we have weak evidence that the impurities may be confounded with almost every other class.

Can the other classes be reasonably well separated from each other when impurities are ignored? Figure 8(c) and (d) shows the projections of the training and test data, respectively, when the impurities are removed from the data. Therefore, our question is answered positively.

Considering again all observations, Figure 9(a) shows classification results for the best classifier (SVM, according to our protocol). Given the perceived poor visual separation, this result may be considered surprisingly good, which shows that perceived confusion is not definitive evidence. In section "Discussion," we will show an extreme example of this behavior. In a number of cases, however, we have seen that the evidence is much stronger in the other direction: when the perceived visual separation between classes in a projection is good, the classification results are also good.

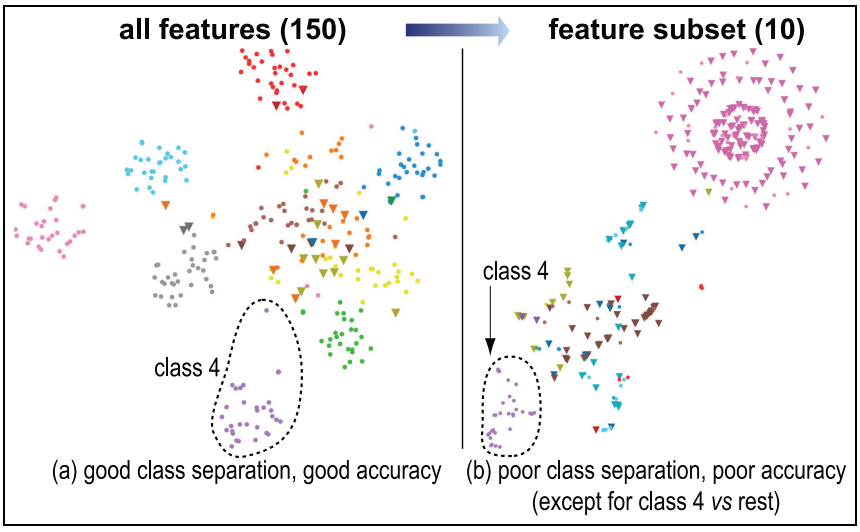

Figure 7. Corel classification: (a) RFC (AC: 91.81%) and (b) RFC, feature subset (AC: 34.55%, 4 vs rest AC: 99.7%).

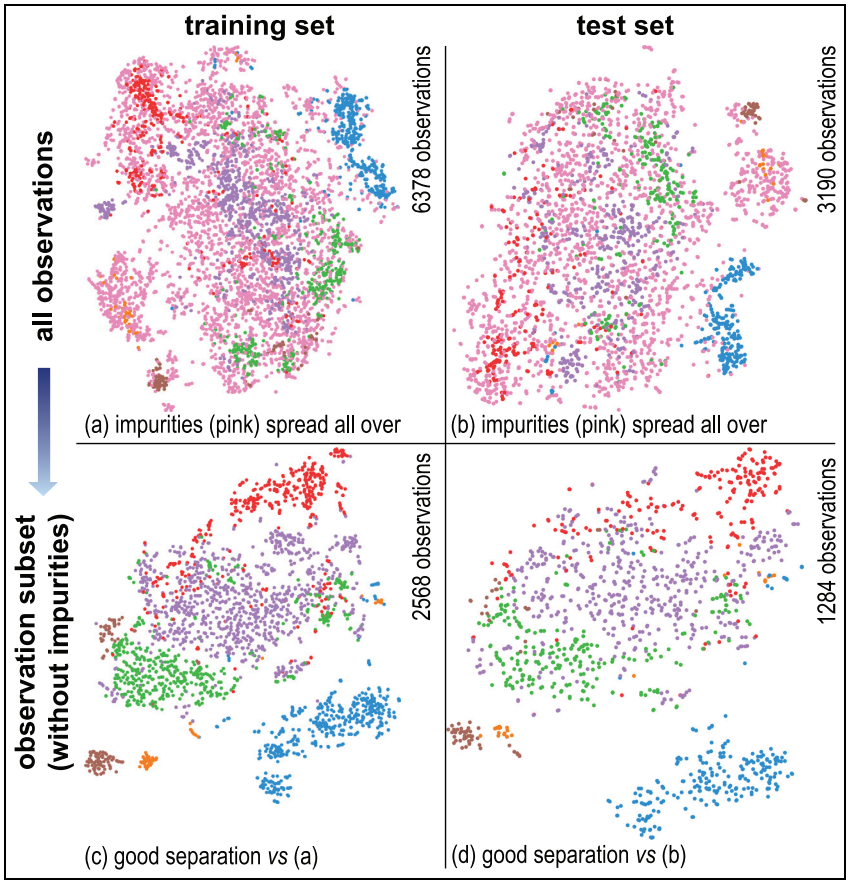

Figure 8. Parasites dataset: (a) training set (NH: 74.35%), (b) test set (NH: 68.49%), (c) training set, observation subset (NH: 87.22%), and (d) test set, observation subset (NH: 82.31%).

Consider next our dataset restricted to all the classes except impurities. Figure 9(d) shows KNN classification results, which are improved from 82.29% to 89.49% accuracy. However, SVM results are not significantly improved in this restricted task (approximately 2% accuracy increase). Once again, note how

