Class Visualization of High-Dimensional Data with Applications

Inderjit S. Dhillon
Department of Computer Sciences, University of Texas, Austin, TX 78712-1188, USA.
email: inderjit@cs.utexas.edu

Dharmendra S. Modha and W. Scott Spangler
IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099.
emails: {dmodha,spangles}@almaden.ibm.com

January 29, 2003

Abstract

Consider the problem of visualizing high-dimensional data that has been categorized into various classes. Our goal in visualizing is to quickly absorb inter-class and intra-class relationships.

Towards this end, *class-preserving projections* of the multidimensional data onto two-dimensional planes, which can be displayed on a computer screen, are introduced. These class-preserving projections maintain the high-dimensional class structure, and are closely related to Fisher's linear discriminants. By displaying sequences of such two-dimensional projections and by moving continuously from one projection to the next, an illusion of smooth motion through a multidimensional display can be created. We call such sequences *class tours*. Furthermore, we overlay *class-similarity graphs* on our two-dimensional projections to capture the distance relationships in the original high-dimensional space.

We illustrate the above visualization tools on the classical Iris plant data, the ISOLET spoken letter data, and the PENDIGITS on-line handwriting data set. We show how our visual examination of the data can uncover latent class relationships.

Keywords: Class-preserving projections, Classification, Class tours, Linear projections, Multidimensional visualization, Similarity graphs.

1 Introduction

Classification and clustering are central tools in machine learning and data mining, and are used in a variety of applications such as fraud detection, signal processing, time-series analysis and optical character recognition. Classification is the problem of assigning class labels to unlabeled data items given a collection of labeled data items, while clustering refers to the problem of assigning class labels to data items where no prior labeling is known. The *Yahoo!* hierarchy of the World-Wide Web is a prime example of the value of such class labelings (www.yahoo.com). There are many other examples where class labels are either assigned manually to the data items or obtained using clustering methods such as k-means or vector quantization [1, 2, 3]. In this paper, we will assume that class labels are assigned to each data item and will not worry where the labeling comes from. Hence, we will use the terms clustering and classification interchangeably, unless explicitly stated otherwise.

The assignment of class labels to individual data items conveys limited information. Indeed, as recognized in [4, p.269] :

> "The use of any clustering method necessarily confronts the user with a set of 'clusters' whether or not such clusters are meaningful. Thus, from a data-analytic viewpoint, there is a crucial need for procedures that facilitate the interpretation of the results and enable a sifting of useful findings from less important ones and even methodological artifacts. Formal tests for the statistical significance of clusters have been proposed but informal more data-oriented methods that make fewer assumptions are also needed".

Information visualization is one such effective and informal procedure. In this paper we propose a scheme for visually understanding inter-class and intra-class relationships. In addition to the existence of class labels for data items, we will assume that the data is embedded in high-dimensional Euclidean space R^d, and that proximity in R^d implies similarity. The data may naturally occur in this form, or *vector-space* models of the underlying data may be constructed, for example, voice, images or text documents may be treated as vectors in a multidimensional feature space, see [5, 6, 7, 8].

Our main aim in this paper is to visually understand the spatial relationships between various classes in order to answer questions such as:

1. how well-separated are different classes?
2. what classes are similar or dissimilar to each other?
3. what kind of surface separates various classes, for example, are the classes linearly separable?
4. how coherent or well-formed is a given class?

Answers to these questions can enable the data analyst to infer inter-class relationships that may not be part of the given classification. Additionally, visualization can help in gauging the quality of the classification and quality of the feature space. Discovery of interesting class relationships in such a visual examination can help in designing better algorithms and in better feature selection. We can term this visual process as *visual discriminant analysis*. More concretely, such an analysis would be useful while designing a classifier in a "pre-classification" phase, or in evaluating the quality of clusters in a "post-clustering" phase, where classification and clustering refer to the classical machine learning problems mentioned above.

In order to achieve the above objectives for large data sets that are becoming increasingly common, we believe the user must have access to a real-time, interactive visualization toolbox. The visualization system should allow the user to compute different local and global views of the data "on the fly", and allow seamless transitions between the various views. Thus, an overriding demand on our visualization algorithms is that they be computationally efficient in order to support such a system.

In view of the above goals, we propose the use of carefully chosen two-dimensional projections to display the data, augmenting them with similarity graphs and motion graphics. In more detail, the following are our main contributions in this paper.

1. **Class-preserving projections and class-eigenvector plots** are our main tools for visual explorations. Class-preserving projections and their generalizations, class-eigenvector plots, are linear projections of the data onto two-dimensional planes that attempt to preserve the inter-class structure present in the original multidimensional space R^d.

   The idea of projecting high-dimensional data to lower dimensions is an old trick that has been profitably exploited in many applications, see, for example, principal components analysis (PCA)[1], projection pursuit [9, 10], Kohonen's self-organizing maps (SOM)[11] and multidimensional scaling (MDS)[12, 13]. For visualization purposes, two-dimensional projections based on principal components, canonical correlations and data sphering were considered in [14]. Later, [15, 16] also incorporated projections based on projection pursuit. The non-linear Kohonen self-organizing maps were used for visualization in [17, 18, 19]. A detailed review of visualization schemes for high-dimensional data is given in [20]. However, most existing visualization schemes present only global projections of the data with the SOM and MDS providing exactly one global view. Moreover, the non-linear SOM and MDS projections and the linear PCA projections are computationally expensive, especially when the data is high-dimensional and sparse.

   In contrast, our projections offer an entire library of local and global views to the user, and are extremely efficient to compute. Our most expensive methods are the class-eigenvector plots which require computation of the two leading singular vectors of a $d \times k$ matrix, where d is the dimensionality of the data and k is the number of clusters. In comparison, PCA requires singular vector computations on the entire data set (a $d \times n$ matrix where n is the number of data points), which has a computational complexity of $O(d^2n)$. Typically n is much larger than k as can be seen from two sample data sets we will use later in this paper. In the ISOLET speech recognition data set, $k = 26$, $n = 7797$ and $d = 617$, while in the PENDIGITS example $k = 10$, $n = 10992$ and $d = 16$. The non-linear SOM and MDS projections are even more expensive to compute than PCA. Our superior efficiency is essential for interactive visualization and has allowed us to build a real-time visualization software toolbox where many local and global projections can be computed on the fly. Also note that PCA and MDS are often unable to visually discriminate between classes as they do not even use the class labels.

   Our class-preserving projections and class-eigenvector plots confine visualizations to an informative $(k-1)$-dimensional subspace of the data. Since these plots are relatively cheap to compute, an important question is: what is the quality of data approximation achieved by our projection schemes? It turns out that class-eigenvectors plots are qualitatively nearly as good as PCA; Figure 10 in Section 2.2.2 gives an example. Thus we are able to obtain qualitatively good projections but at a much reduced cost.

In spirit, our projection schemes are most closely related to the projections of classical linear discriminant analysis [1], see the end of Section 2.1 for a comparison. However, linear discriminant analysis requires the solution of a large generalized eigenvalue problem and comparatively, our methods are much more computationally efficient. Our visualization methodology of showing both local and global views is closest to the work of Gnandesikan, Kettenring, and Landwehr [4]. These authors considered two types of projections: (a) local projections that focus on one class and show its relationship to nearby classes, and (b) a global projection onto the two leading principal components of the between-class scatter matrix, S_B. The latter projection is very similar to one of our class-eigenvector plots, see Section 2.2.1 for details. However our work differs from [4] in many ways. We have considered a much wider variety of projections, each of which gives a local view of a user-specified subset of the classes. Using class tours (see below), we also attempt to view projections onto the entire column subspace of S_B, and not just onto the two leading principal components.

2. **Class-similarity graphs** enhance each individual two-dimensional projection of the data. These graphs provide a skeleton of the data and serve as guides through the various projections reminding the user of similarities in the original multidimensional space.

   Our class-similarity graphs have been inspired by the data-similarity graphs of Duda & Hart [1]. In data-similarity graphs, each data point corresponds to a vertex. In contrast, in class-similarity graphs it is each centroid that corresponds to a vertex. When faced with more than 10,000 points (as in our PENDIGITS data), data-similarity graphs are too detailed and incomprehensible. On the other hand, class-similarity graphs give a high-level and easily understandable view of class relationships.

3. **Class tours** show sequences of two-dimensional class-preserving projections to create the illusion of smooth motion through a multidimensional display. Class tours allow us to "view" higher-dimensional subspaces.

   Other types of *tours* have been considered earlier. Asimov, Buja and colleagues first proposed grand tours in [21, 22, 23]. A grand tour displays a carefully designed sequence of two-dimensional projections that covers the entire high-dimensional space. However, grand tours are independent of the underlying data distribution and when the dimensionality of the original data is high, these tours are computationally prohibitive and can severely test the user's patience as the informative views are few and far between. For even modest dimensionality, grand tours are practically impossible to view in a reasonable amount of time. For example consider the PENDIGITS data set (with $d = 16$). To come within 10 degrees of any plane in such a feature space would require the user to view 0.3×10^{23} randomly sampled planes (see [21]). The interesting information, for example, the class structure is spread out over a large number of projections and hence, is hard to cognitively piece together. To alleviate the computational problems with grand tours, data-dependent guided tours which show sequences of projections using principal components, projection pursuit, etc. were proposed in [14, 15, 16]. The interactive dynamic data visualization software, XGobi [24] contains these guided tours.

In contrast to grand tours, our class tours are data-driven and focus on a small number of judiciously chosen two-dimensional planes all geared towards uncovering the class structure. The class tours we propose are also much more efficient than both the grand tours and the PCA-guided tours. In class tours, we confine our projections to a $(k-1)$-dimensional subspace whereas grand tours show projections in the entire d-dimensional space. As noted earlier the difference between k and d can be quite dramatic, for example, in the ISOLET data, $d = 617$ while k is only 26. In the important application area of text visualization which we have explored in related research, d ranges in the thousands while k is about 10-50. PCA-guided tours suffer a similar cognitive problem as grand tours in uncovering class structure.

Based on the ideas in this paper, we have implemented a software tool named CViz. The CViz software is written in the platform-independent JAVA language, and is currently available as free test software from IBM's Alphaworks site, www.alphaworks.ibm.com/tech/cviz[1]. In fact, most of the figures in this paper are screen shots of plots produced by CViz.

An earlier, shorter version of this paper, where the focus was on visualizing high-dimensional data arising from text documents, was presented at the 1998 Interface Conference [25]. We have also successfully used our projection scheme in an application other than visualization; that of constructing compact representations of large, sparse text data that arise in text mining and information retrieval [26].

We now briefly sketch the outline of the paper. Section 2 introduces class-preserving projections and class-eigenvector plots, and contains several illustrations of the Iris plant and ISOLET speech recognition data sets [27]. Class-similarity graphs and class tours are discussed in Sections 3 and 4. We illustrate the value of the above visualization tools in Section 5, where we present a detailed study of the PENDIGITS on-line handwriting recognition data set [27]. This visual examination allows us to uncover some surprising class relationships.

