Published in final edited form as:
Int Conf Digit Signal Process Proc. 2008 December 11; 2008: 426–433. doi:10.1109/ICMLA.2008.46.

Force feature spaces for visualization and classification

Dragana Veljkovic and **Kay A. Robbins**
Department of Computer Science, University of Texas at San Antonio, USA

Dragana Veljkovic: dveljkov@cs.utsa.edu; Kay A. Robbins: krobbins@cs.utsa.edu

Abstract

Distance-preserving dimension reduction techniques can fail to separate elements of different classes when the neighborhood structure does not carry sufficient class information. We introduce a new visual technique, K-epsilon diagrams, to analyze dataset topological structure and to assess whether intra-class and inter-class neighborhoods can be distinguished.

We propose a force feature space data transform that emphasizes similarities between same-class points and enhances class separability. We show that the force feature space transform combined with distance-preserving dimension reduction produces better visualizations than dimension reduction alone. When used for classification, force feature spaces improve performance of K-nearest neighbor classifiers. Furthermore, the quality of force feature space transformations can be assessed using K-epsilon diagrams.

1. Introduction

A common approach for analysis and visualization of high-dimensional datasets is to reduce the data through dimension reduction and feature extraction while preserving as much of the underlying information as possible. The goal is to find a two-dimensional or three-dimensional representation of the original dataset that separates different classes into visually intuitive groups while preserving the initial structure. Low-dimensional projection techniques have been widely applied in the past decade; see Saul [1] and van der Maaten [2] for reviews and comparison of the existing methods. The algorithms attempt to preserve either global or local data similarities by maximizing different optimization criteria.

Though the emphasis has been on distance and neighborhood preservation, little work has been done in analysis of topological neighborhood structures, their preservation in the projection space, and their effect on classification. In this paper we introduce a visual method for analysis of these topological neighborhood structures in high-dimensional data called K-epsilon diagrams. These diagrams show the changes in the distribution of point density in local neighborhoods, with respect to the neighborhood radius. Datasets with excessively unstructured neighborhood topologies are usually deemed poor candidates for visualization through low-dimensional projections. By analyzing changes in neighborhood topologies between the original space and the projection space, we can estimate how well the low-dimensional projection preserves the point relations.

A desirable trait of a low-dimensional projection is intuitive visual separation of points belonging to different classes. The degree of visual class separation is often used as a measure of projection quality. In this paper we compare neighborhood topologies between pairs of points in the same class with pairs of points in different classes. If these datasets have a substantial number of points in different classes that are close by, projection techniques that rely on distance and point neighborhood preservation will fail to visually separate classes, because there is not enough class information contained in distances. In these cases additional information needs to be included in the computation of the low-dimensional projection to properly separate classes of data.

We propose a technique that overcomes this topological obstacle by incorporating class information into the data representation. The dataset is lifted to a feature space and points in the feature space are rearranged using a force directed approach. The new representation, called the force feature space (FFS), is designed to emphasize similarities between points in the same class and to enhance class separability. We show that the FFS transform combined with a distance-preserving low-dimensional projection significantly increases the quality of the resulting visualizations. When used for classification, force feature space improves performance of K-nearest neighbor classifiers.

1.1. Related work

This section presents related work in fields of dimension reduction and projection quality, force directed placement (FDP), and semi-supervised clustering.

Commonly-used linear dimension reduction methods include principal component analysis (PCA), multidimensional scaling (MDS) and independent component analysis (ICA). The most popular nonlinear methods include isometric feature mapping (ISOMAP), locally linear embedding (LLE), Laplacian Eigenmaps (LE), maximum variance unfolding (MVU) and kernel PCA. Recently, Saul et al. [1] showed that these seemingly unrelated methods are all rooted in spectral decomposition of various inner-product or distance based matrices. Ham et al. [3] show that ISOMAP, LLE and LE can be viewed as special cases of kernel PCA using data-dependent kernel matrices instead of predefined kernel functions. Xiao et al. [4] show that ISOMAP, LE and LLE are strongly related to either the primal or the dual MVU problem. Lespinats et al. [5] propose data driven high-dimensional scaling (DD-HDS) algorithm, specifically designed to accommodate specific distance distribution of high-dimensional data and minimize both tears and false overlaps in projections. Venna and Kaski [6] propose a method called local multidimensional scaling that optimizes trustworthiness and continuity of projections. A detailed comparative review with applications to real and artificial data is given by [2]. The authors conclude that though nonlinear methods work better on artificial data, PCA performs as well if not better on real datasets.

Force directed placement (FDP) for graph visualization was first introduced by Eades in [7]. The technique models graph connections as springs and realigns positions of nodes based on forces applied to each node. The FDP idea has recently been used by Lespinats et al. [5] to compute the low-dimensional projection that minimizes tearing of neighborhood connections. Omote et al. [8] apply a modification of the FDP method to draw intersecting clustered graphs of complex structures. Noack [9] develop edge repulsion and energy based clustering to find dense subgraphs using normalizing cuts. Wittkop et al. [10] introduce the FORCE clustering algorithm for detecting groups of functionally related proteins. The attraction and repulsion forces are calculated using a thresholded similarity function. Santamaria et al. [11] develop a bicluster visualization tool that uses force directed graph placement for cluster placement and visualization.

2. The K-epsilon diagram

This section introduces the K-epsilon diagram, a new visual approach for analysis of dataset topology. The K-epsilon diagram intuitively represents dataset topology and provides feedback on how the projection affects both small and large distances, as well as feedback on the preservation of global neighborhood structure.

Special cases of the K-epsilon diagram, called the intra-class and inter-class K-epsilon diagrams, are used to evaluate whether the dataset topology carries enough class information to produce a quality low-dimensional projection.

2.1. Definition

A K-epsilon diagram plots the two-dimensional histogram of distances of nearest neighbors.