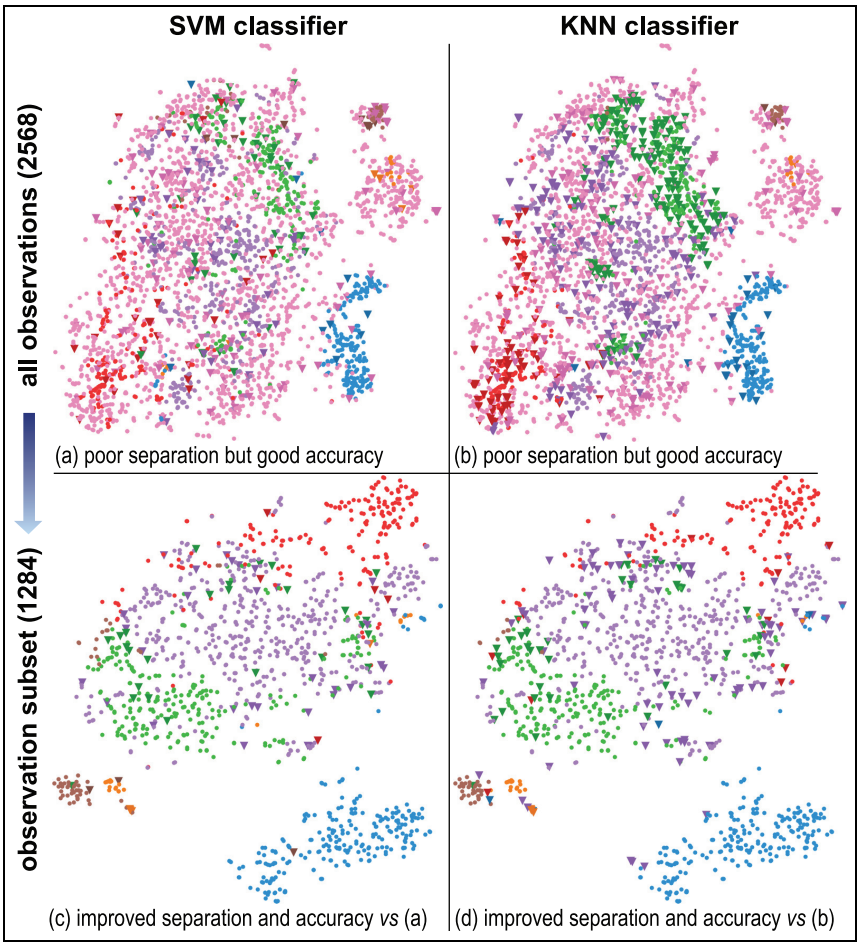

Figure 9. Parasites classification: (a) SVM (AC: 92.7%), (b) KNN (AC: 82.29%), (c) SVM, observation subset (AC: 94.55%), and (d) KNN, observation subset (AC: 89.49%).

the confusion zones contain the majority of misclassifications. Apparently, the SVM learning algorithm is able to deal better with the confusion between impurities and parasites. In this case, the projection was better to anticipate the behavior of the distance-based learning algorithm.

This is the largest dataset considered in our experiments. Note that the projections of the training and test sets are somewhat similar (e.g. Figure 8(c) and (d)). This highlights the importance of using representative datasets to study a problem using projections.

The difficulty of separating impurities from other classes could also be diagnosed from a confusion matrix. In practice, this insight could be used by the designer to study the classification of impurities as a separate problem. However, projections provide a more compelling visual representation of the same phenomenon, allowing the designer to inspect the observations in confusion zones. Such spatial information about relationships is lost in a confusion matrix.

As a last example for this section, we now show how additional visual feedback may be encoded into a projection.

Consider the *aggregate projection error*, a per-point metric of distance preservation after DR.[42] Intuitively, a point has a high aggregate error when its corresponding high-dimensional distances to the other observations are poorly represented by the low-dimensional distances in the projection. This feedback about the quality of a given projection is also important to our methodology.

Figure 10(a) shows the aggregate error for the parasites test set restricted to non-impurities (higher errors in darker colors). We see a point near the center of the projection with a relatively high aggregate error (square in Figure 10(a)). As colors map relative errors, this does not necessarily mean that the absolute aggregate error is high. Yet, this point is clearly an *outlier* in aggregate error when compared to its low-dimensional neighbors. In Figure 10(b), we see that the point is

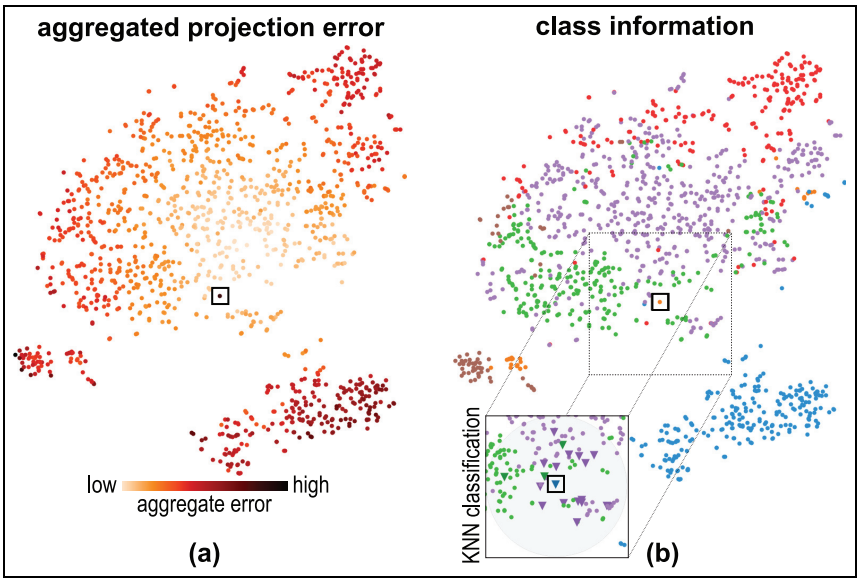

Figure 10. Parasites test set, observation subset: (a) aggregate error and (b) original classes, inset showing KNN classification.

surrounded by points belonging to other classes. By our definition, this point is an outlier with respect to its positioning given its class label. Note that the aggregate error is computed without any information about class labels and also draws attention to this particular observation.

One possible explanation for a high aggregate error is that the projection placed a point in a poor manner. In fact, the point is correctly classified by RFC and SVM, which weakly supports this hypothesis. However, KNN classified the point incorrectly (see inset in Figure 10(b)). Therefore, it is still unclear whether this point is a true outlier in the feature space. However, the error visualization was successful in focusing attention into an *interesting* observation, which warrants further inspection of its characteristics and features.

Several other error metrics and visual depictions of projection quality could be employed to enable similar feedback and help interpreting projections.[42,54]

Task 1: conclusion

The experiments performed for the four datasets in this section support our claim that projections can provide useful visual feedback about the ease of designing a good classification system. This visual feedback helps finding outliers, overall separation between observations in distinct classes, distribution of observations of a given class in the feature space, and presence of neighborhoods with mixed class labels (confusion zones). Arguably, the first two tasks have the most well-developed traditional feedback mechanisms: outlier detection, manual misclassification inspection, efficacy measures, and confusion matrices. The qualitative nature of the last two tasks makes them more difficult. This makes a strong case for the use of projections, even if there is no hard guarantee that the visual feedback offered by projections is definitely helpful for a given dataset. In section "Discussion," we present an extreme example of this issue.

T2: improving system efficacy

The previous section showed how projections can be useful for predicting classification system behavior. If a particular system performs well, there is no further effort required from the system designer. Instead, consider a classification system that generalizes poorly to unseen data. Because the design space (feature descriptors, learning algorithms, and hyperparameters) is immense, designers can benefit from insightful feedback about their choices. In that case, we have already shown that qualitative feedback from projections can be highly valuable.