2 Class-Preserving Projections

Our main tool for visualizing multidimensional data will be linear projections onto 2-dimensional planes which can then be displayed on a computer screen. The key difficulty is the inevitable loss of information when projecting data from high dimensions to just 2 dimensions. This loss can be mitigated by carefully choosing the 2-dimensional planes of projection. Thus, the challenge is in the choice of these planes. *We want to choose those planes (projections) that best preserve inter-class distances.*

2.1 Discriminating Three Classes

We first consider the canonical case where the data is divided into three classes. Let $x_1, x_2, \ldots x_n$ denote the d-dimensional data points divided into the three classes $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$. The corresponding class-means (or class-centroids) are defined as

$$m_j = \frac{1}{n_j} \sum_{x_i \in \mathcal{C}_j} x_i, \qquad j = 1, 2, 3,$$

[1]there have been over 9,166 downloads of the CViz software since it was first released in June, 1998

where n_j is the number of data points in $\mathcal{C}_j$.

For the purpose of visualization, we want to linearly project each x_i onto a 2-dimensional plane. Let $w_1, w_2 \in R^d$ be an orthonormal basis of the candidate plane of projection. The point x_i is projected to the pair $(w_1^T x_i, w_2^T x_i)$ and consequently, the means m_j get mapped to

$$(w_1^T m_j, w_2^T m_j), \quad j = 1, 2, 3.$$

One way to maintain good separation of the projected classes is to maximize the distance between the projected means. This may be achieved by choosing vectors $w_1, w_2 \in R^d$ such that the objective function

$$Q(w_1, w_2) = \sum_{i=1}^{2} \{| w_i^T(m_2 - m_1) |^2 + | w_i^T(m_3 - m_1) |^2 + | w_i^T(m_3 - m_2) |^2\}$$

is maximized. The above may be rewritten as

$$\begin{aligned} Q(w_1, w_2) &= \sum_{i=1}^{2} \{w_i^T\{(m_2 - m_1)(m_2 - m_1)^T + (m_3 - m_1)(m_3 - m_1)^T \\ &\quad + (m_3 - m_2)(m_3 - m_2)^T\}w_i\} \\ &= w_1^T S_B w_1 + w_2^T S_B w_2 \\ &= \text{trace}(W^T S_B W), \end{aligned}$$

where

$$\begin{aligned} W &= [w_1, w_2], \quad w_1^T w_2 = 0, \quad w_i^T w_i = 1,\ i = 1, 2, \quad \text{and} \\ S_B &= (m_2 - m_1)(m_2 - m_1)^T + (m_3 - m_1)(m_3 - m_1)^T + (m_3 - m_2)(m_3 - m_2)^T. \end{aligned} \tag{1}$$

The positive semi-definite matrix S_B can be interpreted as the *inter-class or between-class scatter matrix.* Note that S_B has rank ≤ 2 since $m_3 - m_2 \in \text{span}\{m_2 - m_1, m_3 - m_1\}$.

It is clear that the search for the maximizing w_1 and w_2 can be restricted to the column (or row) space of S_B. But as we noted above, this space is at most of dimension 2. Thus, in general, *the optimal w_1 and w_2 must form an orthonormal basis spanning the plane determined by the vectors $m_2 - m_1$ and $m_3 - m_1$*. In the degenerate case when S_B is of rank one, that is, when m_1, m_2 and m_3 are collinear (but distinct), w_1 should be in the direction of $m_2 - m_1$ while w_2 can be chosen to be any unit vector orthogonal to w_1.

Geometrically, the plane spanned by the optimal w_1 and w_2 is parallel to the plane containing the three class-means m_1, m_2 and m_3. It should be noted that projection onto this plane *exactly preserves* the distances between the class-means, that is, the distances between the projected means are *exactly equal* to the corresponding distances in the original d-dimensional space. Thus, in an average sense, we can say that inter-class distances are preserved, and we call such a projection a *class-preserving projection.*

We illustrate such a class-preserving projection on the famous Iris plant data set in Figure 1. This data set is 4-dimensional and contains 50 members in each of the three classes of plants: *Iris setosa*, *Iris versicolor* and *Iris virginica* (the 4 dimensions are sepal length, sepal width, petal length and petal width). Note that the three classes are well separated in this figure. For comparison, in Figure 2 we have shown the projection onto a poorly chosen plane where the

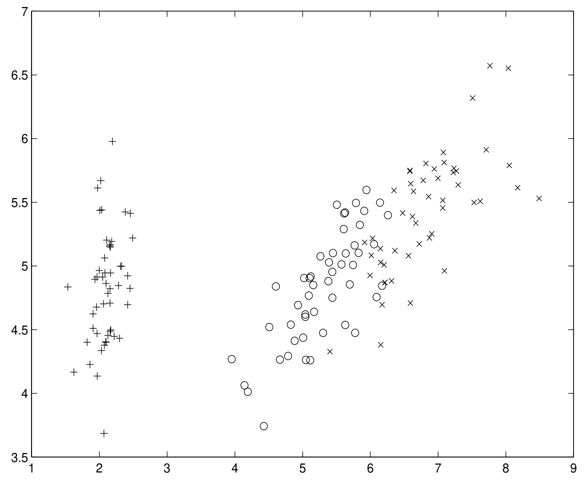


Figure 1: A *Class-Preserving Projection* of the Iris data

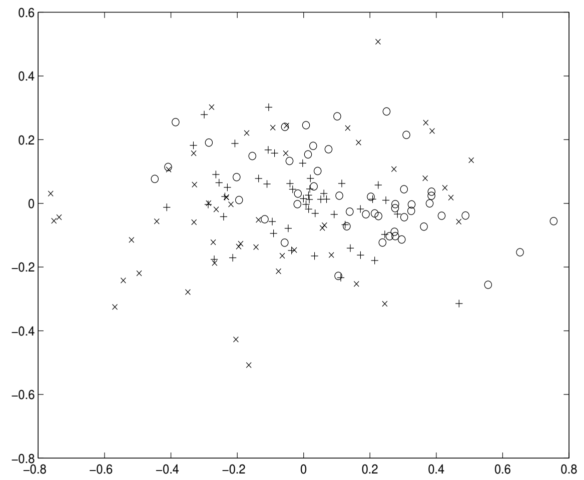


Figure 2: A poor projection of the Iris data

distinction between the classes is lost. Figure 1 allows us to infer that the Iris setosa plants (on the left part of the figure) are easily distinguished by the feature set from the other two Iris varieties, while the latter two are harder to distinguish. Furthermore, we see that the Iris setosa class is linearly separable from the other classes (since linear separability in any two-dimensional projection implies linear separability in the entire 4-dimensional space).

Projection schemes similar to ours have previously been used in classical linear discriminant analysis. In particular, our class-preserving projections are closely related to Fisher's linear discriminant and its generalizations, which maximize the ratio

$$\frac{\mathrm{trace}(W^T S_B W)}{\mathrm{trace}(W^T S_W W)}, \tag{2}$$

where S_B and W are as in (1), and S_W is the *within-class scatter matrix*, see [1]. Finding W that maximizes (2) requires the solution of a generalized eigenvalue problem, $S_B x = \lambda S_W x$, which can be computationally demanding for high-dimensional and sparse data sets. We have chosen to ignore the *within-class scatter* S_W for two reasons: (a) we need to be able to compute projections on the fly for large and high-dimensional data sets, (b) if some scaling of variables is indeed desired we can do so *a priori*, for example, by applying the whitening transform that scales different variables depending on the total scatter [1]. For more details on linear discriminant analysis, the reader is referred to [28, 1, 29, 30, 31]. Earlier in this section, we observed that our class-preserving projections preserve distances between the three class-means, that is, the *multidimensional scaling error* for these class-means is zero. For the more general problem of preserving inter-point distances between all the projected data points, see [12, 13, 1] and [29, Chapter 14].

2.2 Discriminating More than Three Classes

In the above discussion, we arrived at a certain two-dimensional projection that "best" preserves the relationship between three classes. But in most practical situations, we will encounter a greater number of classes. The *Yahoo!* hierarchy, for instance, segments web pages into thousands of categories. What projections will enable us to examine the interplay between these multitude of classes?

Clearly, one option is to examine the k classes three at a time by taking projections identical to those described in the previous section. There are a total of $\binom{k}{3}$ such 2-dimensional projections, each determined by three class-means. Each of these projections gives us a local view of three classes. We illustrate such local views on an interesting speech recognition data set, that we first describe.

The ISOLET (Isolated Letter Speech Recognition) data consists of speech samples of the 26 English alphabets [5], and is publicly available from the UCI Machine Learning Repository [27]. The data was collected from 150 subjects who spoke the "name" of each letter twice. After some preprocessing, 617 real-valued attributes were chosen to describe each spoken letter. These features consist of various spectral coefficients (sonorant, pre-sonorant, post-sonorant, etc.) of the sampled data, plus other measurements that capture pitch and amplitude characteristics. For a complete list of the feature set and a detailed description, the reader is referred to [5]. Each of the 617 features is mapped to the interval $[0.0, 1.0]$ and is normalized to utilize this entire range. The entire data set comprises 7,797 samples.

For clarity of presentation, we will only consider samples of the seven spoken letters A through G. There are a total of $2,098$ such samples. Note that this data set is extremely high-dimensional ($d = 617$) and occupies nearly 5 MBytes of memory. In their paper [5], Fanty and Cole designed a neural network classifier for spoken letter recognition. Their classifier used domain knowledge to achieve high accuracy, for example, they trained a separate classifier to distinguish between the letters in the E-set, namely B, C, D and E. We now see how our class-preserving projections enable us to visually uncover such domain knowledge, and capture various *inter-letter* and *intra-letter* relationships in this multidimensional speech data set.