Let X = {$\mathbf{x}_1$, …, $\mathbf{x}_n$} be a dataset in a *D*-dimensional space and let $d(\mathbf{x}_i, \mathbf{x}_j)$ be the scaled distance between point *i* and point *j*. All distances are scaled to be in the interval [0, 1]. Furthermore, let $N_i(k)$ be the set of *k* nearest neighbors of point *i*. We define $\varepsilon_i(k)$ as the minimum radius of the ε-ball centered at point *i* that encapsulates all points in $N_i(k)$:

$$\varepsilon_i(k)=\max_{j\in N_i(k)} d(\mathbf{x}_i,\mathbf{x}_j).$$

Let *b* be the number of histogram bins in the K-epsilon diagram. The computed ε-ball radius is discretized to a bin number by:

$$p_i(k)=\text{floor}(\varepsilon_i(k)\cdot b)+1.$$

The K-epsilon diagram of the dataset X is defined as the cumulative two-dimensional histogram $H(p_i(k), k)$ where $i = 1, \dots, n$ and $k = 1, \dots, n-1$. For each value of *i* and *k*, the corresponding histogram bin count $H(p_i(k), k)$ is increased by one. The vertical axis of the K-epsilon diagram denotes the nearest neighbor number *k*, while the horizontal axis denotes the radius of the smallest epsilon ball containing that neighbor $p_i(k)$.

The number of bins, *b*, affects the granularity of the diagram. A small value of *b* produces a coarse diagram, while a large number of bins results in a histogram that has too few points per bin. The diagrams shown in this paper have *b* = 100. A darker color indicates more elements in the corresponding bin. The epsilon values are in the interval [0, 1]. The nearest neighbor number ranges from 1 to the number of points minus 1.

2.1.1. Intra- and inter-class K-epsilon diagrams—

To compare the topological structure of points in the same class with that of points in different classes, we compute the intra-class and inter-class K-epsilon diagrams.

Only neighborhood relations and distances between pairs of points belonging to the same class are considered when computing the intra-class diagram. The algorithm in section 2.1 is modified so that only points that are in the same class as the point *i* are included in the $N_i(k)$ neighborhood. The maximum value of *k* depends on the number of points belonging to the corresponding class.

For the inter-class diagrams, only neighborhood relations and distances between pairs of points belonging to different classes are considered. In this case, only points that are not in the same class as the point *i* are included in the $N_i(k)$ neighborhood.

2.2. Analysis of dataset topology

We illustrate the analysis of topological neighborhood structure with the Fisher iris dataset, available from the UCI repository [12]. The dataset contains three classes, 50 instances each. The dataset is four-dimensional, with each dimension measuring a single property of the plant. The data is preprocessed by normalizing each dimension to have the zero mean and unit variance. Topology is defined using the Euclidean distance.

Fig. 1a shows the two-dimensional MDS projection of the complete iris dataset. Class 1 (red) is linearly separable from the other two, whereas classes 2 (yellow) and 3 (blue) have significant overlap.

For illustration purposes, we also use a subset of the iris dataset containing only classes 1 and 2. These two classes are separable, as shown in Fig. 1b.

Fig. 2 shows the K-epsilon diagrams of the complete iris dataset. The top row shows the neighborhood structure in the original space, and the bottom row shows the neighborhood structure of the MDS projection. Intra- and inter- class diagrams for the iris dataset are shown in the middle and right column of Fig. 2, respectively.

The top left panel of Fig. 2 shows that most points have nearest neighbors close by. This is a desirable quality for computing low-dimensional projections, and confirms that point-neighborhoods are well-sampled for this dataset.

The top middle panel shows that the majority of same-class nearest neighbors are close by and that the distance between most same-class points is less than 50% of the maximum dataset distance. Note that the top two thirds of the diagram contains no points. For each point in the iris dataset, two-thirds of the points belong to a different class and are not shown in the intra-class diagram.

The small gap between the diagram lines and the bottom-left corner in the top right diagram indicates the existence of several pairs of inter-class points that are close by. Distance preserving projections attempt to map these points close by, compromising class separability.

The bottom row of Fig. 2 displays the corresponding K-epsilon diagrams of the two-dimensional MDS projection. The similarity with diagrams in the original space indicates that the MDS projection closely preserves the topological structure of the iris dataset.

To illustrate the desired structure of K-epsilon diagrams, we compute diagrams for the partial iris dataset shown in Fig. 1b. The large gap in the inter-class diagrams of the reduced dataset shows that the nearest neighbors of different classes are significantly separated (right column of Fig. 3). The intra-class diagrams confirm that most of the same-class nearest neighbors are close to one another (middle column of Fig. 3). The structure of these diagrams indicates that distance-preserving projections will be able to preserve class separability.

To illustrate topologies of more complex datasets, we compute the K-epsilon diagrams of the liver and Monks2 datasets from the UCI database. The first column of Fig. 4 shows the K-epsilon diagrams of the liver dataset in the original space. The poor quality of the projection of this dataset, as seen in Fig. 12a, can be predicted by the lack of distinction between the intra-class and inter-class diagrams. The steep curve of the K-epsilon diagram of the complete dataset, shown in the left panel, indicates that there is not a significant difference between distances to neighbors of various orders.

The diagrams in the second row correspond to the Monks2 dataset. The discontinuous shape of the diagrams suggests a discrete distribution of distances. The steepness of the shapes indicates the existence of several groups of neighbors of different orders that have nearly identical distance to the base point. The gap in both of these diagrams is significant. Intra-class and inter-class diagrams are almost identical. These properties of the K-epsilon diagrams indicate highly overlaid classes and explain the poor classifier performance seen in Table 2.

2.3. Topology of low-dimensional projections

Finding reliable 2D or 3D projections is important for visualization of high-dimensional data. In this section we compare the topology preservation of three commonly-used dimension reduction techniques, MDS, ISOMAP, and LE. We apply these techniques to visualize the diagnostic breast cancer data [12]. The dataset contains 569 samples, each with 30 attributes. All samples are classified as benign or malignant. The data in each dimension is normalized to have zero mean and unit variance, and the topology is defined using the Euclidean distance.

Fig. 5 shows the two-dimensional projections of the breast cancer dataset using the three projection techniques. Class 1 (malignant) is shown in red, and class 2 (benign) is shown in yellow. We set the number of nearest neighbors to 2 for ISOMAP and to 15 for LE. The shapes and forms of the three projections vary, and the classes overlap.