Building on the use of projections for the first task (*T1*), this section focuses on the use of projections for the task of improving classification system efficacy (*T2*). In section "Proposed methodology and tooling," we present our significant extension of the visual feedback methodology proposed in Rauber et al.,[55] which enables *T2*. In sections "Madelon: relationship between relevant features," "Melanoma: alternative

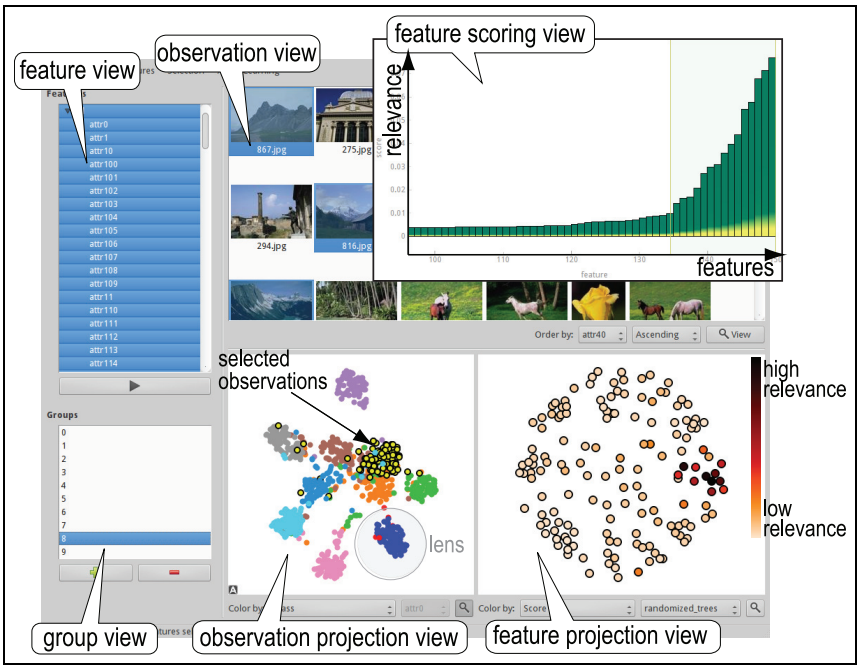

Figure 11. Feature exploration tool, showing the Corel dataset. The figure shows the observation view, feature view, group view, observation projection view (lensing observations, colored by classification; yellow observations are selected), feature scoring chart (showing best features to discriminate yellow class vs rest), and feature projection view (showing best features to discriminate yellow class vs rest, using a heat colormap).

feature scores," and "Corel: class-specific relevant features," we describe use cases that employ this methodology.

Proposed methodology and tooling

Our methodology for classification system improvement through interactive projections is implemented into a tool (available at: http://www.cs.rug.nl/svcg/People/PauloEduardoRauber-featured) composed of six linked views (Figure 11) as follows.

The observation view shows the image associated to each observation x in the dataset $\mathcal{D}$, if any, which are optionally sorted by a feature of choice. This provides an easy way to verify if a feature corresponds to user expectations.

The feature view shows all features $\mathcal{F}$, optionally organized as a hierarchy based on semantic relationships. Within this view, users can select a feature subset $\mathcal{F}' \subseteq \mathcal{F}$ to further explore.

The group view allows the creation and management of arbitrary observation groups by direct selection in the observation view or in the observation projection view (discussed next). Initially, groups correspond to classes.

The observation projection view shows a scatterplot of the projection of $\mathcal{D}_{\mathcal{F}'}$, the dataset composed of all observations restricted to the currently selected feature subset $\mathcal{F}'$. Points can be colored by a user-selected characteristic (such as class label or feature value) and are highlighted to show the selected set of observations.

Figure 11 also illustrates lensing, which optionally displays secondary characteristics on a neighborhood. In this particular case, the secondary characteristic is classification outcome (correct classifications in blue, incorrect in red).

The feature scoring chart ranks the features in $\mathcal{F}'$ by a relevance metric chosen by the user. We provide a variety of feature-selection techniques, including extremely randomized trees (which we employed in section "T1: predicting system efficacy"),[46] randomized logistic regression,[56] recursive feature elimination,[57] and others. The feature scoring view also allows the user to select a subset of $\mathcal{F}'$ through interactive rubber-banding.

The feature projection view is a new addition to the tool presented in Rauber et al.[55] Each point in this view corresponds to a feature in $\mathcal{F}$. Features are placed in 2D by a technique that tries to preserve the structural *similarity* between features. For our purposes, we define the dissimilarity $d_{i,j}$ between features i and j as $d_{i,j} = 1 - |r_{i,j}|$, where $r_{i,j}$ is the (empirical) Pearson correlation coefficient between features i and j. This

dissimilarity metric captures both positive and negative linear correlations between pairs of features. The dissimilarity matrix, which contains the dissimilarity between all pairs of features, can be represented in two dimensions by a *projection*, which is analogous to the projection of observations. As already mentioned in section "Related work," similar visualizations already exist in the literature.[28,29] However, we combine the feature projection view with task-specific information in a novel manner, as shown in the next sections.

We chose (absolute metric) MDS[58] to compute feature projections. According to preliminary experiments, MDS presents more coherent relationships between features and classes than t-SNE, which is important in the next sections. This is probably due to the difference in goals between the two techniques: absolute metric MDS attempts to preserve (global) pairwise dissimilarities,[58] while t-SNE is particularly concerned with preserving (local) neighborhoods.[13] Alternative (dis)similarity metrics between features are also available in the tool, including mutual information, distance correlation, and Spearman's correlation coefficient. The feature projection view provides a counterpart to the observation projection view and enables several interactions that will be detailed in the next sections.

Our visual analysis tool is implemented in Python and uses numpy,[59] scipy,[60] pyqt, matplotlib,[61] skimage,[62] sklearn,[63] pyqtgraph, and mlpy.[64]

The next sections describe how our tool can be used to support the task of classification system improvement based on visual feedback obtained from both observation and feature projections. For an overview of tool usage, see section "Proposed workflow."

Madelon: relationship between relevant features

Goal. In this section, we illustrate how the feature projection view can be used to select features by considering relationships between relevant features. As already mentioned, feature selection is a major challenge in classification system design. In particular, insight into the feature space can be very valuable when hand-engineered (off-the-shelf) features are used.