In Figure 3, we show the class-preserving projection determined by the centroids (or means) of the spoken letters B, C and F. We depict each data point by the corresponding lower-case letter, while the centroid is denoted by the upper-case letter. In Figure 3, the upper-case letters enclosed by a box, that is, $\boxed{\mathsf{B}}$, $\boxed{\mathsf{C}}$ and $\boxed{\mathsf{F}}$, denote the centroids that determine the class-preserving projection. We will use this convention throughout the remaining plots in this paper. In our ensuing discussion, we will "overload" the upper-case letters A,B,...,G to denote various quantities. Thus the letter A may indicate (i) the spoken letter A, (ii) the

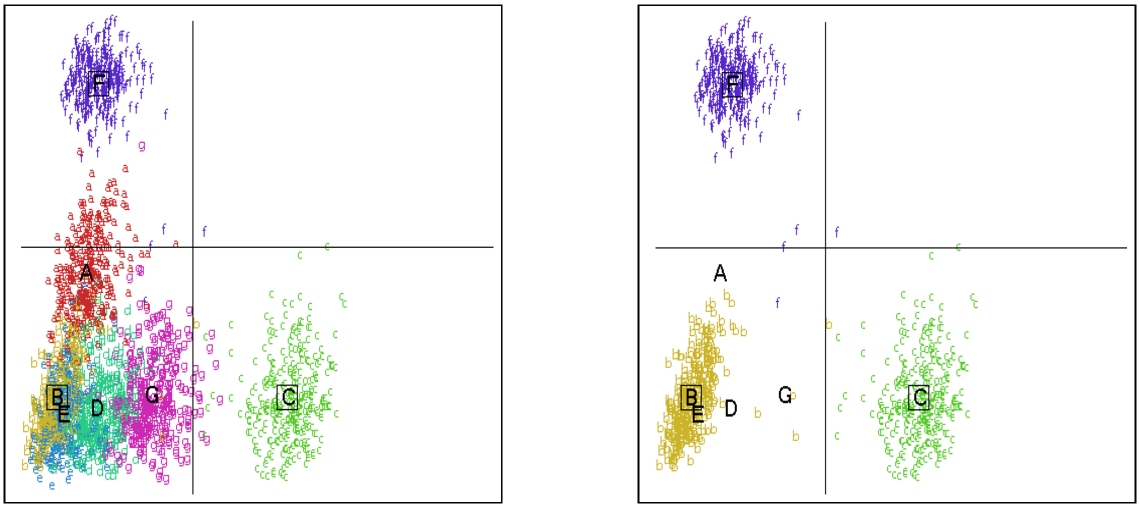


Figure 3: Class-preserving projection to discriminate B, C and F in the ISOLET data. Both left and right panels show the same projection; the left panel shows samples of all 7 letters, while the right panel does not show individual samples of A, D, E and G.

class containing all the data samples of the spoken letter A, or (iii) the corresponding class-centroid. The particular usage should be clear from the context. The origin along with a pair of orthogonal axes is also shown in all our figures. Note that the (x,y) coordinates of each point are dimension-preserving, i.e., they preserve the scale of each data dimension as provided by the user.

Figure 3 shows that the classes B, C and F are well-separated and almost equally far apart. The D and E classes seem to be quite close to B, while most A samples lie between B and F. The left panel of Figure 3 displays all the $2,098$ data points and hence is rather crowded. Note that the A, D, E and G samples do not influence the choice of the plane of projection shown. We can hide such secondary samples in order to get a clearer view of the discriminated classes. The right panel of Figure 3 shows the same projection as the left panel except that A, D, E and G data samples are not displayed. Only centroids of these classes are shown.

Figure 3 shows a projection expressly chosen to preserve only the distances between the B, C and F centroids. Distances from the other centroids are not preserved by this projection. Thus proximity of the D and E centroids to the B centroid in this figure may be misleading, that is, although D and E appear close to B in this 2-dimensional projection they may be quite far in the original 617-dimensional space. To check if B and D are close, we look at the class-preserving projection that preserves distances between B, D and F in Figure 4. Here we see that D is indeed quite close to B, and they are both well separated from F (as validation, note that the spoken letters B, D and E can be hard to distinguish). The right plot in Figure 4 shows a similar relationship between B, E and F.

In all the above figures we see that A lies between B and F. We may suspect that A is closer to F than these figures indicate (note that none of the above figures preserve distances to A). However, in Figure 5, the class-preserving projection determined by A, B and F shows that A and F are not particularly close. In fact, the A and F classes are seen to be linearly separable. In Figure 5, if we project all the data samples onto the X-axis, we see that F samples are closer to A than to B. Thus, it can be deduced that the F samples share some correlated features with

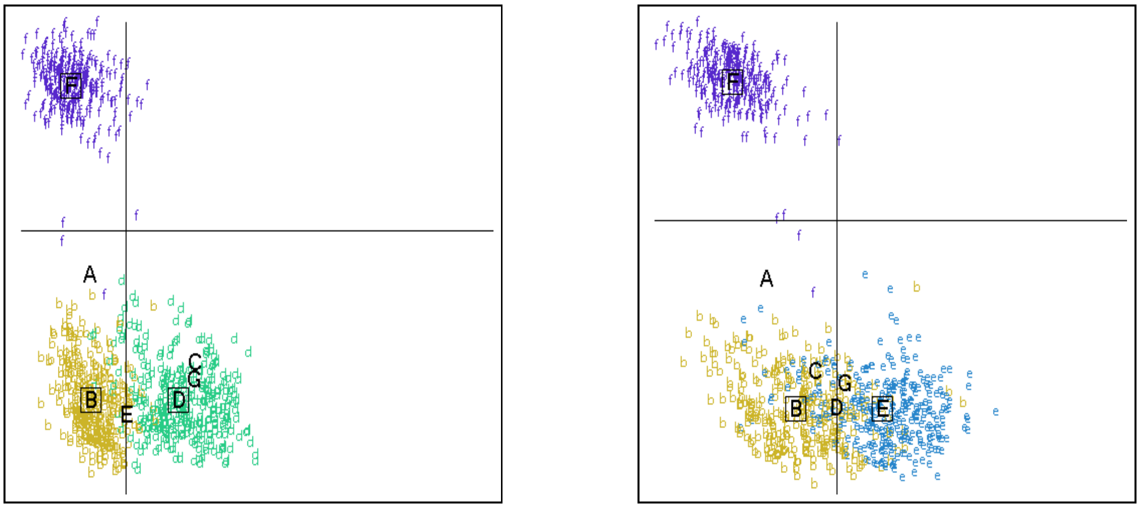


Figure 4: Projections to discriminate B, D, F and B, E, F in the ISOLET data

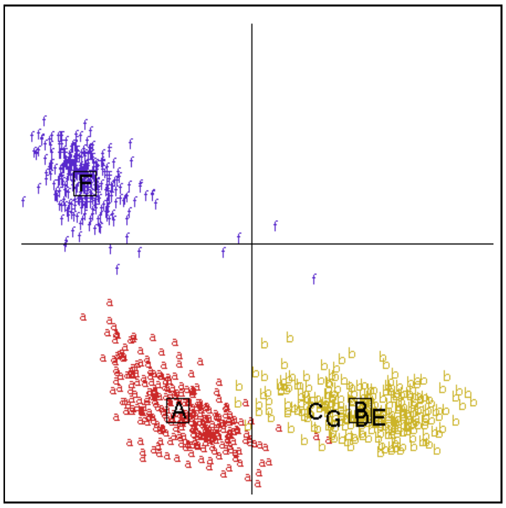


Figure 5: Projection to discriminate A, B and F in the ISOLET data

A that distinguish them from B.

2.2.1 Class-Eigenvector Plots

The ISOLET data that we have considered contains seven classes. However, thus far we only have a mechanism to display the inter-class relationships between three classes at a time. In general, *we would like to differentiate between more than three classes in the same view.*

More formally, we want to obtain a 2-dimensional projection that best discriminates the q classes with class-means $m_1, m_2, \ldots m_q$, each containing $n_1, n_2, \ldots n_q$ data points, respectively. Taking an approach similar to that of Section 2.1, we can formulate the above objective as the search for orthonormal $w_1, w_2 \in R^d$ that maximizes

$$Q(w_1, w_2) \quad = \quad \text{trace}(W^T S_B W), \tag{3}$$

where

$$\begin{aligned} W &= [w_1, w_2], \quad w_1^T w_2 = 0, \quad w_i^T w_i = 1,\ i = 1, 2, \quad \text{and} \\ S_B &= \sum_{i=2}^{q} \sum_{j=1}^{i-1} n_i n_j (m_i - m_j)(m_i - m_j)^T. \end{aligned} \tag{4}$$

Note that the positive semi-definite matrix S_B has rank $\leq q-1$ since the vectors $m_i - m_j$, $j \neq 1$ are linearly dependent on the $q-1$ vectors $m_i - m_1$, $i = 2, \ldots, q$. It is well known that *the vectors w_1 and w_2 that maximize the objective function in (3) are the eigenvectors (or principal components) corresponding to the two largest eigenvalues of S_B.* The reader should note that for $q > 3$, in general, there is no two-dimensional plane that *exactly* preserves the distances between the q centroids $m_1, m_2, \ldots, m_q$. The plane spanned by the optimal w_1, w_2 preserves inter-class distances to the largest extent possible, where the error due to projection is measured in the 2-norm or Frobenius norm, or any unitarily invariant norm [32, 29].

The reader might have noticed the extra factor $n_i n_j$ in (4) that was not present in (1). By weighting each $(m_i - m_j)(m_i - m_j)^T$ term in (4) by the factor $n_i n_j$, we are placing greater emphasis on preserving distances between the class-means of larger classes.

The matrix S_B as given above in (4) is the sum of $\binom{q}{2}$ rank-one matrices. We can show that S_B can be expressed more compactly as the sum of q rank-one matrices. In particular,

$$S_B \quad = \quad n^{(q)} \sum_{i=1}^{q} n_i (m_i - m^{(q)})(m_i - m^{(q)})^T, \tag{5}$$

where

$$\begin{aligned} n^{(q)} &= n_1 + n_2 + \cdots + n_q, \quad \text{and} \\ m^{(q)} &= \frac{1}{n^{(q)}} (n_1 m_1 + n_2 m_2 + \cdots n_q m_q) \end{aligned}$$

is the mean of all the data points in the q classes under consideration. This alternate expression for S_B is more compact and better for computational purposes. The interested reader can look at the Appendix for a proof of the equivalence of formulae (4) and (5).