We compare the neighborhood topologies of the three dimension reduction techniques using K-epsilon diagrams. The first row of Fig. 6 displays the three K-epsilon diagrams of the original breast cancer data. The smooth lines in the left panel indicate a consistent topological structure. A gap between the beginning of the curve and the bottom left corner indicates that even the smallest distances between nearest neighbors in this dataset are above a certain threshold. High-dimensionality, noise and complexity of the data are the main causes of this behavior. A big gap in the K-epsilon diagram indicates that a dataset is a poor candidate for visualization using a distance-preserving dimension reduction technique. In the breast cancer data, the gap size is not significant enough to affect the performance of dimension-reduction algorithms.

The intra-class diagram in the middle panel consists of two well defined shapes, suggesting existence of two data classes of different topologies. The inter-class diagram shown in the right panel also suggests the presence of two classes. However, similar gaps in intra-class and interclass diagrams indicate that strategies that only preserve distances will not be able to separate the two classes.

Rows two to four of Fig. 6 show the corresponding K-epsilon diagrams of the two-dimensional projections. The overall topological structure is best preserved with MDS. All three projections somewhat preserve the intra- and inter- class topologies, which also indicates that the projections fail to separate between data classes. Note that the projection diagrams do not have a gap in the bottom left corner.

3. Force feature space

We propose a FDP-based transformation that emphasizes both similarities between intra-class points and dissimilarities between inter-class points. Let X = {$\mathbf{x}_1, \ldots, \mathbf{x}_n$} be the original dataset in a D-dimensional space and let Y = {$\mathbf{y}_1, \ldots, \mathbf{y}_n$} be the dataset after the transform. We will refer to Y as the force feature space.

The algorithm proceeds as follows. Initially, $\mathbf{y}_i^0=\mathbf{x}_i$. In each iteration, every point $\mathbf{y}_i^t$ in the force feature space experiences a force from all other points, $\mathbf{y}_j^t$, $i \neq j$. If point j is in the same class as point i, an attractive force F_A pushes $\mathbf{y}_i^t$ towards $\mathbf{y}_j^t$:

$$F_A(\mathbf{y}_i^t, \mathbf{y}_j^t)=\exp(-\alpha_A t)\left[1-\exp(-\lambda_A d(\mathbf{y}_i^t, \mathbf{y}_j^t))\right]\mathbf{v}.$$

Here t denotes the iteration number, $\mathbf{v}$ is a unit vector in direction $\mathbf{y}_j - \mathbf{y}_i$, and α_A, and λ_A are scale parameters. The first exponential function reduces the attractive force magnitude with each iteration, ensuring convergence of the algorithm. The second exponential magnitude function provides strong attractive forces for distant points in the same class. $d(\mathbf{y}_i, \mathbf{y}_j)$ is the scaled distance between $\mathbf{y}_i$ and $\mathbf{y}_j$. We use Euclidean distance in this paper, but any symmetric measure of similarity can be used. The distances are scaled to be in the interval [0, 1].

If points $\mathbf{y}_i^t$ and $\mathbf{y}_j^t$ are not in the same class, a repulsive force F_R pushes $\mathbf{y}_i^t$ away from $\mathbf{y}_j^t$:

$$F_R(\mathbf{y}_i^t, \mathbf{y}_j^t) = -\exp(\alpha_R t)\exp\left[-\lambda_R d(\mathbf{y}_i^t, \mathbf{y}_j^t)\right]\mathbf{v},$$

where α_R are λ_R scale parameters. The second exponential function provides strong repulsive forces for neighboring points that are not in the same class.

The total force for $\mathbf{y}_i^t$ in iteration t, $F_T(i, t)$ is the sum of all forces acting on $\mathbf{y}_i^t$. The position of each point is adjusted according to the total force acting on that point:

$$\mathbf{y}_i^{t+1} = \mathbf{y}_i^t + F_T(i, t)\Delta^2,$$

where Δ is the time increment. These steps are repeated for a preset number of iterations or until the system becomes stable.

Let τ denote the final iteration. The value of $\mathbf{y}_i$ is set as $\mathbf{y}_i = \mathbf{y}_i^\tau$. The cumulative force F_C on $\mathbf{y}_i$ is defined as the sum of total forces F_T over all iterations:

$$F_C(i) = \sum_{t=0}^{\tau-1} F_T(i, t).$$

Note that the cumulative force F_C is equal to the force needed to move point $\mathbf{y}_i^0$ to its final position $\mathbf{y}_i$:

$$\mathbf{y}_i = \mathbf{y}_i^0 + F_C(i)\Delta^2.$$

3.1. Force feature space illustrations

We illustrate lifting to force feature space using the iris dataset with parameters $\lambda_A = \lambda_R = 3$, $\alpha_A = \alpha_R = 0.7$, and $\Delta = 0.13$. Fig. 7a and 7b show the MDS projection of the original and the force feature space data, respectively.

Points in classes 2 and 3 of Fig. 7a are overlaid, while those of Fig. 7b show good class separation. Furthermore, the shape and structure of the three clusters in Fig. 7a are well preserved in Fig. 7b.

The top and bottom rows of Fig. 8 show K-epsilon diagrams of the FFS representation of the iris data and its MDS projection. The K-epsilon diagram of the complete dataset, shown in the top left panel, resembles a 3-step-function. This structure suggests existence of three clusters with similar topologies, confirmed by Fig. 7b.

The top middle panel of Fig. 8 is similar to the top middle panel of Fig. 2, indicating that FFS transform preserves intra-class topologies. The top right panels in Fig. 2 and Fig. 8, however, are significantly different. The diagram shown in Fig. 8 has a significant gap between dark lines and the bottom left corner. The gap indicates that the distances between nearest neighbors of different classes are above a certain threshold. The step-function shape suggests the presence of three well-separated clusters in the force feature space. The bottom row of Fig. 8 illustrates that the MDS projection of the force feature space preserves neighborhood topologies.