Consider a selection of the 20 best features to discriminate between the two classes of the Madelon dataset (section "Madelon dataset"), performed using the feature scoring chart based on extremely randomized trees. The corresponding projections of observations and features are shown, respectively, in Figure 12(a) and (b). Each feature in the feature projection view is colored according to its relevance score (darker colors represent higher relevance according to extremely randomized trees). The 20 selected features are outlined in black. Note that the most relevant selected features (darker colors) are placed near the center of the feature projection, except for the least relevant one. This finding is notable, since the feature projection is created without any information about feature scores. This shows that relevant features are related (according to the feature dissimilarity and relevance scoring metrics) in this dataset. Note that, in general, relevant features are not necessarily related.[3] For instance, a feature can simply *complement* the discriminative role of other features.

Showing the relationships between feature scoring and feature similarity is a main asset of the feature projection view. Figure 12(c) and (d) shows how such insight can be used: by removing the *outlier* feature (i.e. the feature that is apparently unrelated to the rest of the selection), visual separation is preserved. In other words, the feature projection view let us *prune* the feature space while maintaining the desired visual separation (and NH), thereby reducing the size of the data that needs to be considered next.

Improvement. Table 1 presents the results of each learning algorithm on the Madelon test set, following the protocol described in section "Experimental protocol." Experimental protocol, before and after removing the outlier feature mentioned above. As conveniently anticipated by the observation projection of the training set (Figure 12(c)), the classification efficacy is maintained (and perhaps slightly improved). In summary, the feature removal suggested by the feature projection view has reduced the data size, but maintained classification accuracy.

Corel: class-specific relevant features

Goal. This section shows how the feature projection view can be used together with the observation projection view to find class-specific relevant features, using the Corel dataset (section "Corel dataset") as an example. When improving system efficacy, such information is useful both for feature selection and for understanding classification system behavior.

We already showed (section "Corel dataset") that we can choose features that are good to discriminate one of the classes in the Corel dataset (class 4, which corresponds to dinosaur drawings) while making discrimination between the other classes very difficult. Figure 13(a) and (b) shows the corresponding observation and feature projections. Once again, we see that the discriminative features are highly related.

Consider an analogous feature selection aimed to discriminate class 3 (bus pictures) from the other

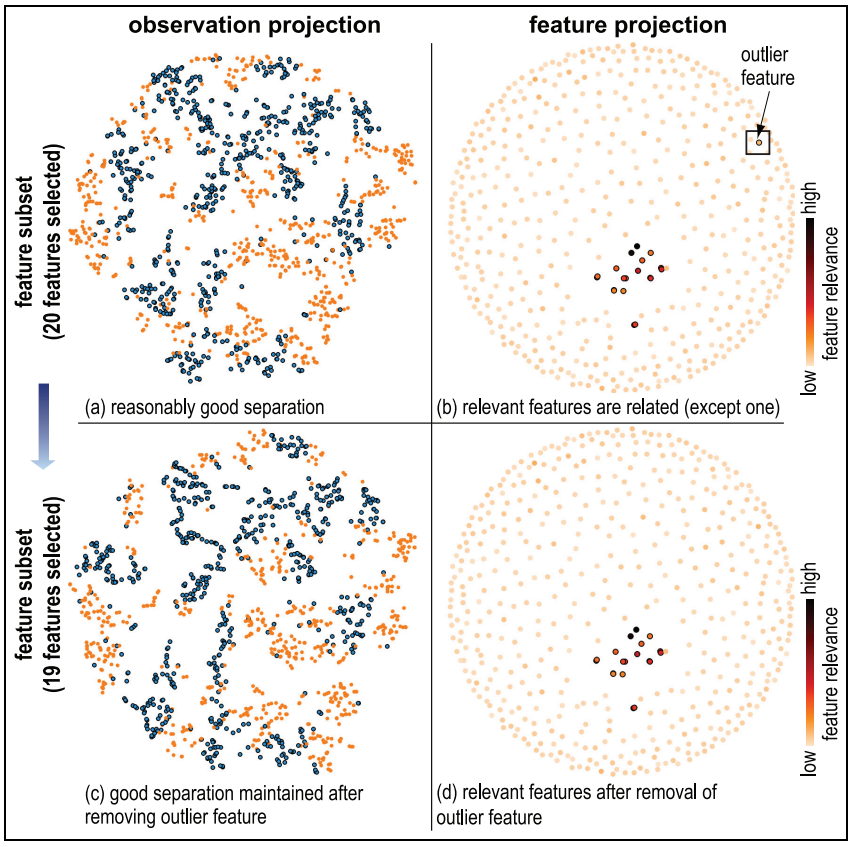

Figure 12. Madelon training set: (a, b) observation and feature projections, 20 features selected (NH: 83.56%), (c, d) observation and feature projections, one less feature (NH: 84.55%).

Table 1. Madelon test set accuracies, feature selection according to Figure 12.

Features/algorithm	KNN	RFC	SVM
20 features	88.62%	88.92%	86.68%
21 features	88.92%	88.92%	89.22%

KNN: *k*-nearest neighbors; RFC: random forest classifiers; SVM: support vector machine.

classes. Figure 13(c) and (d) shows the corresponding projections. Comparing the feature views (Figure 13(b) and (d)), we easily see that the sets of powerful discriminative features for the two classes are disjoint. This information could not be easily obtained from the feature-scoring bar chart mentioned in section "Proposed methodology and tooling," since features are generally difficult to locate in that visualization. As inspecting the precise ranking of each feature is easier in the bar chart, the two views are complementary. These interactions require very little effort from the user, who can inspect several feature combinations in a few minutes.

If the user is interested in a rough estimate of classification efficacy, our tool can also compute and display classification results (for a chosen learning algorithm) based on *k*-fold cross-validation. This process partitions the current data into k disjoint validation sets, and a classifier trained on the rest of the data is used to classify each validation set. Classification results for the distinct validation sets are aggregated and displayed, leading to images similar to Figure 7. These representations do not replace proper evaluation in a held-out test set (as in section "T1: predicting system efficacy" or the following paragraph), but are useful feedback sources during the interactive feature analysis process.