In Figure 6, we show a 2-dimensional projection (obtained using the 2 largest eigenvectors of the corresponding S_B in (5)) that preserves inter-class distances between four of the seven

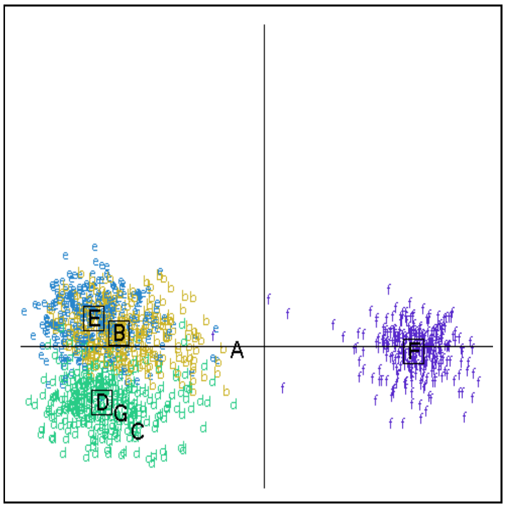


Figure 6: Projection to discriminate B, D, E and F in the ISOLET data

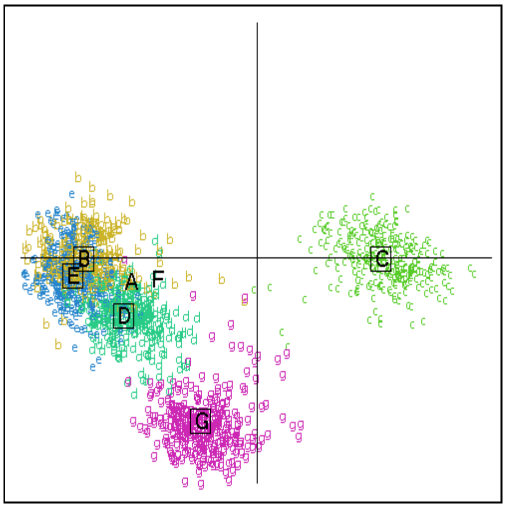


Figure 7: Projection to discriminate B, C, D, E and G in the ISOLET data

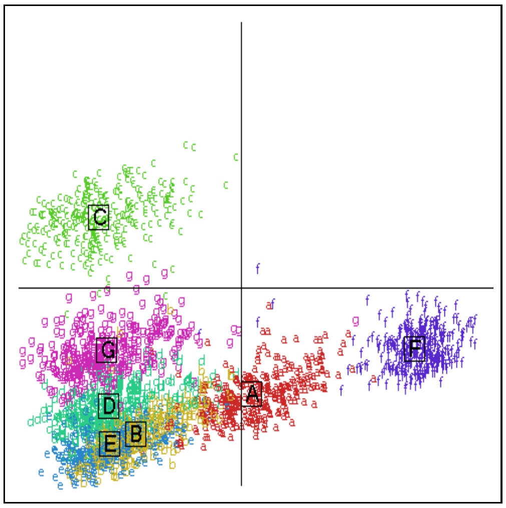


Figure 8: Projection to discriminate all the 7 letters in the ISOLET data

classes — B, D, E and F. This view confirms our earlier observations that B, D and E are close to each other and quite distant from F. Figure 7 attempts to differentiate between the B, C, D, E and G classes. Here we see that G is closer to B, D and E while C is more distant. Observe that A and F appear close to each other and to B, D and E in Figure 7, but note that this projection does not preserve distances to the A and F classes. Recall Figure 5 which showed that A and F are not so close. Figure 8 has $q = k = 7$, and gives a more accurate estimate of the relationships between all the classes.

Finally, in Figure 9, we consider all 7,797 data samples of the 26 English alphabets that comprise the complete ISOLET data. This figure shows a projection that tries to discriminate between all 26 letters. The letters B, C, D, E, G, P, T, V and Z, which constitute the so-called E-set are seen to be very close in this projection. The reader should note that the above observations seem "correct" and "natural" in the context of this intuitive speech data. In cases where there is not much domain knowledge about the data, such visual discoveries could be even more valuable.

Thus, given k classes we have presented a mechanism for viewing the inter-class relations between any q of them. There are a total of

$$\sum_{q=3}^{k} \binom{k}{q} = 2^k - \frac{k(k+1)}{2} - 1$$

such informative class projections. However, as q gets larger, the distinction between classes starts getting blurred in these views. Figure 9 gives a crowded view that attempts to discriminate all 26 classes. Such crowding is inevitable since our linear projections are limited to 2-dimensional planes. Ideally, we would like to "view" projections onto higher-dimensional subspaces.

2.2.2 Projections to Higher-Dimensional Subspaces

Suppose for a moment that we can visualize class-eigenvector projections of the data onto a p-dimensional subspace, $p \geq 3$. As before, we want to preserve distances between all the centroids

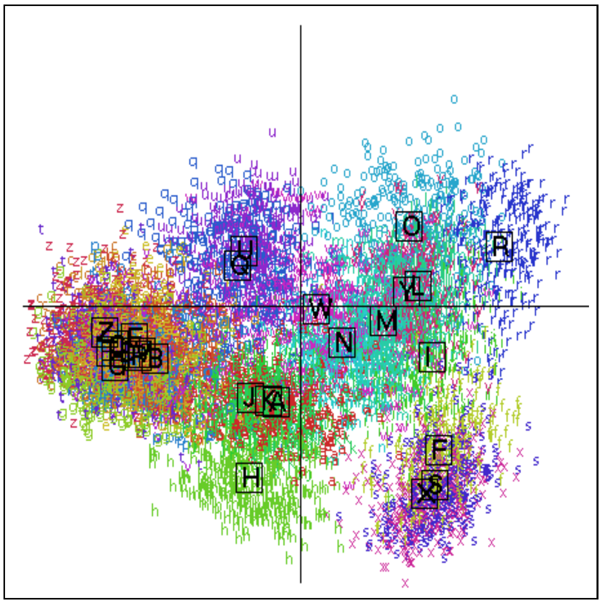


Figure 9: Projection to discriminate all 26 letters in the entire ISOLET data

after projection. Thus, given k classes, we want to find orthonormal $w_1, w_2, \ldots, w_p \in R^d$ such that the objective function

$$Q(w_1, \ldots, w_p) \;=\; \mathrm{trace}(W^T S_B W),$$

is maximized, where

$$W = [w_1, \ldots, w_p], \quad w_i^T w_j = 0, \quad w_i^T w_i = 1, \quad 1 \le i, j \le p, \quad i \ne j,$$

and S_B is as in (4) or (5) with q replaced by k.

The optimal w_i are given by the p eigenvectors (principal components) of S_B corresponding to its p largest eigenvalues. When $p = k - 1$, the desired subspace is the entire column (or row) subspace of S_B, and $w_1, w_2, \ldots, w_p$ can be any orthonormal basis of this subspace. In this case, distances between all the class-means are *exactly preserved*. Since we are considering all k classes we say that these projections capture a global view of the data.

It is interesting to compare the quality of such class-eigenvector projections with the traditional projections provided by PCA (principal components analysis) of the entire data set. The quality of various projection schemes can be measured by the so-called "energy lost" due to the projections, which we call the approximation error. In precise terms, let $X = [x_1 - m, x_2 - m, \ldots, x_n - m]$, where x_i is the i-th data point and m is the mean of the entire data set. Then the approximation error due to projection on to the orthonormal vectors $w_1, w_2, \ldots, w_p$ is given by the matrix

$$E \;=\; X - \sum_{i=1}^{p} w_i w_i^T X.$$

The squared Frobenius norm of E is simply the sum of squares of all entries in E and provides a way to measure the size of the error. Note that setting the w_i in the above formula to the p leading principal components of X gives the approximation error in PCA.

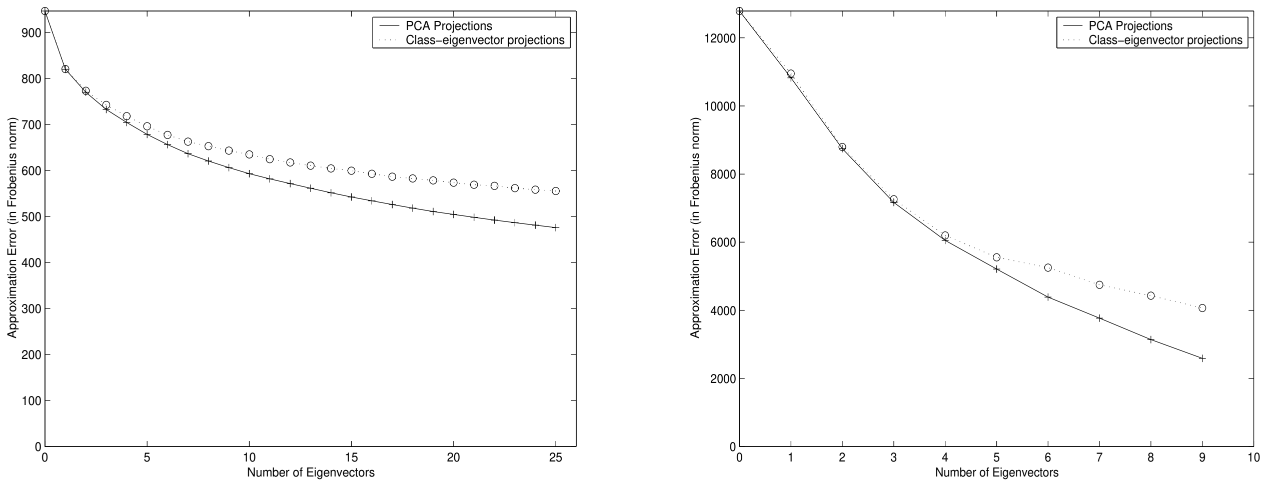


Figure 10: Plots showing that the approximation errors in our class-preserving projections closely track those in the optimal PCA projections. The left plot shows the error for the entire ISOLET data set ($n = 7797$, $d = 617$, $k = 26$) while the right plot shows the error for the PENDIGITS data set ($n = 10992$, $d = 16$, $k = 10$). Details of the PENDIGITS data set are given later in Section 5.2.

Figure 10 shows that the approximation errors due to class-eigenvector projections are comparable to those due to PCA. Note that PCA is provably optimal in minimizing the approximation error [29, Section 8.2.3], but involves computations on the entire $d \times n$ data matrix. Thus our class-eigenvector projections offer a computationally superior method without compromising on the quality of the projections. We have also observed this behavior in another application — dimensionality reduction of high-dimensional and sparse document vectors, see [26] for details.

Of course, it is not directly possible to visualize a p-dimensional subspace for $p > 3$. We are limited to 2-dimensional computer displays. In Section 4, we propose a way to explore the entire column space of S_B. But before we do so, we look at a tool that enhances each individual projection.

3 Class-Similarity Graphs

Two-dimensional linear projections are a continuous transformation of the data; two points which are close in R^d will remain close in each of these projections. However, two points which are close in a 2-dimensional projection need not be close in the original space R^d. To mitigate this information loss, we add *class-similarity graphs* to our 2-dimensional plots.

We define a class-similarity graph as follows. The vertices of this graph are the class centroids $m_1, m_2, \ldots, m_k$, and there is an edge between the means (or vertices) m_i and m_j if

$$d_2(m_i, m_j) \leq \tau, \tag{6}$$

where d_2 denotes the Euclidean distance, while τ is a user-controlled threshold parameter. If τ is very large, then all centroids will be connected. On the other hand, if τ is very small, then no centroids will be connected. It is thus intuitively clear that the choice of this threshold

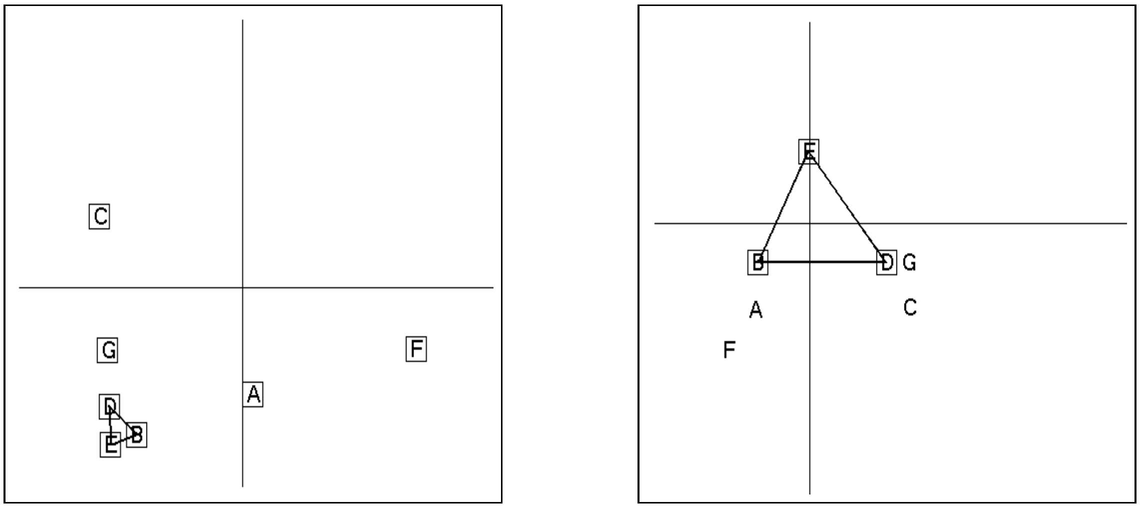


Figure 11: Class-similarity graph overlaid on two projections in the ISOLET data: B, D and E are seen to form a clique at the chosen threshold τ

parameter is important in revealing similarities (or dissimilarities) between the class-means. To obtain "natural" connections between the centroids, τ will have to be greater than typical distances between related classes but less than typical distances between unrelated ones.