Fig. 9a shows the MDS projection of the breast cancer data. Fig 9b shows the same dataset lifted to the force feature space using $\lambda_A = \lambda_R = 3$, $\alpha_a = \alpha_R = 0.7$, and $\Delta = 0.12$. The two classes are clearly separated, and the cluster shape shown in Fig. 9a is preserved.

The K-epsilon diagrams of the breast cancer dataset in the force feature space are shown in the first row of Fig. 10. The upper-left diagram reveals the existence of two clusters with different local topologies. Each class is shown as a step-function, suggesting that points in the same cluster are separated from those in the other. This observation is confirmed by the intra-class and inter-class diagrams shown in the middle and right panels, respectively. These diagrams show that same-cluster points have small epsilon-radiuses for nearest neighbors, while points from the other cluster have large epsilon-radiuses for a significant number of nearest neighbors. The shape of the intra-class and inter-class diagrams is well-preserved between the original data and the feature space data, indicating preservation of neighborhood topologies. Note that the force feature space diagram does not have the gap visible in the original space diagram.

The bottom row of Fig. 10 shows the K-epsilon diagrams of the MDS projection of the force feature space. As in the original data, the topology of the force feature space is well-preserved in the MDS projection.

4. Classification using force feature spaces

Section 3 describes the use of force feature spaces for visualization of known classes. This section applies the force feature space for classification.

Let X = $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be the original dataset in a D-dimensional space and Y = $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ be the corresponding set in the force feature space. Let **z** be a new point of unknown class. We wish to find a point **w** in the feature space that best represents the point **z**.

The position of the point **w** is determined by an iterative algorithm that combines cumulative forces of its k nearest neighbors. Let $\mathbf{w}^t$ be the position of the point **w** in iteration t. Initially $\mathbf{w}^0 = \mathbf{z}$.

In the first iteration, the k nearest neighbors of point $\mathbf{w}^0$ are $\mathbf{x}_j$, $j \in N^0$. Point $\mathbf{w}^0$ lies in the original space, and its neighbors are selected from the original dataset X. The force $G(0)$ acting on the point $\mathbf{w}^0$ is defined as the average of cumulative forces of its nearest neighbors scaled by a function of distance of $\mathbf{w}^0$ from that neighbor:

$$G(0)=\frac{1}{k}\sum_{j\in N^0}\left[F_c(j)\exp(-\lambda d(\mathbf{w}^0,\mathbf{x}_j))\right].$$

Here λ is mean of λ_A and λ_R, $F_C(j)$ is the cumulative force on the j^{th} data element, and $d(\mathbf{w}^0, \mathbf{x}_j)$ is the scaled distance between point $\mathbf{w}^0$ and its j^{th} nearest neighbor in the original space, $\mathbf{x}_j$. All the distances have been scaled by the maximum distance to point $\mathbf{w}^0$.

At the end of first iteration the point $\mathbf{w}^0$ is moved in the direction of the force $G(0)$ and the new position $\mathbf{w}^1$ is computed:

$$\mathbf{w}^1=\mathbf{w}^0+G(0)\Delta^2.$$

In iteration two and all subsequent iterations, the position of the point $\mathbf{w}^t$ in the feature space is adjusted using the cumulative forces acting on its current nearest neighbors $\mathbf{y}_j$, $j \in N^t$. Note that $\mathbf{w}^t$ now lies in the force feature space, so its neighbors are sought among points in Y. In iteration t the force acting on point $\mathbf{w}^t$ is similar to the force computed in the first iteration:

$$G(t)=\frac{\exp(-\beta t)}{k}\cdot\sum_{j\in N^t}\left[F_C(j)\exp(-\gamma d(\mathbf{w}^t,\mathbf{y}_j))\right].$$

Here β and γ are scale parameters, and $d(\mathbf{w}^t,\mathbf{y}_j)$ is the scaled distance between points $\mathbf{w}^t$ and its j^{th} nearest neighbor in the feature space. As in iteration one, these distances are scaled by the maximum distance to $\mathbf{w}^t$. The new position $\mathbf{w}^{t+1}$ is computed as:

$$\mathbf{w}^{t+1}=\mathbf{w}^t+G(t)\Delta^2.$$

These steps are repeated for a preset number of iterations or until the system converges. Let τ denote the final iteration. The value of $\mathbf{w}$ is set as $\mathbf{w} = \mathbf{w}^\tau$. Classification can then be performed in the force feature space to estimate the class of the point $\mathbf{z}$.

4.1. Results on real data

To demonstrate force feature space based classification we use selected datasets from the UCI repository [12] with ten times 10-fold cross validation. The datasets and their parameters are shown in Table 1. In all datasets $\alpha_A = 0.7$, and for the first five datasets $\tau = 6$. For the three Monk's datasets $\tau = 9$. We tested FFS-based classification using Matlab implementations of the K-nearest neighbors classifier (KNN) with two nearest neighbors and the support vector machine classifier (SVM).

Table 2 compares the performance of the KNN and SVM classifiers before and after mapping to force feature space. Mapping to force feature space improved the performance of the KNN classifier, making it competitive with the SVM. The FFS transformation does not have as much of an effect on the correct rate of the SVM classifier in most datasets. However, the FFS remapping makes the projections more understandable.

Fig. 11 shows the MDS projection of one fold of 10-fold cross validation of the breast cancer data using the KNN classifier. The classifier output is denoted by shape: squares for class 1, and triangles for class 2. The ground truth for all points is denoted by color: red for class 1, and yellow for class 2. Fig. 11a shows the classification in the original space. The classes overlap and the black arrow highlights a misclassified point. Fig. 11b shows the classification in force feature space. The classes are well separated and form clusters in the projection space. The KNN FFS-classifier has a high correct rate, which is expected because the inter- and intra-class diagrams of the dataset show good class separability (Fig. 10).

Fig. 12 shows a similar MDS projection for the liver dataset. The classifier correct rate for this dataset is significantly lower. The two classes have a significant overlap, and the force feature space can not fully separate them without distorting the original intra-class topologies. Note that class 2 (yellow) is densely sampled while points in class 1 (red) are scattered.