Improvement. Table 2 presents the result of each learning algorithm on the Corel test set, following the protocol in section "Experimental protocol," for the task of discriminating classes 3 and 4 from the rest (i.e. classes 3 and 4 are treated as a single class in a binary classification task) for all features and the subset of 26 features that were considered (separately) relevant for classes 3 and 4. As predicted by the observation projections of the training set shown in Figure 13(a) and (c), the

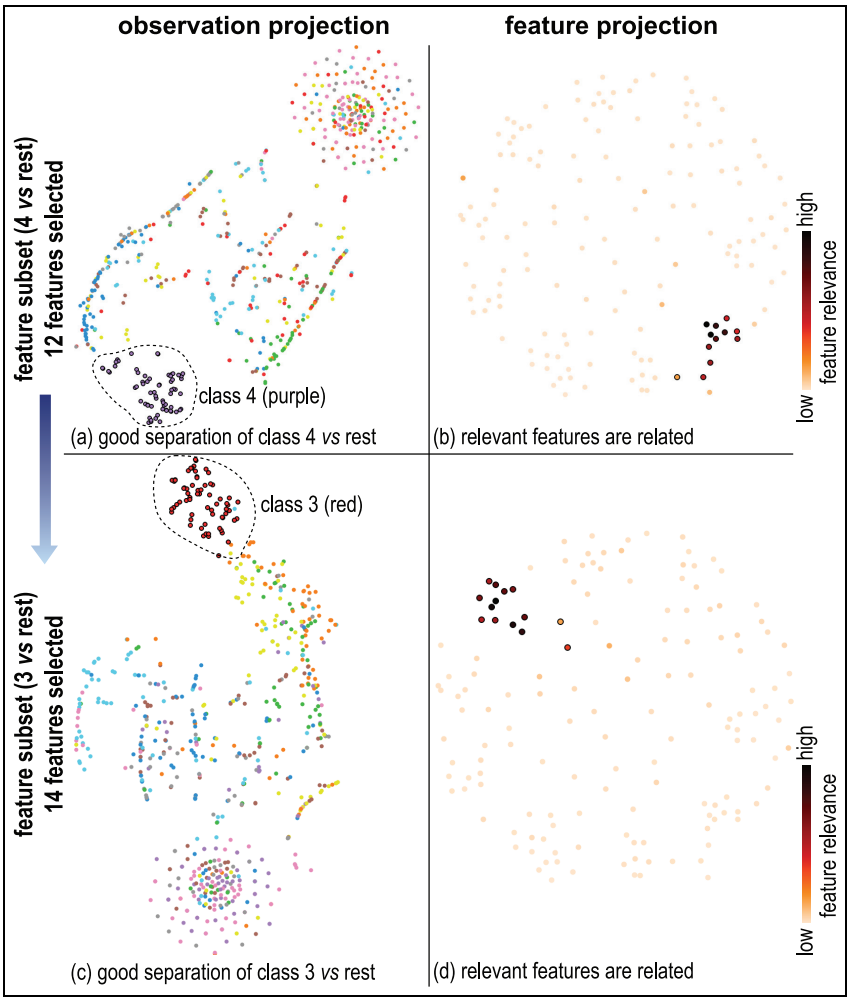

Figure 13. Corel training set: (a, b) observation and feature projections, feature subset (4 vs rest, Binary NH: 99.73%) and (c, d) observation and feature projections, feature subset (3 vs rest, Binary NH: 99.25%).

Table 2. Corel test set accuracies, classes 3 and 4 versus rest, relevant features according to Figure 13.

Features/algorithm	KNN	RFC	SVM
All (150) features	98.18%	98.79%	98.48%
26 features	98.48%	98.79%	98.79%

KNN: *k*-nearest neighbors; RFC: random forest classifiers; SVM: support vector machine.

classification efficacy is preserved. In summary, our visual analysis allowed us to prune the feature space from 150 to only 26 features, and construct a binary classifier for classes 3 and 4 versus rest that has the same quality as a classifier that uses all features.

Melanoma: alternative feature scores

Goal. The joint display of feature similarity and relevance is useful in other ways, as shown next. Here, our representation enables comparing the results of different feature-scoring techniques. Since the techniques are based on distinct principles, comparing their results to find features that are consistently considered effective is a valuable task for improving system efficacy.

Consider the feature projection view of the melanoma training set (section "Melanoma dataset") shown in Figure 14(a). As usual, colors represent the relevance of each feature to discriminate between the two classes present in the dataset (according to extremely randomized trees). We see a concentration of relevant features between the center and the bottom right. Again, the feature placement reinforces the feature-scoring information. The presence of *zones* of highly relevant features is highly suggestive for the exploration of the feature space, as shown in section "Madelon: relationship between relevant features."

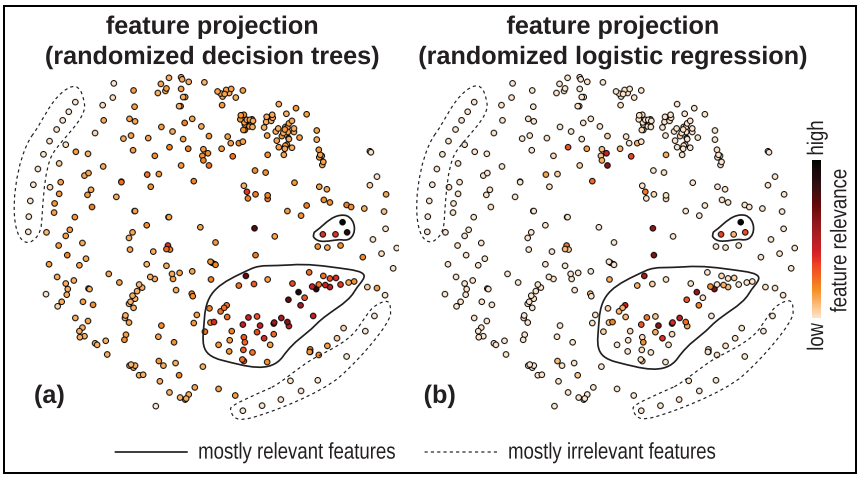

Figure 14. Feature projection for melanoma training set: (a) feature scoring by randomized decision trees and (b) feature scoring by randomized logistic regression.

Consider an alternative feature (relevance) scoring obtained by another technique—in this case, randomized logistic regression[56]—shown in Figure 14(b). We see that the distribution of relevancies is very different according to the second technique, which places higher cumulative relevance into fewer features. However, note that the two techniques agree on the irrelevance of the features in the bottom right and top left. This visual metaphor, where similar features are placed near each other, is a natural way to display such information.

The image features in this dataset have meaningful names, which can be inspected by hovering over the points. Using this mechanism, we find that the irrelevant peripheral points correspond mostly to histogram bins that have little (or even zero) variance across all images in our dataset. As expected, these features have almost no predictive power.