We can display class-similarity graphs by overlaying them on our 2-dimensional class-preserving projections. The left plot in Figure 11 shows one such graph on the projection that discriminates between all the seven classes (this projection is identical to the one in Figure 8). Note that to show the similarity graph clearly in this figure, we have shown only the seven centroids and removed the individual data points. The B, D and E classes are seen to form a clique, which is consistent with our observations about their closeness in Figures 6, 7 and 8. The class-similarity graph provides a skeleton of the data and reminds us of the proximity relationships between classes through various views of the data. For example, the right plot in Figure 11 shows the same similarity graph as the left plot, but overlaid on the projection that discriminates between B, D and E. This graph reminds us that B, D and E are the closest among the seven letters even though they appear far apart in this view. Thus the class similarity graph adds another valuable *information dimension* to linear projections. Finally, in Figure 12 we display the similarity graph at a higher threshold τ (see (6) above), and observe that B, D, E and G form a clique indicating the nearness of these letters.

In the context of speech recognition systems, the closeness of B, D, E and G to each other indicates that these letters may be the most difficult to classify and hence "recognize". Indeed, in [5, p.223] the authors remark that they "trained separate (neural) networks for just the letters in the E-set" (see also [33]).

Our class-similarity graphs are extensions of the data-similarity graphs between *all* the data points given in [1, p.567]. However, in data-similarity graphs each data point corresponds to a vertex, whereas in class-similarity graphs it is each centroid that corresponds to a vertex. Consider the projection of Figure 9 which displays 7,797 data points. Clearly a data-similarity graph in this case would be too detailed and incomprehensible. On the other hand, Figure 11 shows that class-similarity graphs give a high-level and easily interpretable view of a data set's class structure.

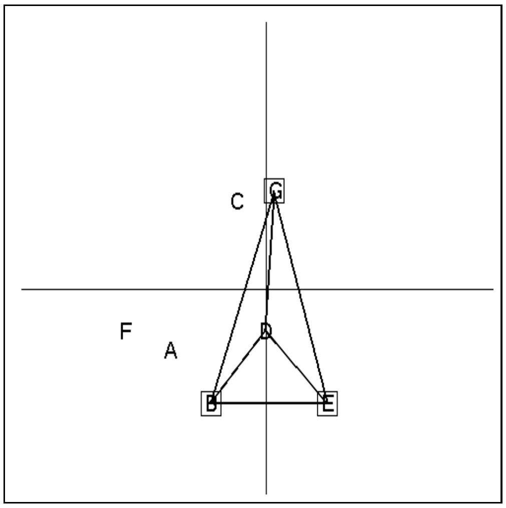


Figure 12: B, D, E and G form a clique at a higher threshold τ

4 Class Tours

Thus far we have limited ourselves to static 2-dimensional snapshots of the data set, each of which conveys some limited information. Projection onto the entire $(k-1)$-dimensional subspace spanned by the vectors $m_i - m_1$, $i = 2, \ldots, k$, contains more global inter-class information since it *exactly* preserves the distances between all the k class-means. Ideally, we would like a mechanism for viewing this *class-preserving* linear subspace.

In order to simulate multidimensional displays, Asimov proposed the use of motion graphics in [21]. Specifically, he introduced the concept of *tours* which are sequences of 2-dimensional projections interspersed with a number of intermediate projections. These intermediate projections are obtained by interpolation, and thus tours create an illusion of continuous smooth motion through a multidimensional display. The *grand tours* proposed in [21, 22] try to display a carefully designed sequence of 2-dimensional projections that are dense in the set of all such projections. However, such sequences are independent of the data set to be visualized, require substantial computation and can severely test the user's patience when the data is high-dimensional. Guided tours proposed in [14] alleviate this problem by choosing sequences tailored to the underlying data; these tours may be guided by principal components, canonical correlations, data sphering or projection pursuit. See [14, 15, 16] for details.

We have found our class-preserving projections to give good local data displays, for example, see the Figures in Section 2. To get a more global view of the data, we propose *class tours* which are sequences of class-preserving 2-dimensional projections, and are an effective tool to "view" the $(k-1)$-dimensional class-preserving subspace. Metaphorically speaking, a class tour constructs a dynamic, global "movie" of this subspace from a number of static, local snapshots.

The basic idea behind class tours is simple: choose a target 2-dimensional projection from the subset of nearly 2^k class-preserving projections and class-eigenvector plots, move smoothly from the current projection to this target, and continue. The main questions of interest are (a) the choice of the intermediate 2-dimensional projections, and (b) the choice of the orthonormal basis used for viewing each projection so that the motion appears smooth. For this purpose, the use of geodesic interpolation paths between the current and the target planes has been proposed in [21, 22]. Each geodesic path is simply a rotation in the (at most) 4-dimensional linear subspace containing both the current and the target 2-d planes. Various smoothness properties of such geodesic paths are explored in great detail in [23].

For the sake of completeness, we now describe how to construct a geodesic path between a *current* plane U and a *target* plane V.

1. Compute the so-called principal vectors and associated principal angles, which have the following important properties.

   (a) The first pair of principal vectors, $u_0 \in U$ and $v_0 \in V$, makes the smallest possible angle θ_0 among all pairs of vectors, one drawn from U and the other from V, i.e., $\{u_0, v_0\} = \mathrm{argmax}_{u \in U, v \in V} \cos \angle(u, v)$. The angle $\theta_0 = \angle(u_0, v_0)$ is called the largest principal angle.

   (b) The second pair of principal vectors, $u_1 \in U$ and $v_1 \in V$, makes the smallest possible angle in the orthogonal complement of u_1 in U and v_1 in V. The corresponding angle θ_1 is called the second principal angle.

   The pairs (u_0, u_1) and (v_0, v_1) give orthonormal bases for U and V respectively. Computationally, these principal angles and principal vectors can be obtained from the singular value decomposition of the 2×2 matrix $Q_U^T Q_V$, where the columns of Q_U and Q_V comprise (arbitrary) orthonormal bases of U and V respectively. More details can be found in [34] and [32, Section 12.4.3].

2. Orthogonalize v_0 against u_0, and normalize to obtain the unit vector $u_0^\perp$. Similarly obtain $u_1^\perp$ by orthogonalizing v_1 against u_1.

3. Let the intermediate projection planes between U and V have basis vectors $(x_0(t), x_1(t))$, which are given by

   $$x_0(t) = \cos(t\bar{\theta}_0)u_0 + \sin(t\bar{\theta}_0)u_0^\perp, \qquad x_1(t) = \cos(t\bar{\theta}_1)u_1 + \sin(t\bar{\theta}_1)u_1^\perp,$$

   where $\bar{\theta}_0 = \theta_0/\sqrt{\theta_0^2 + \theta_1^2}$ and $\bar{\theta}_1 = \theta_1/\sqrt{\theta_0^2 + \theta_1^2}$. The parameter t is varied from 0 to $\sqrt{\theta_0^2 + \theta_1^2}$, and the planes spanned by the bases $(x_0(t), x_1(t))$ give a geodesic from U to V.

Note that the above procedure specifies a geodesic between a current plane and a target plane. However, such a path may be embedded in a long sequence of 2-dimensional planes. In such a case, we must use properly rotated principal vectors for the horizontal and vertical axes. This avoids subjecting the user to meaningless within-screen rotations whenever a new geodesic path is to be resumed. Thus the above procedure results in interpolating planes rather than specific pairs of basis vectors. More computational details may be found in [14, section 2.2.1].

Although we have presented the main ideas above, it is hard to illustrate class tours through static snapshots. We encourage the interested reader to experiment with our CViz software, which is currently available at www.alphaWorks.ibm.com/formula/cviz.

5 A Case Study

In this section, we present a case study that illustrates how we can use the techniques developed above to explore a real-life multidimensional handwriting recognition data set. As we shall see, our visual exploration allows us to uncover some surprising characteristics of the data set. Although we cannot hope to fully convey the discovery process, we present snapshots of the interesting findings to illustrate how we can incrementally build upon our discoveries.

5.1 CViz Software

Our CViz visualization software is built upon the tools developed in the earlier sections, namely, class-preserving projections, similarity graphs and class tours. The CViz software allows the user to choose from among the nearly 2^k class-preserving projections and class-eigenvector plots, providing a seamless way to move from one projection to another. This enables the user to navigate through various local and global views of the data. To facilitate visual discovery, the CViz software provides the following additional features:

1. A facility for brushing selected points with a separate color, thus making it easier to follow the relationship of the brushed points to the rest of the data through various projections.

2. A "zoom-in" or magnification feature that allows closer examination of a sub-region occupied by the data.

3. At the user's request, a matrix of different local and global projections can be viewed. Each individual projection is miniaturized and the user can then choose to enlarge any of these projections to examine in greater detail. Further ideas on using such tables of projections may be found in [35].

4. A feature to view contour plots of the data thus enabling efficient display and easy recognition of high-density regions in each plot.

5. The user can choose to display various points as glyphs that may make more semantic sense in an application. Note that we have used alphabets as glyphs in the display of the ISOLET data (see Figures 3 through 8), and will use digits as glyphs in plots of the upcoming PENDIGITS data. For more details on the use of glyphs in visualization, see [36].

6. A facility to highlight outliers, which can be defined as data points furthest from the class centroid, in order to detect them and view the spread of a cluster.