The top row of Fig. 13 shows the K-epsilon diagrams of the liver dataset in the original space. The poor quality of the projection in Fig. 12a can be predicted by the lack of distinction between intra and inter class diagrams. The step-shaped curve in the bottom left panel of Fig 13 corresponds to the well-clustered yellow class after FFS mapping. The lack of discontinuity in the other shape reflects the scattered structure of the red class. This difference in structure is reflected in the middle panel.

4.2. Parameter selection

The flexibility of the force feature space transform lies in its many parameters. The FFS transform can be optimized to fit any dataset by adjusting the parameter value.

The time interval Δ defines the length of the time interval in which forces act on each point. A longer interval results in a bigger change of the point position. By increasing Δ the user can increase separability of classes in the force feature space. A Δ which is too large will result in severe distortion of the cluster shape. The default value for Δ is 0.1.

Parameters α_A and α_R define the falloff of the maximum magnitude of attractive and repulsive forces in each iteration, respectively. The default value of these parameters is 0.7. Decreasing the value of one of these parameters will increase the maximum magnitude of the corresponding forces and, in turn, the effect of the said force on the transform. For example, $\alpha_R = 0.55$ in the Monks2 dataset increases the effect of the repulsive forces and boosts class separation. Because parameter α_A remains the same, changes to attractive forces and to the cluster structure are minimal.

Parameters λ_A and λ_R define the magnitude of the attractive and repulsive forces based on the distance between the two points. The default value for these parameters is 3. Increasing λ_A augments the magnitude of the attractive forces, while decreasing λ_R augments the magnitude of the repulsive forces.

Changes in parameters β and γ have effects that are similar to the changes of parameters α_R and λ_R, respectively. The default number of iterations is 5, but this value can be increased if greater precision is needed.

5. Discussion

This paper proposes lifting the dataset to a force feature space prior to projection and/or classification to enhance visualization and improve classification. Similar ideas are seen in kernel methods. In contrast to kernel methods, the force feature spaces are explicitly represented and have the same dimensionality as the original space. When lifting the dataset to a kernel space, the user has little control over the resulting space or its properties. In fact, there is no proof that kernels unfold and simplify the data in the higher-dimensional space.

Linear discriminant analysis (LDA) uses class information to determine a linear low-dimensional projection that maximizes class separability. If the classes are not separable, LDA will fail to find a good projection. Furthermore, the projection dimensionality is bound by the class number and can not be larger than the class number minus 1, making LDA unsuitable for visualization of complex two-class datasets.

Force directed placement is often used to enhance an existing low-dimensional projection. This approach is used in DD-HDS where FDP is employed to correct positions of projection points whose original space and projection space distances were not proportional. DD-HDS uses only distance information from the original and the projected space to compute the forces.

The technique introduced in this paper incorporates both class and distance information in forming a feature space for visualization and classification. We use K-epsilon diagrams to determine how well the FFS transform distinguishes between inter- and intra-class neighborhoods. The quality of the FFS-classifier can be estimated by inter-class and intra-class separation in the K-epsilon diagrams. We can use this information as a basis for adjusting the algorithm parameters. The K-epsilon diagrams also provide information about the complexity of the dataset.

FFS combined with distance preserving dimension reduction produces intuitive visualizations that separate points belonging to different classes while preserving the original class structure. Unlike support vectors in SVM classifiers, which are often difficult to visualize, the FFS-based classifiers include intuitive visualization using force feature space positions of the testing points.

Acknowledgments

This work was supported by NIH Research Centers in Minority Institutions 2G12RR1364-06A1.

References

1. Saul, LK.; Weinberger, KQ.; Ham, J.; Sha, F.; Lee, DD. Spectral methods for dimensionality reduction. In: Chapelle, BSO.; Zien, A., editors. Semisupervised Learning. MIT Press; Cambridge, MA: 2006.
2. van der Maaten LJP, Postma EO, van den Herik HJ. Dimensionality reduction: A comparative review. Submitted to Neurocognition. 2008
3. Ham, J.; Lee, DD.; Mika, S.; Schölkopf, B. A kernel view of the dimensionality reduction of manifolds. Conference on Machine Learning; 2004. p. 369-376.
4. Xiao, L.; Sun, J.; Boyd, S. A duality view of spectral methods for dimensionality reduction. International conference on Machine learning; 2006. p. 1041-1048.
5. Lespinats S, Verleysen M, Giron A, Fertil B. DD-HDS: A method for visualization and exploration of high-dimensional data. IEEE Transactions on Neural Networks 2007;18:1265–1279. [PubMed: 18220179]
6. Venna J, Kaski S. Local multidimensional scaling. Neural Networks 2006;19:889–899. [PubMed: 16787737]
7. Eades P. A heuristic for graph drawing. Congressus Numerantium 1984;42:149–160.
8. Omote, H.; Sugiyama, K. Force-directed drawing method for intersecting clustered graphs. 6th International Asia-Pacific Symposium on Visualization; 2007. p. 85-92.
9. Noack A. Energy-based clustering of graphs with nonuniform degrees. Graph Drawing 2006:309–320.
10. Wittkop T, Baumbach J, Lobo F, Rahmann S. Large scale clustering of protein sequences with FORCE -A layout based heuristic for weighted cluster editing. BMC Bioinformatics 2007;8:396. [PubMed: 17941985]
11. Santamaría R, Therón R, Quintales L. A visual analytics approach for understanding biclustering results from microarray data. BMC Bioinformatics 2008;9:247. [PubMed: 18505552]
12. Asuncion, A.; Newman, DJ. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science; 2007. [http://www.ics.uci.edu/~mlearn/MLRepository.html]