Improvement. Table 3 presents the result of each learning algorithm on the Melanoma test set, following the protocol in section "Experimental protocol," for all 369 features and the 58 (mostly) relevant features shown in Figure 14(a) and (b). Although the KNN and SVM results deteriorated slightly, the RFC result improved. Also, our analysis allowed us to discard a significant number of hand-engineered features. Besides saving significant time in feature extraction, the insight provided by our visual analysis of the feature space helps in deciding which types of features are most relevant for classification.

Table 3. Melanoma test set accuracies, relevant features according to Figure 14.

Features/algorithm	KNN	RFC	SVM
All (369) features	73.71%	76.49%	77.69%
58 features	73.31%	77.29%	76.10%

KNN: *k*-nearest neighbors; RFC: random forest classifiers; SVM: support vector machine.

Proposed workflow

We now summarize the value added by the insights described in sections "T1: predicting system efficacy" and "T2: improving system efficacy" by revisiting the high-level workflow outlined in Figure 1.

Our workflow begins when the user loads the data into our analysis tool and considers the observation projection. If the perceived class separation in this projection is good, the classification task is likely quite simple (as discussed in section "T1: predicting system efficacy"). As an extreme example, consider the projection of the Corel dataset, where even a 1-nearest neighbor algorithm in the 2D projection space would achieve good results. In such cases, the user can follow the traditional machine learning pipeline, with a high expectation that the system will perform well.

A more interesting scenario occurs when the perceived class separation in the projection is poor. In this case, the next step is to use the mechanisms provided by our tool to find a feature subset that brings separation. This may require several iterations of feature scoring, analysis, and backtracking. If no separation improvement can be found, there are two possible scenarios: classification efficacy is satisfactory (the projection is misleading with respect to classifier behavior) or unsatisfactory. The first case is easy to diagnose and consists of conducting experiments following the

traditional machine learning pipeline. The second case is the most complicated. In this case, we have shown that the qualitative aspects of our proposed visualizations are crucial in enabling the designer to diagnose the system. For this purpose, our tool provides mechanisms to detect the presence of outliers and confusion zones, and also to inspect classification results based on a visual metaphor that represents observations in a consistent way. By inspecting the observation projection, the designer receives visual feedback about which features are important to eliminate confusion between classes. Furthermore, using the feature projection view and feature scoring methods, the designer can reason about the discriminative power of features and focus effort on related (or complementary) feature descriptors. The new alternatives devised during this analytic process can be fed back into the tool, closing the cycle.

Discussion

This section discusses several important aspects of our proposed methodology and experiments.

Coverage

As any experimental study, many conclusions are limited to the datasets that we presented. The particular random choice of training and test data also affects the results, although the amount of data we considered diminishes this concern. Importantly, the extent of our validation (i.e. experimental protocol, number of datasets, and learning algorithms) is in line with most similar papers in visual analytics and machine learning.

While we have conducted experiments in additional datasets (not presented for the sake of space, considering our focus in qualitative aspects), the four datasets discussed in the text illustrate well all types of feedback that can be obtained from projections. We also experimented with other DR techniques,[14,15] but obtained the best predictive feedback from t-SNE.[45] Although the success of our methodology is dependent on choosing an appropriate DR technique, a comparative analysis between such techniques was considered beyond the scope of our work. Instead, we simply aim to show how a particular technique can be successfully combined with our methodology.

Our choice of learning algorithms for validation (KNN, RFC, and SVM) considers their widespread popularity and aims to make our approach appealing to a large number of practitioners. The positive results obtained with these highly distinct algorithms suggest that our approach is valuable for other learning algorithms.

Limitations

It is easy and instructive to construct a synthetic example where projections do not provide valuable visual feedback for classification system design. This is described next. Consider the task of classifying observations sampled from two 10-dimensional parallel (affine) hyperplanes that correspond to distinct classes. Consider also that the distance between these hyperplanes is small when compared to the expected distance between any pair of neighboring sample elements from the same hyperplane. By construction, this classification task is very easy for a linear SVM, which consistently obtains 100% accuracy following the experimental protocol detailed in section "Experimental protocol." At the same time, a DR technique that tries to preserve the original distances in the high-dimensional space will not show a clear separation between the two classes, as shown in Figure 15. In simple terms, the visual feedback is misleading, because the classification task is easy, but there is no apparent visual separation between classes. It is important to note that other learning algorithms did not perform well on this test set (KNN: 51.20%, RFC: 54.94%). However, we believe it is also possible to construct examples where the visual feedback is unhelpful for those algorithms.

Despite this worst-case behavior, we argue that the results presented in sections "T1: predicting system efficacy" and "T2: improving system efficacy" support our claims that our proposed approach is highly valuable, particularly considering the very low investment necessary to explore data by our proposed methodology and tooling.

Scalability

Our feature space exploration approach benefits from the visual scalability of projections to hundreds of thousands of high-dimensional observations and hundreds of dimensions, although visual clutter eventually becomes an issue for the quality of the visual feedback.

Even in cases where features are difficult to interpret, we have shown that our methods can be used to effectively support the tasks *T1* and *T2*. However, more study is needed to assess how suitable our methods are for datasets containing thousands of features.

The computational scalability limits are imposed by the requirement of near-interactive response times. For instance, considering a dataset (Madelon) composed of $N = 2000$ observations and $D = 500$ dimensions, the tool employed in our experiments requires approximately 20 s to compute both observation and feature projections and present them for exploration using a typical desktop computer (3.5 GHz Linux PC,

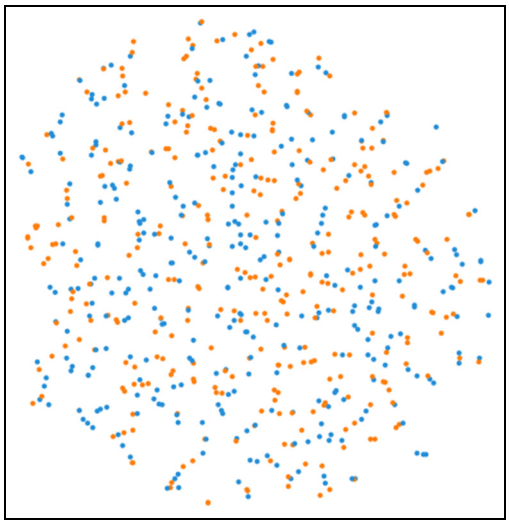

Figure 15. Plane classification, linear SVM (AC: 100%).