7. A dynamic slider to choose the threshold τ in displaying class-similarity graphs (see (6) in Section 3).

5.2 The On-line Handwriting Recognition Data Set

The PENDIGITS data set consists of 250 handwriting samples from 44 writers [37, 6], and is publicly available from the UCI Machine Learning Repository [27]. The authors of [6] collected the raw data from a pressure sensitive tablet that sent x and y coordinates of the pen at fixed time intervals of 100 milliseconds. Two preprocessing steps were performed to reduce meaningless variability arising from variations in writing speed and sizes of the written digits:

Normalization makes the data invariant to translations and scale distortions. Both x and y coordinates of the raw data were scaled so that the coordinate which has the maximum range varied between 0 and 100.

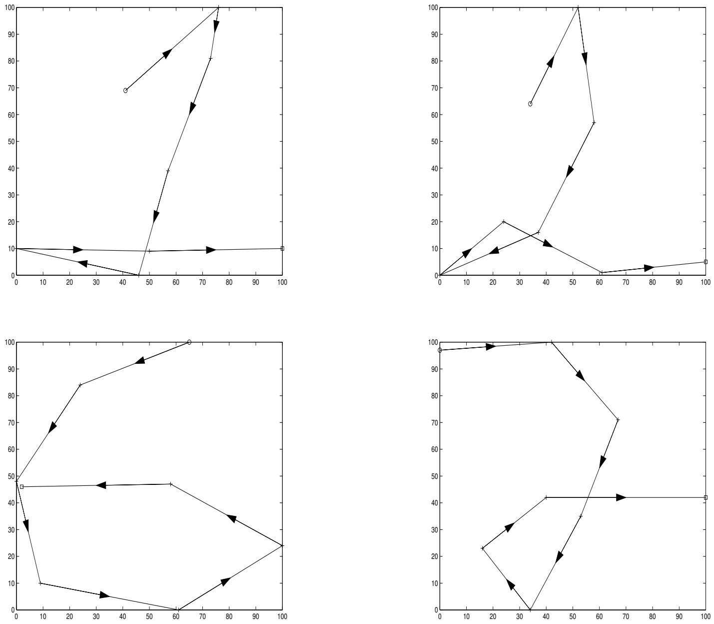


Figure 13: Sample "reconstructed" handwritten digits — 1, 2, 6 and 7

Resampling represents each digit as a constant length feature vector. In this data set, spatial resampling with simple linear interpolation was used to obtain 8 regularly spaced points on the trajectory of each digit.

Thus, each digit is represented by 8 (x, y) coordinates leading to a 16-dimensional feature vector. Each of the 16 attributes is an integer varying from 0 to 100. In Figure 13, we show some of the reconstructed digit samples, where the directions of the arrows indicate the pen's trajectory. The starting point of the pen's path is indicated by a 'o', the end point by a '□', while the 6 intermediate points are marked by '+'. Note that these displayed samples are not replicas of the pen's original trajectory but are *reconstructed* from the 8 (x, y) coordinates by connecting adjacent coordinates by a straight line. Hence, even though the 8 (x, y) coordinates are regularly spaced in arc length of the pen trajectory, they do not appear so in Figure 13. Also note that since the pen pressure values are not stored the "pen lifts" are lost, as in the handwritten 7 in Figure 13.

The entire PENDIGITS data set consists of 10,992 samples. More information on this data set may be obtained from [37, 6, 27].

5.3 Visual Exploration of the PENDIGITS Data

We start our visual examination with Figure 14 which displays the class-preserving projection that preserves distances between the centroids of digits 0, 1 and 2. As in most of the figures of Section 2.2, we only show centroids of the other classes, and not their individual data points. The centroids that determine the class-preserving projection are denoted by the corresponding digit enclosed by a box, for example, $\boxed{0}$, $\boxed{1}$ and $\boxed{2}$ in Figure 14.

We now enumerate our findings in an order in which the discovery process might progress.

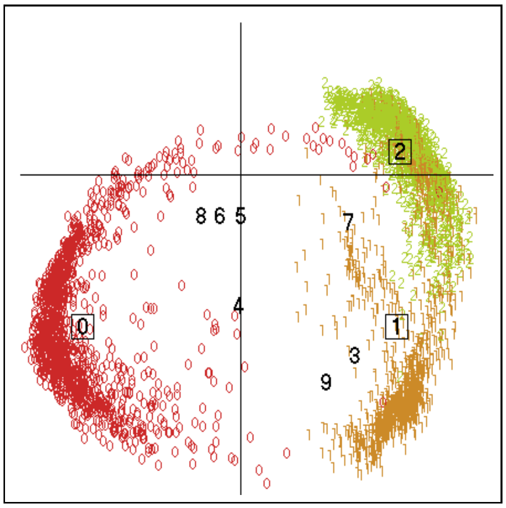


Figure 14: Class-Preserving Projection discriminates between 0, 1 and 2 in the PENDIGITS data

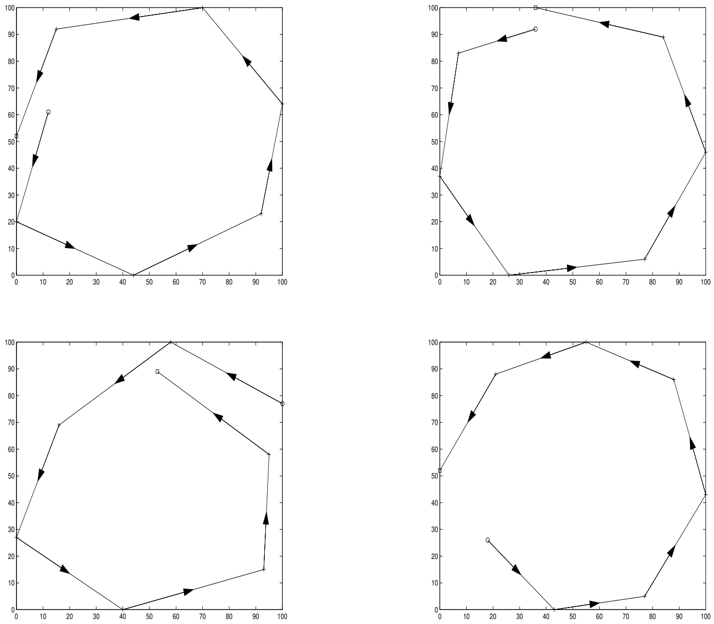
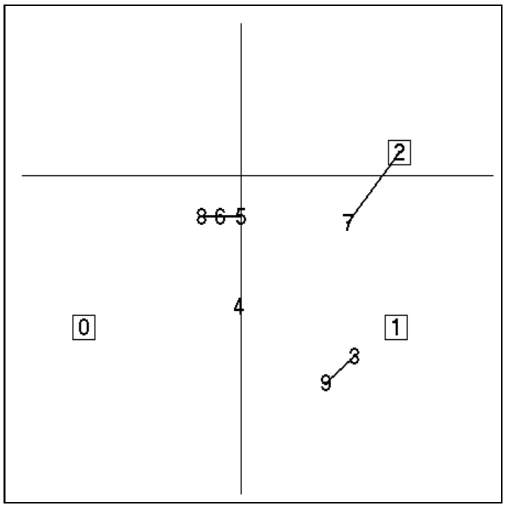


Figure 15: Different handwritten zeros



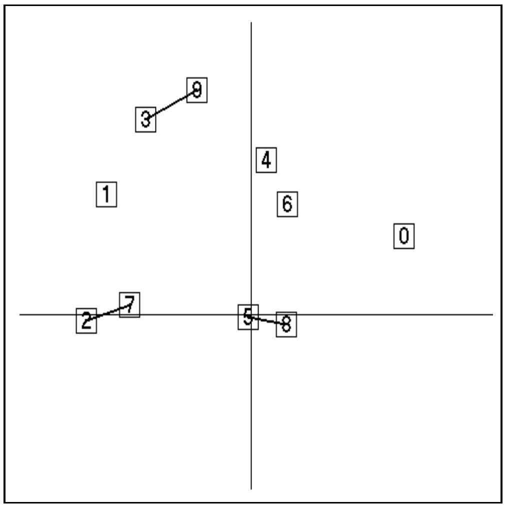


Figure 16: Class similarity graph for the PENDIGITS data overlaid on two projections

1. Figure 14 shows that the digits 1 and 2 are closer to each other than to 0. The 0 class appears to have a large variance, while the cluster of 2's is more coherent. In fact, we see that the 0 class forms a boomerang-like shape. The large variance implies that there are a variety of ways of writing 0, while the boomerang shape suggests a continuum in the different written 0's. We can delve deeper into this hypothesis by looking at different written 0 samples, some of which are shown in Figure 15. Indeed, we see that these different 0's seem to be rotations of each other — each 0 traces an anti-clockwise arc but has different starting and ending points.

   In Figure 14, we also see that centroids for the digits 5, 6 and 8 lie near each other, while centroids of 3 and 9 are also close. But this projection tries to preserve distances only between the 0, 1 and 2 centroids, and the proximity of other digits may be misleading here. To get an estimate of inter-class distances in the original 16-dimensional space, we turn to class-similarity graphs.

2. Class-similarity graphs at an interesting threshold τ are shown in Figure 16. Note that the projection on the left plot of Figure 16 is identical to the one in Figure 14. Since they are connected by edges, centroids for digits 2 & 7, and digits 3 & 9 are close to each other in the original space. The similarity between the written digits 2 and 7 is also clearly seen in Figure 13. The third "edge" in the class-similarity graph seems to connect the three digits 5, 6 and 8. However it is not clear if this triplet forms a clique, so we look at the same class-similarity graph in another projection.

3. The right plot in Figure 16 clearly shows that the centroids for 5 and 8 are close. However, 6 is not particularly close to either 5 or 8. Note that this projection attempts to discriminate between all ten digits and gives a good estimate of their relative distances from each other.