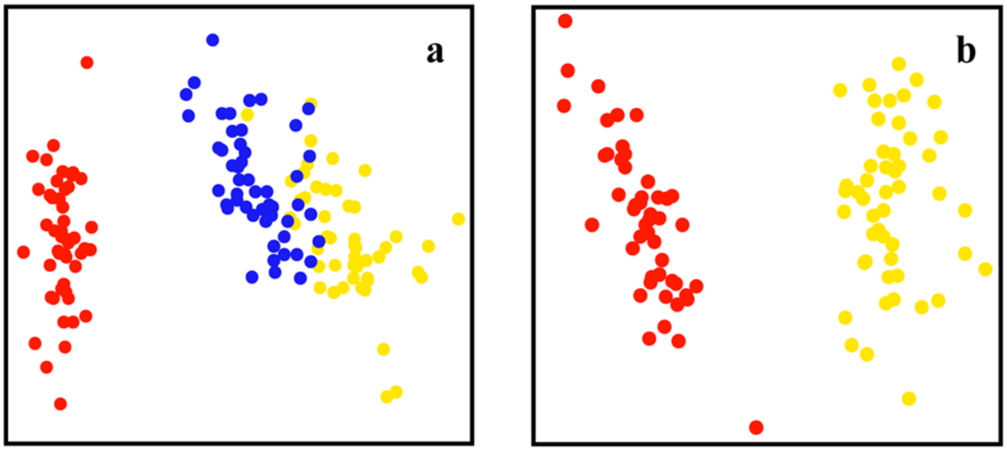

Figure 1.

(a) MDS projection of the complete iris dataset. (b) MDS projection of the subset of iris dataset containing only two classes.

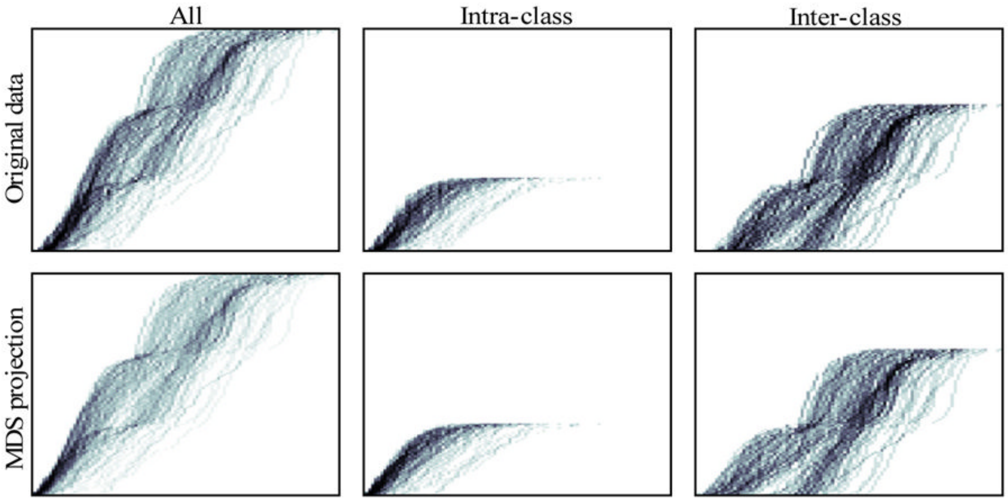

Figure 2.

K-epsilon diagrams of the complete iris dataset. Left: complete diagrams. Middle: intra-class diagrams. Right: interclass diagrams. Top row shows neighborhoods in the original space, bottom row shows neighborhoods of the 2D MDS projection.

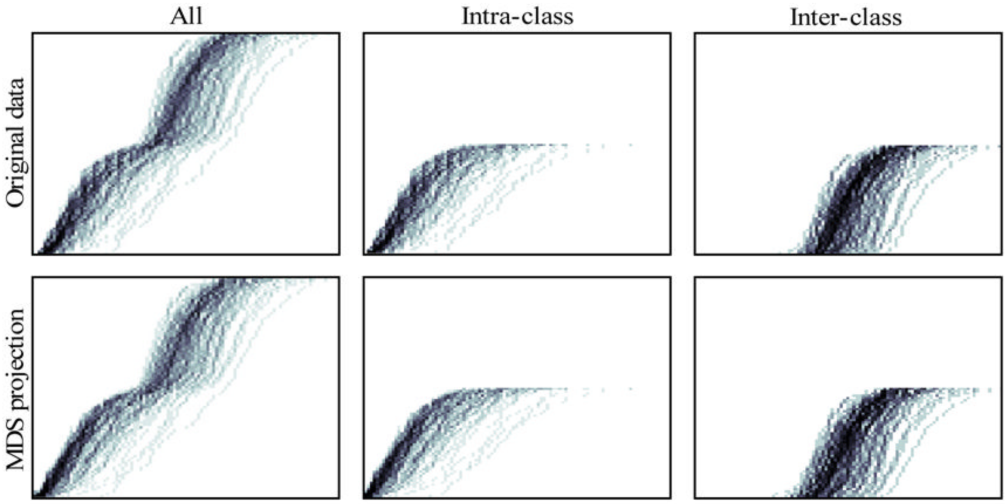

Figure 3.

K-epsilon diagrams of the partial Iris dataset shown in Fig. 1a. Rows and columns same as in Fig. 2.

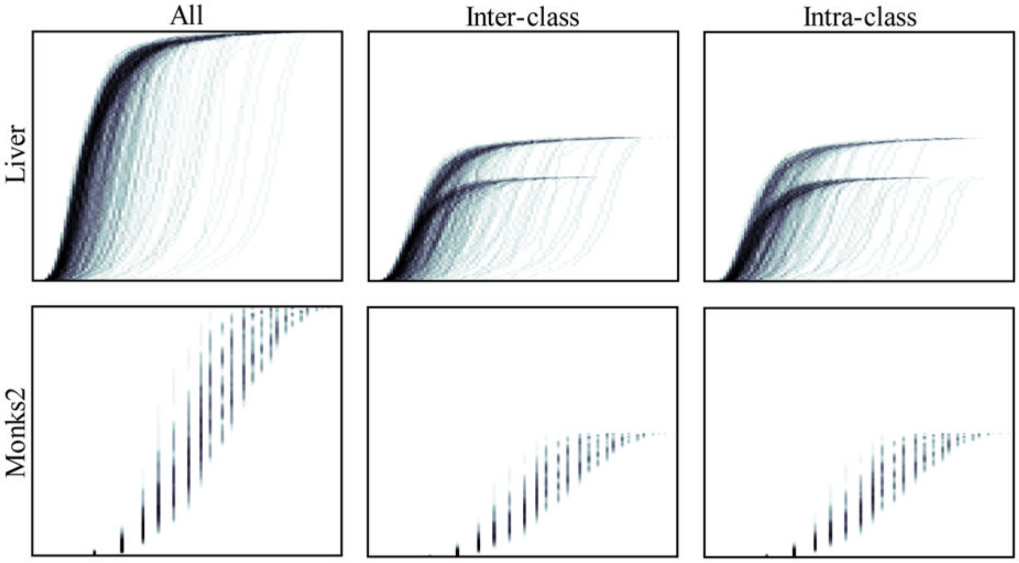

Figure 4.
K-epsilon diagrams of the liver and Monks2 datasets.