16 GB RAM). Clearly, the main bottleneck consists of recomputing such projections for different subsets of features. For some DR techniques,[13] this issue becomes significant in datasets containing more than a few thousands of observations, while others are able to deal with hundreds of thousands of observations at interactive rates.[14,15]

Conclusion

In this article, we have shown that projections are useful tools for predicting classification system efficacy in several real and synthetic datasets. The visual feedback given by projections is especially helpful in qualitative tasks. These tasks include inspecting the presence of outliers, overall separation between observations in distinct classes, distribution of observations of a given class in the feature space, and presence of neighborhoods with mixed class labels.

We also introduced a methodology that uses projections as a basis for an interactive system designed to give insight into the feature space. This methodology, and associated tooling, can aid a designer in improving classification systems, either directly (by suggesting features that should be eliminated from consideration) or indirectly (by providing feedback about which types of features are most important and for which observations). In particular, we showed how a projection representing observations can be integrated with an interactive representation of feature similarity to aid in this task.

As future work, we will consider studying further the connection between the observation and feature projections. We will also consider specific features of some DR techniques, such as control point positioning, which may be valuable for our methodology. Furthermore, we intend on providing visual support to semi-supervised learning tasks, such as active learning. Other important direction for future work consists of designing inverse mappings from the 2D observation projection to the feature space to allow users to synthesize improved features by interactively moving misclassified observations to their desired neighborhoods.

Acknowledgements

The authors would like to thank Dr M. Emre Celebi for providing the *Melanoma* dataset.

Funding

This study was financially supported by the CAPES, FAPESP (2012/24121-9, 2014/12236-1), CNPq (302970/2014-2, 479070/2013-0), and the Ubbo Emmius Fund (University of Groningen).

References

1. Murphy KP. *Machine learning: a probabilistic perspective*. London: MIT Press, 2012.
2. Deselaers T, Keysers D and Ney H. Features for image retrieval: an experimental comparison. *Inform Retrieval* 2008; 11(2): 77–107.
3. Guyon I and Elisseeff A. An introduction to variable and feature selection. *J Mach Learn Res* 2003; 3: 1157–1182.
4. Liu H and Yu L. Toward integrating feature selection algorithms for classification and clustering. *IEEE T Knowl Data En* 2005; 17(4): 491–502.
5. Krizhevsky A, Sutskever I and Hinton GE. ImageNet classification with deep convolutional neural networks. In: *Proceedings of the advances in neural information processing systems*, Lake Tahoe, 3–6 December 2012. New York: ACM.
6. Bengio Y. Learning deep architectures for AI. *J Found Trend Mach Learn* 2009; 2(1): 1–127.
7. Bengio Y. Practical recommendations for gradient-based training of deep architectures. In: Montavon G, Orr GB and Müller KR (eds) *Neural networks: tricks of the trade*. Berlin: Springer, 2012, pp.437–478.
8. Wolpert DH. The lack of a priori distinctions between learning algorithms. *Neural Comput* 1996; 8(7): 1341–1390.
9. Domingos P. A few useful things to know about machine learning. *Commun ACM* 2012; 55(10): 78–87.
10. Brehmer M, Sedlmair M, Ingram S, et al. Visualizing dimensionally-reduced data: interviews with analysts and a characterization of task sequences. In: *Proceedings of the fifth workshop on beyond time and errors: novel evaluation methods for visualization (BELIV)*, Paris, 10 November 2014. New York: IEEE.

11. Liu S, Wang B, Bremer PT, et al. Distortion-guided structure-driven interactive exploration of high-dimensional data. *Comput Graph Forum* 2014; 33(3): 101–110.
12. Lee JA and Verleysen M. Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomputing* 2005; 67: 29–53.
13. Van der Maaten L and Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008; 9: 2579–2605.
14. Paulovich FV, Nonato LG, Minghim R, et al. Least square projection: a fast high-precision multidimensional projection technique and its application to document mapping. *IEEE T Vis Comput Gr* 2008; 14(3): 564–575.
15. Joia P, Paulovich FV, Coimbra D, et al. Local affine multidimensional projection. *IEEE T Vis Comput Gr* 2011; 17(12): 2563–2571.
16. Caldas J, Gehlenborg N, Faisal A, et al. Probabilistic retrieval and visualization of biologically relevant microarray experiments. *Bioinformatics* 2009; 25(12): 145–153.
17. Chen L and Buja A. Stress functions for nonlinear dimension reduction, proximity analysis, and graph drawing. *J Mach Learn Res* 2013; 14: 1145–1173.
18. Liu S, Maljovec D, Wang B, et al. Visualizing high-dimensional data: advances in the past decade. *IEEE T Vis Comput Gr* 2017; 23(3): 1249–1268. DOI: 10.1109/TVCG.2016.2640960.
19. Heinrich J and Weiskopf D. *State of the art of parallel coordinates*. Eurographics state of the art reports, 2013, https://pdfs.semanticscholar.org/b85c/cac3e7c21741626 3edcc6c55db508b5c4c0d.pdf
20. Hoffman P, Grinstein G, Marx K, et al. DNA visual and analytic data mining. In: *Proceedings of the IEEE visualization*, Phoenix, AZ, 24 October 1997. New York: IEEE.
21. Chambers J, Cleveland W, Kleiner B, et al. *Graphical methods for data analysis*. Belmont, CA: Wadsworth, 1983.
22. Kandogan E. Star coordinates: a multi-dimensional visualization technique with uniform treatment of dimensions. In: *Proceedings of the IEEE information visualization symposium*, Salt Lake City, UT, 9–10 October 2000. New York: IEEE.
23. Rao R and Card SK. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: *Proceedings of the SIGCHI conference on human factors in computing systems*, Boston, MA, 24–28 April 1994. New York: ACM.
24. Becker R, Cleveland W and Shyu M. The visual design and control of trellis display. *J Comput Graph Stat* 1996; 5: 123–155.
25. Fadel S, Fatore F, Duarte F, et al. LoCH: a neighborhood-based multidimensional projection technique for high-dimensional sparse spaces. *Neurocomputing* 2014; 150: 546–556.
26. Sedlmair M, Munzner T and Tory M. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE T Vis Comput Gr* 2013; 19(12): 2634–2643.
27. Tatu A, Maas F, Farber I, et al. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In: *Proceedings of the IEEE conference on visual analytics science and technology (VAST)*, Seattle, WA, 14–19 October 2012, pp.63–72. New York: IEEE.
28. Yuan X, Ren D, Wang Z, et al. Dimension projection matrix/tree: interactive subspace visual exploration and analysis of high dimensional data. *IEEE T Vis Comput Gr*, 2013; 19(12): 2625–2633.
29. Turkay C, Filzmoser P and Hauser H. Brushing dimensions - a dual visual analysis model for high-dimensional data. *IEEE T Vis Comput Gr* 2011; 17(12): 2591–2599.
30. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th international joint conference on artificial intelligence (IJCAI)*, Montreal, QC, Canada, 20–25 August 1995. New York: ACM.
31. Fawcett T. An introduction to ROC analysis. *Pattern Recogn Lett* 2006; 27(8): 861–874.
32. Talbot J, Lee B, Kapoor A, et al. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In: *Proceedings of the SIGCHI conference on human factors in computing systems*, Boston, MA, 4–9 April 2009, pp.1283–1292. New York: ACM.
33. Van den Elzen S and Van Wijk JJ. Baobabview: interactive construction and analysis of decision trees. In: *Proceedings of the 2011 IEEE conference on visual analytics science and technology (VAST)*, Providence, RI, 23–28 October 2011, pp.151–160. New York: IEEE.
34. Schulz A, Gisbrecht A and Hammer B. Using discriminative dimensionality reduction to visualize classifiers. *Neural Process Lett* 2014; 42: 27–54.
35. Rauber PE, Fadel SG, Falcão AX, et al. Visualizing the hidden activity of artificial neural networks. *IEEE T Vis Comput Gr* 2017; 23(1): 101–110.
36. Paiva JGS, Schwartz WR, Pedrini H, et al. An approach to supporting incremental visual data classification. *IEEE T Vis Comput Gr* 2015; 21(1): 4–17.
37. Mühlbacher T, Piringer H, Gratzl S, et al. Opening the black box: strategies for increased user involvement in existing algorithm implementations. *IEEE T Vis Comput Gr* 2014; 20(12): 1643–1652.
38. Heimerl F, Koch S, Bosch H, et al. Visual classifier training for text document retrieval. *IEEE T Vis Comput Gr* 2012; 18(12): 2839–2848.
39. Hoferlin B, Netzel R, Hoferlin M, et al. Inter-active learning of ad-hoc classifiers for video visual analytics. In: *Proceedings of the IEEE conference on visual analytics science and technology*, Seattle, WA, 14–19 October 2012. New York: IEEE.
40. Krause J, Perer A and Bertini E. Infuse: interactive feature selection for predictive modeling of high dimensional data. *IEEE T Vis Comput Gr* 2014; 20(12): 1614–1623.
41. Brandoli B, Eler D, Paulovich F, et al. Visual data exploration to feature space definition. In: *Proceedings of the 23rd SIBGRAPI conference on graphics, patterns and images*