4. We may now wish to examine the pairs 2 & 7, 3 & 9, 5 & 8 more closely. To do so, we look at each pair relative to the 0 class. These three projections are shown in Figures 17 and 18. These views allow us to make the following inferences:

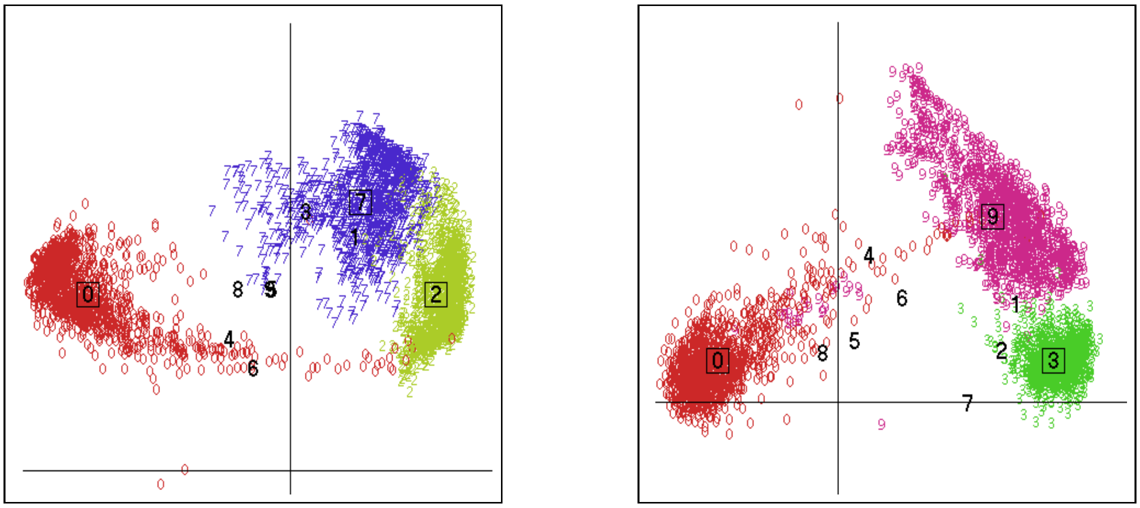


Figure 17: Projections to discriminate (a) 0, 2 and 7 and (b) 0, 3 and 9

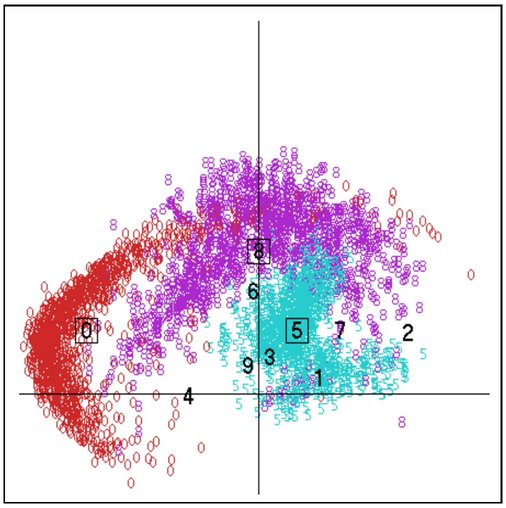


Figure 18: Projection to discriminate 0, 5 and 8

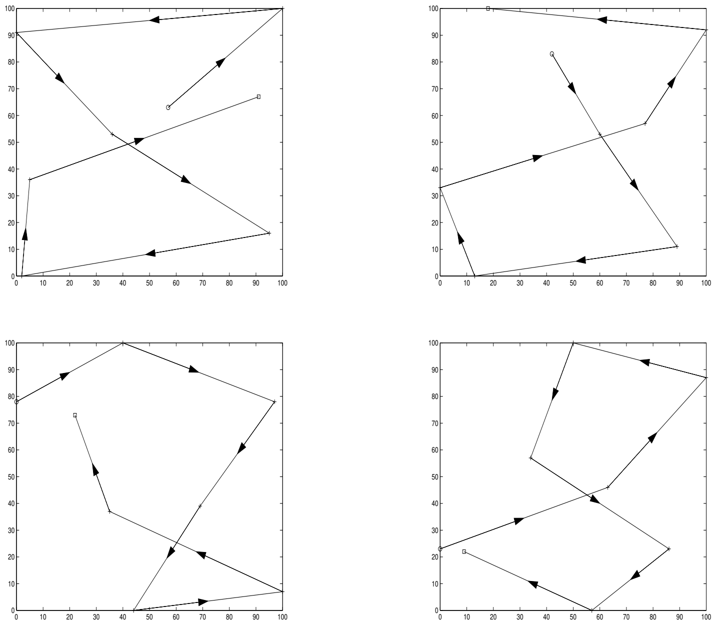


Figure 19: Different types of 8

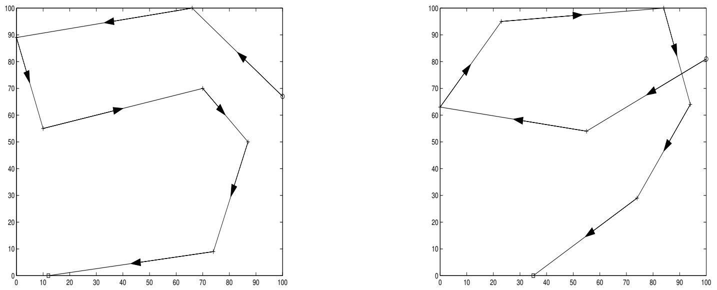


Figure 20: Different types of 9

(a) The two pairs 2 & 7, 3 & 9 appear quite close to each other, and rather distant from 0.

(b) Figure 18 shows that some of the 8's are similar to the 0's. Some other samples, such as 5's and 8's, also appear intermingled. In the context of designing a classifier, these samples could be misclassified by a linear classifier.

(c) The classes 2 and 3 are very coherent and indicate a consistency among the writers in writing these digits.

(d) The classes 8 and 9 are seen to have large variances. This observation is confirmed by the variety of ways of writing 8 and 9 seen in Figures 19 and 20.

5. While surfing through other projections, we found a particularly interesting one that we now present. Figure 21 attempts to maximally preserve the distances between the centroids of digits 3, 5, 8 and 9. As expected, we see that 3 and 9 samples lie close to each other. However, the class-structure of 5 is revealing. *The* 5 *samples appear in two*

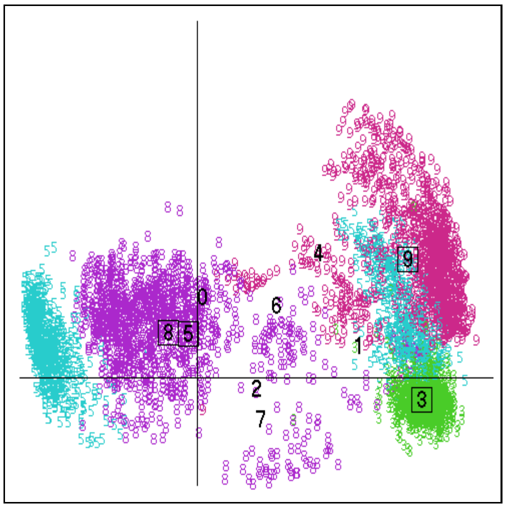


Figure 21: Projection to discriminate 3, 5, 8 and 9

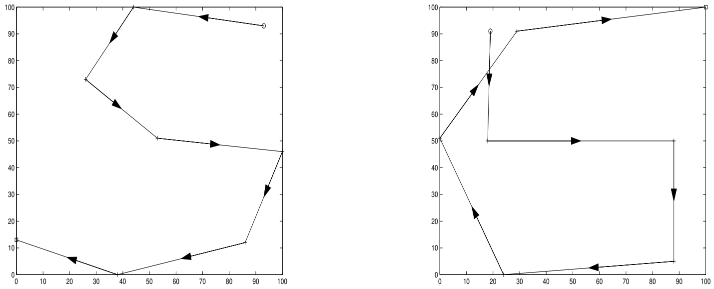


Figure 22: Different types of 5

very distinct clouds — the first is near 3 *and* 9*, while the second is to the left of* 8 *and far from* 3 *and* 9*. The centroid for the digit* 5 *is seen to lie in the middle of these two clouds and very close to the* 8 *centroid, but there are no individual data samples near the centroid!* This behavior invites a closer look at handwritten samples of 5.

Figure 22 shows two quite different ways of writing 5 and explains the two different clouds in Figure 21. The left plot of Figure 22 gives the cursive way of writing 5 where there is no "pen lift", and these 5's are seen to lie in a cloud near 9 and 3. Indeed, the left plots in Figures 20 and 22 exhibit the similarity between 9 and the cursive 5. The right plot of Figure 22 shows the non-cursive 5 where the writer lifts the pen during writing. These 5's are somewhat similar to some of the 8's shown in Figure 19. The two clouds of 5 are also visible in Figure 23 (one below 8 and one near 3 and 9), which displays a projection that attempts to discriminate between all 10 digits.

This visual discovery was unexpected but pleasing. It underscores the power of visualization — in this case, visual discovery can play a role in the design of a better handwriting recognition system by making the classifier recognize two types of 5, and thus recognize eleven "different" digits.

6. Finally, in Figure 24 we draw the centroids of all the ten digits in the same manner we

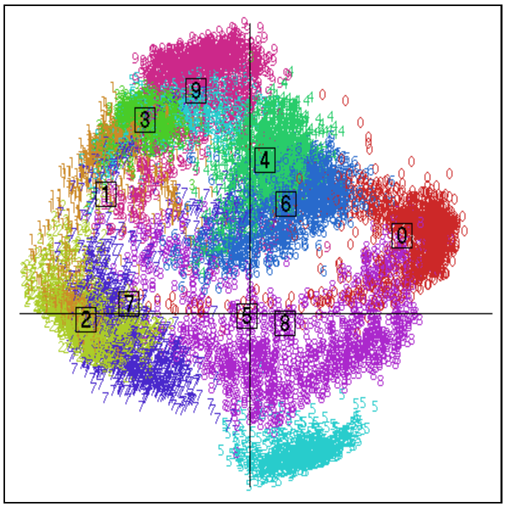


Figure 23: Projection discriminates all the ten digits in the PENDIGITS data

displayed individual digit samples in Figure 13. We observe that the class centroids for 2, 3, 4 and 6 look like "normal" written digits. This observation is consistent with our visual explorations where we found these classes to be quite coherent, for example, see Figure 17. We also observe that the 5 and 9 centroids bear little resemblance to the corresponding written digits. The centroid for 5 is the average of the cursive and non-cursive 5, while the average 9 reflects the confusion between the clockwise and anti-clockwise arcs in writing 9 (see Figures 20 and 22).

6 Conclusions

In this paper, we have proposed the use of *class-preserving projections* and *class-eigenvector plots* for *visual discriminant analysis*. These projections satisfy a certain optimality criterion that attempts to preserve distances between the class-means. Our projections are similar to Fisher's linear discriminants that are commonly used in classical discriminant analysis, but are faster to compute and hence are better suited for interactive visualization. *Class-similarity graphs* remind us of proximity relations between classes in each of our class-preserving projections. We also use *class tours* which enable us to view sequences of class-preserving projections, interspersed with interpolating projections that create an illusion of smooth motion through a multidimensional subspace. Class tours allow the touring of all the data points in an informative class preserving linear subspace. All the above ideas comprise our visualization software toolbox CViz that allows us to capture the inter-class structure of complex, multidimensional data.