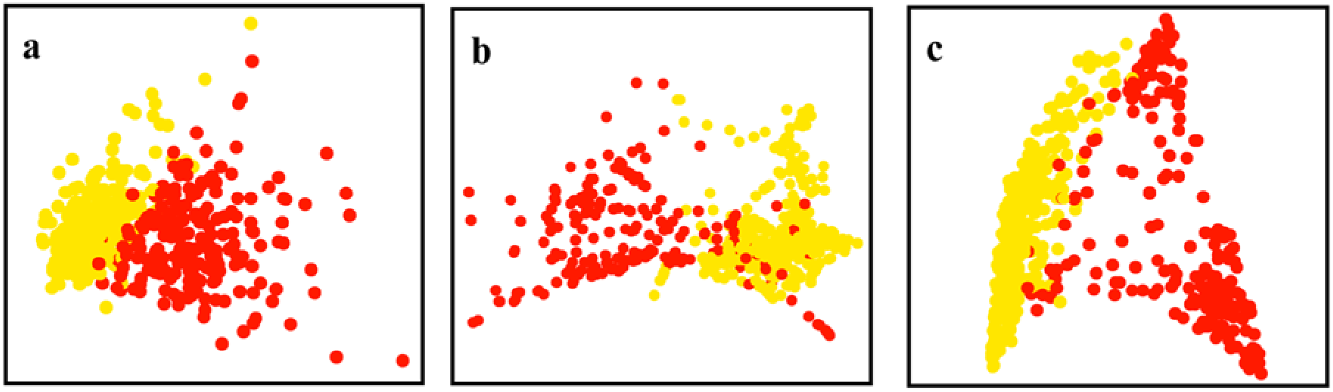

Figure 5.
Two-dimensional projections of the breast cancer data. (a) MDS. (b) ISOMAP. (c) LE.

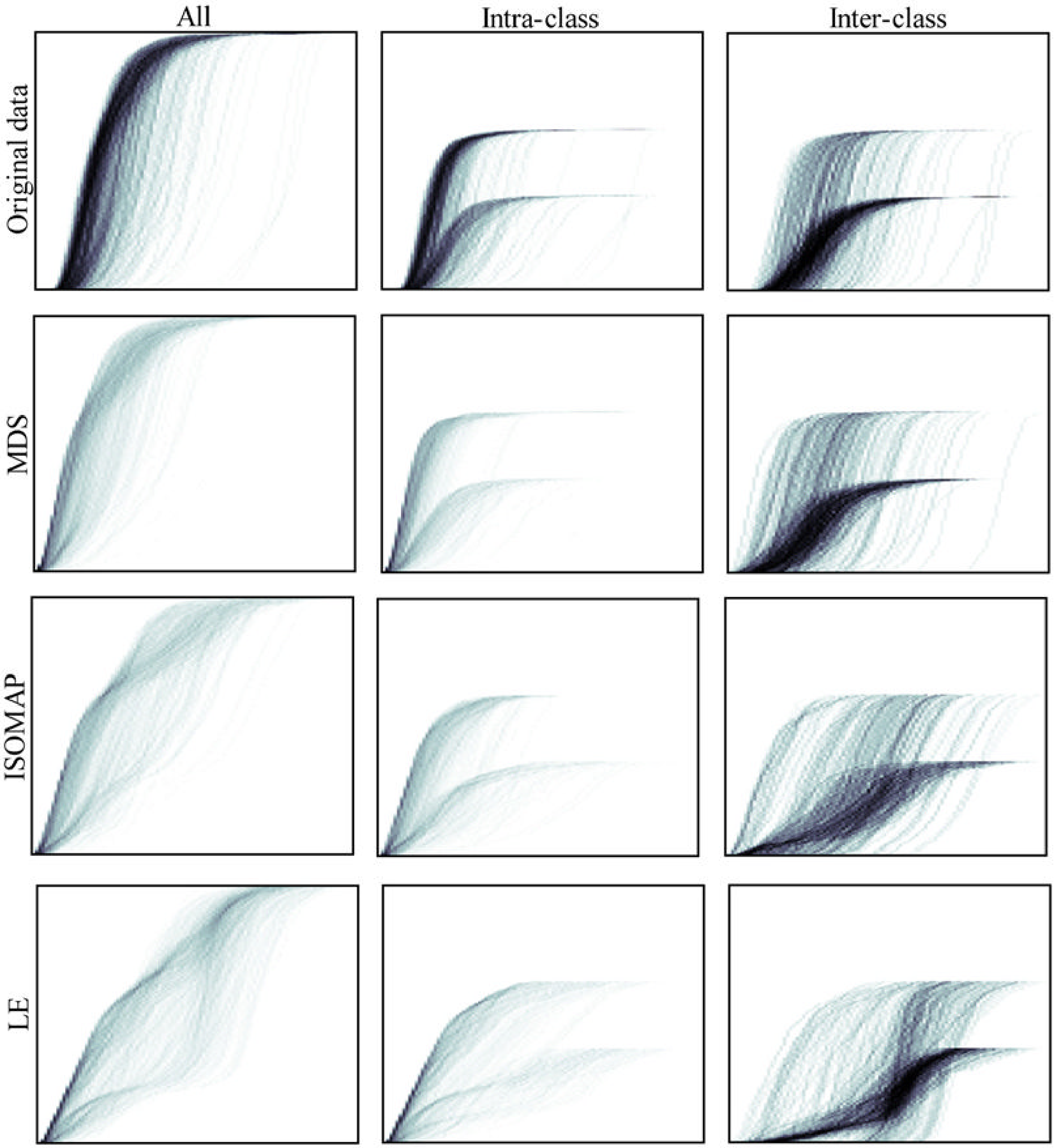

Figure 6.
K-epsilon diagrams of the breast cancer data and its projections.