(SIBGRAPI), Gramado, 30 August–3 September 2010, pp.32–39. New York: IEEE.
42. Martins RM, Coimbra DB, Minghim R, et al. Visual analysis of dimensionality reduction quality for parameterized projections. *Comput Graph* 2014; 41: 26–42.
43. Martins RM, Minghim R and Telea AC. Explaining neighborhood preservation for multidimensional projections. In: *Proceedings of the computer graphics and visual computing (CGVC)*, London, 16–17 September 2015.
44. Gelman A, Carlin JB, Stern HS, et al. *Bayesian data analysis*, vol. 2. Abingdon: Taylor & Francis, 2014.
45. Van Der Maaten L. Accelerating t-SNE using tree-based algorithms. *J Mach Learn Res* 2014; 15(1): 3221–3245.
46. Geurts P, Ernst D and Wehenkel L. Extremely randomized trees. *Mach Learn* 2006; 63(1): 3–42.
47. Boser BE, Guyon IM and Vapnik VN. A training algorithm for optimal margin classifiers. In: *Proceedings of the fifth annual workshop on computational learning theory*, Pittsburgh, PA, 27–29 July 1992, pp.144–152. New York: ACM.
48. Breiman L. Random forests. *Mach Learn* 2001; 45(1): 5–32.
49. Guyon I, Gunn S, Ben-Hur A, et al. Result analysis of the NIPS 2003 feature selection challenge. In: *Proceedings of the 17th international conference on neural information processing systems (NIPS)*, Vancouver, BC, Canada, 13–18 December 2004, pp.545–552. New York: ACM.
50. Argenziano G, Soyer HP, De Giorgio V, et al. *Interactive atlas of dermoscopy*. Milan: EDRA Medical Publishing & New Media, 2000.
51. Feringa S. *Comparison of features used in automatic skin lesion classification*. Master's Thesis, Rijksuniversiteit Groningen, Groningen, 2015.
52. Li J and Wang JZ. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE T Pattern Anal* 2003; 25(9): 1075–1088.
53. Suzuki CT, Gomes JF, Falcão AX, et al. Automatic segmentation and classification of human intestinal parasites from microscopy images. *IEEE T Biomed Eng* 2013; 60(3): 803–812.
54. Aupetit M. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing* 2007; 70(7): 1304–1330.
55. Rauber PE, Silva RRO, Feringa S, et al. Interactive image feature selection aided by dimensionality reduction. In: *Proceedings of the EuroVis workshop on visual analytics*, Cagliari, 25–26 May 2015, http://dx.doi.org/10.2312/eurova.20151098
56. Meinshausen N and Bühlmann P. Stability selection. *J Roy Stat Soc B* 2010; 72(4): 417–473.
57. Guyon I, Weston J, Barnhill S, et al. Gene selection for cancer classification using support vector machines. *Mach Learn* 2002; 46(1–3): 389–422.
58. Borg I and Groenen PJ. *Modern multidimensional scaling: theory and applications*. New York: Springer Science + Business Media, 2005.
59. Van Der Walt S, Colbert SC and Varoquaux G. The NumPy array: a structure for efficient numerical computation. *Comput Sci Eng* 2011; 13(2): 22–30.
60. Jones E, Oliphant T and Peterson P. SciPy: open source scientific tools for Python, 2014, http://www.scipy.org
61. Hunter JD. Matplotlib: a 2D graphics environment. *Comput Sci Eng* 2007; 9(3): 90–95.
62. Van Der Walt S, Schönberger JL, Nunez-Iglesias J, et al. scikit-image: image processing in python. *PeerJ* 2014; 2: e453.
63. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in python. *J Mach Learn Res* 2011; 12: 2825–2830.
64. Albanese D, Visintainer R, Merler S, et al. mlpy: machine learning python (arXiv:1202.6548), 2012.