We have illustrated the use of our visualization tool in discovering interesting class relationships in the Iris, ISOLET and PENDIGITS data. Such discoveries underscore the value of visualization as a quick and intuitive way of understanding a data set. For illustration purposes, we have deliberately used intuitive speech and handwriting data sets, where our visual explorations lead to "natural" conclusions. In cases where there is not much domain knowledge

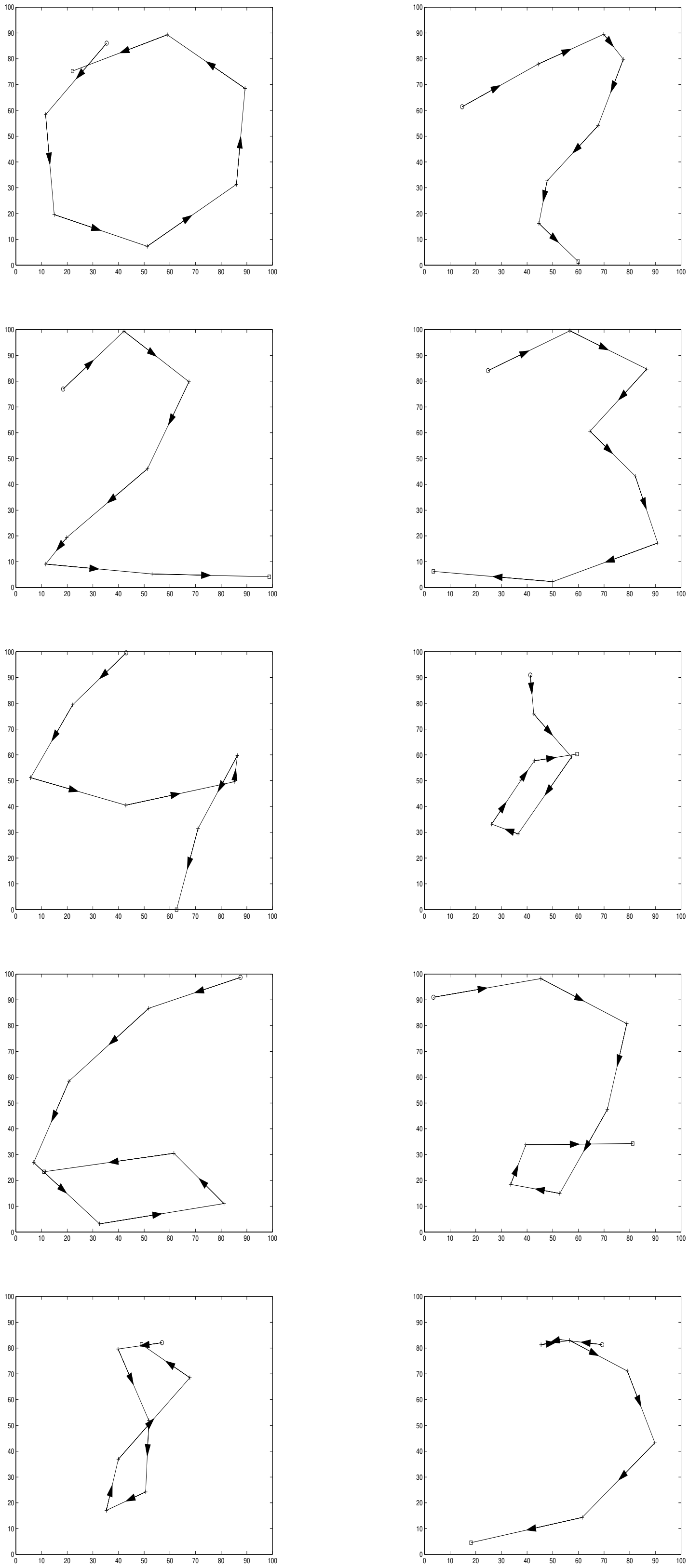


Figure 24: The "average digits" in the PENDIGITS data

about the data, such visual discoveries would be even more valuable.

Free Software: We encourage interested readers to try CViz — the JAVA software tool based on ideas in this paper which can be downloaded from: http://www.alphaworks.ibm.com/tech/cviz.

Acknowledgements. We are grateful to Werner Stuetzle for bringing the work in [4] to our attention, and to Sriganesh Madhvanath for help in the area of handwriting recognition. Comments by anonymous referees greatly improved our presentation.

Appendix

Proposition. *The matrices S_B given in equations (4) and (5) are identical.*

Proof. Expanding each term in equation (4) and using symmetry in i and j, we get

$$\begin{aligned}
S_B &= \sum_{i=1}^{q} \sum_{j=1, j\neq i}^{q} n_i n_j (m_i m_i^T - m_i m_j^T) \\
&= \sum_{i=1}^{q} \left\{ n_i(n_1 + n_2 + \cdots + n_q) m_i m_i^T - n_i^2 m_i m_i^T \right\} \\
&\quad - \sum_{i=1}^{q} \sum_{j=1, j\neq i}^{q} n_i n_j m_i m_j^T, \\
&= n^{(q)} \sum_{i=1}^{q} n_i m_i m_i^T - \sum_{i=1}^{q} \sum_{j=1}^{q} n_i n_j m_i m_j^T, \\
&= n^{(q)} \left(\sum_{i=1}^{q} n_i m_i m_i^T - n^{(q)} m^{(q)} m^{(q)T} \right), \\
&= n^{(q)} \sum_{i=1}^{q} n_i (m_i - m^{(q)})(m_i - m^{(q)})^T.
\end{aligned}$$

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* John Wiley & Sons, 2nd ed., 2000.

[2] J. A. Hartigan, *Clustering Algorithms.* Wiley, 1975.

[3] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2325–2383, 1998.

[4] R. Gnanadesikan, J. R. Kettenring, and J. M. Landwehr, "Projection plots for displaying clusters," in *Statistics and Probability: Essays in Honor of C. R. Rao* (G. Kallianpur, P. R. Krishnaiah, and J. K. Ghosh, eds.), pp. 269–280, North-Holland Publishing Company, 1982.

[5] M. Fanty and R. Cole, "Spoken letter recognition," in *Advances in Neural Information Processing Systems* (R. P. Lippman, J. Moody, and D. S. Touretzky, eds.), vol. 3, (San Mateo, CA), Morgan Kaufmann, 1991.

[6] F. Alimoğlu and E. Alpaydin, "Methods of combining multiple classifiers based on different representations for pen-based handwriting recognition," in *Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96)*, (Istanbul, Turkey), June 1996.

[7] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: the QBIC system," *IEEE Computer*, vol. 28, pp. 23–32, September 1995.

[8] G. Salton and M. J. McGill, *Introduction to Modern Retrieval*. McGraw-Hill Book Company, 1983.

[9] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Transactions on Computers*, vol. C-23, no. 9, pp. 881–890, 1974.

[10] P. J. Huber, "Projection pursuit," *Annals of Statistics*, vol. 13, pp. 435–475, 1985.

[11] T. Kohonen, *Self-organizing Maps*. Berlin: Springer, 1995.

[12] J. B. Kruskal, "Nonmetric multidimensional scaling: A numerical method," *Psychometrika*, vol. 29, pp. 115–129, 1964.

[13] J. B. Kruskal, "Multidimensional scaling and other methods for discovering structure," in *Statistical Methods for Digital Computers* (K. Enslein, A. Ralston, and H. S. Wilf, eds.), pp. 296–339, Wiley, 1977.

[14] C. Hurley and A. Buja, "Analyzing high-dimensional data with motion graphics," *SIAM J. Sci. Stat. Comput.*, vol. 11, pp. 1193–1211, November 1990.

[15] D. Cook, A. Buja, and J. Cabrera, "Projection pursuit indices based on expansions with orthonormal functions," *Journal of Computational and Graphical Statistics*, vol. 2, no. 3, pp. 225–250, 1993.

[16] D. Cook, A. Buja, J. Cabrera, and C.Hurley, "Grand tour and projection pursuit," *Journal of Computational and Graphical Statistics*, vol. 4, no. 3, pp. 155–172, 1995.

[17] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Transactions on Neural Networks*, vol. 6, pp. 296–317, March 1995.

[18] M. A. Kraaijveld, J. Mao, and A. K. Jain, "A nonlinear projection method based on Kohonen's topology preserving maps," *IEEE Transactions on Neural Networks*, vol. 6, pp. 548–559, May 1995.

[19] J. Vesanto, "SOM-based data visualization methods," *Intelligent Data Analysis*, vol. 3, pp. 111–126, April 1999.

[20] G. Grinstein, A. Buja, D. Asimov, and A. Inselberg, "Visualizing multidimensional (multivariate) data and relations-perception vs. geometry," in *Proceedings IEEE Visualization'95* (G. M. Nielsen and D. Silver, eds.), pp. 405–411, IEEE Computer Society Press, Los Alamitos, CA, October 1995.

[21] D. Asimov, "The grand tour: A tool for viewing multidimensional data," *SIAM J. Sci. Stat. Comput.*, vol. 6, pp. 128–143, January 1985.

[22] D. Asimov and A. Buja, "Grand tour methods: an outline," in *Proceedings of the 17th Symposium on Interface of Computer Science and Statistics*, 1985.

[23] A. Buja, D. Cook, D. Asimov, and C. Hurley, "Dynamic projections in high-dimensional visualization: Theory and computational methods," tech. rep., AT&T Labs, Florham Park, NJ, 1997.

[24] D. F. Swayne, D. Cook, and A. Buja, "XGobi: Interactive dynamic data visualization in the X window system," *Journal of Computational and Graphical Statistics*, vol. 7, no. 1, pp. 113–130, 1998.

[25] I. S. Dhillon, D. S. Modha, and W. S. Spangler, "Visualizing class structure of multidimensional data," in *Proceedings of the 30th Symposium on the Interface: Computing Science and Statistics* (S. Weisberg, ed.), vol. 30, (Minneapolis, MN), pp. 488–493, Interface Foundation of North America, May 1998.

[26] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, pp. 143–175, January 2001.

[27] C. Blake, E. Keogh, and C. J. Merz, "UCI repository of machine learning databases," *http://www.ics.uci.edu/~mlearn/MLRepository.html*, 1998.

[28] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. Part II, pp. 179–188, 1936. also in *Contributions to Mathematical Statistics* (John Wiley, New York, 1950).

[29] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis.* Academic Press, 1979.

[30] J. G. Bryan, "The generalized discriminant function: mathematical foundation and computational routine," *Harvard Educ. Rev.*, vol. 21, pp. 90–95, Spring 1951.

[31] S. Kullback, *Information Theory and Statistics.* John Wiley, New York, 1959.

[32] G. H. Golub and C. F. V. Loan, *Matrix computations.* Johns Hopkins University Press, 3rd ed., 1996.

[33] M. Fanty and R. Cole, "Spoken letter recognition," in *Proceedings of the DARPA Workshop on Speech and Natural Language Processing*, (Hidden Valley, PA), June 1990.

[34] A. Björck and G. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematics of Computation*, vol. 27, pp. 579–594, July 1973.

[35] E. H. Chi, P. Barry, J. Riedl, and J. Konstan, "A spreadsheet approach to information visualization," in *Proceedings of the Symposium on Information Visualization (InfoVis '97)*, (Phoenix, Arizona), pp. 17–24, IEEE CS, 1997.

[36] W. Ribarsky, E. Ayers, J. Eble, and S. Mukherjea, "Glyphmaker: Creating customized visualizations of complex data," *IEEE Computer*, vol. 27, pp. 57–64, July 1994.

[37] F. Alimoğlu, "Combining multiple classifiers for pen-based handwritten digit recognition," M.Sc. thesis, Institute of Graduate Studies in Science and Engineering, Boğaziçi University, Istanbul, Turkey, 1996.