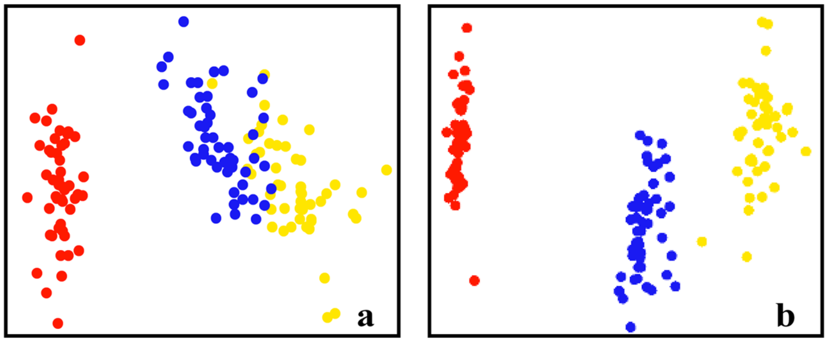

Figure 7.
(a) MDS projection of iris data in the original space. (b) MDS projection of iris data in the force feature space.

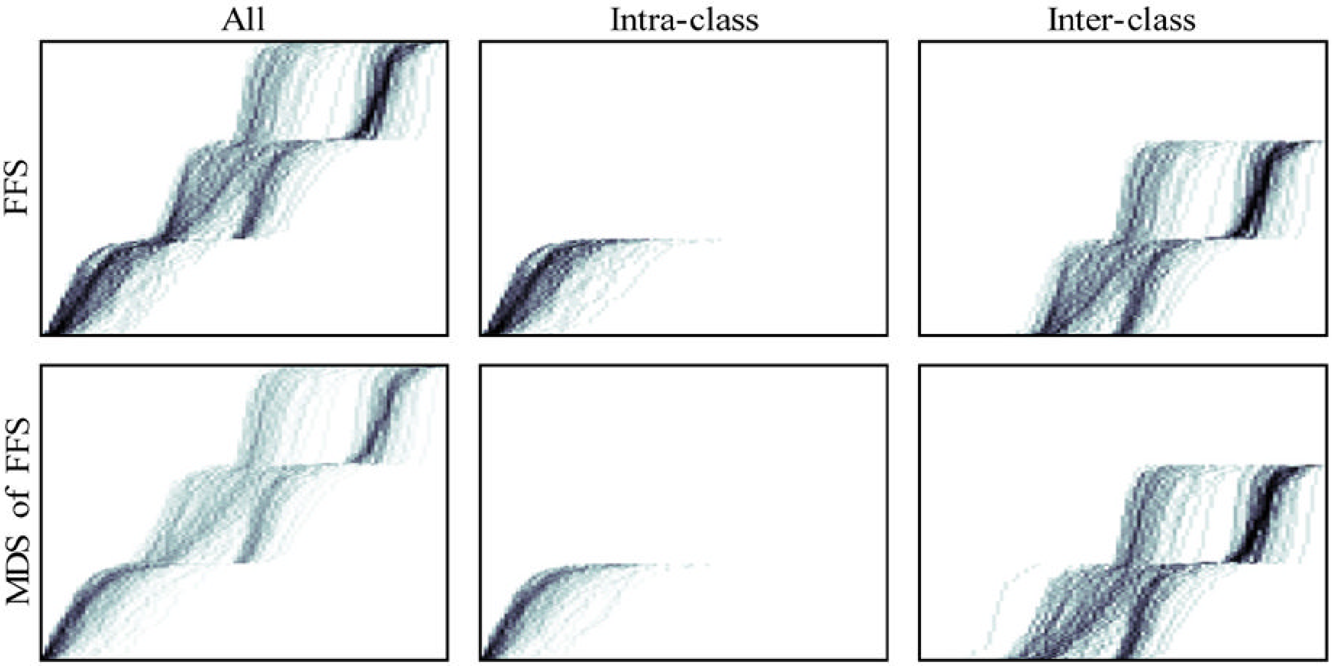

Figure 8.

K-epsilon diagrams of the FFS of the iris dataset.

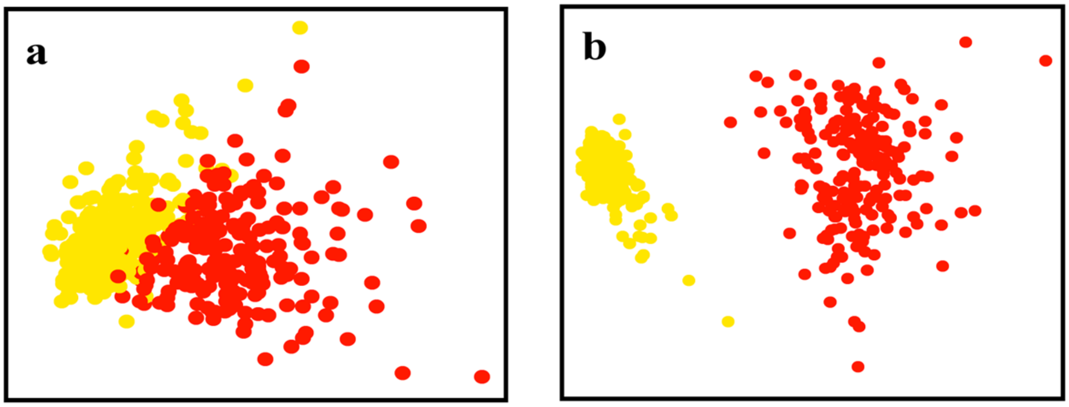

Figure 9.
MDS projection of the breast cancer data. (a) Original data. (b) Force feature space data.

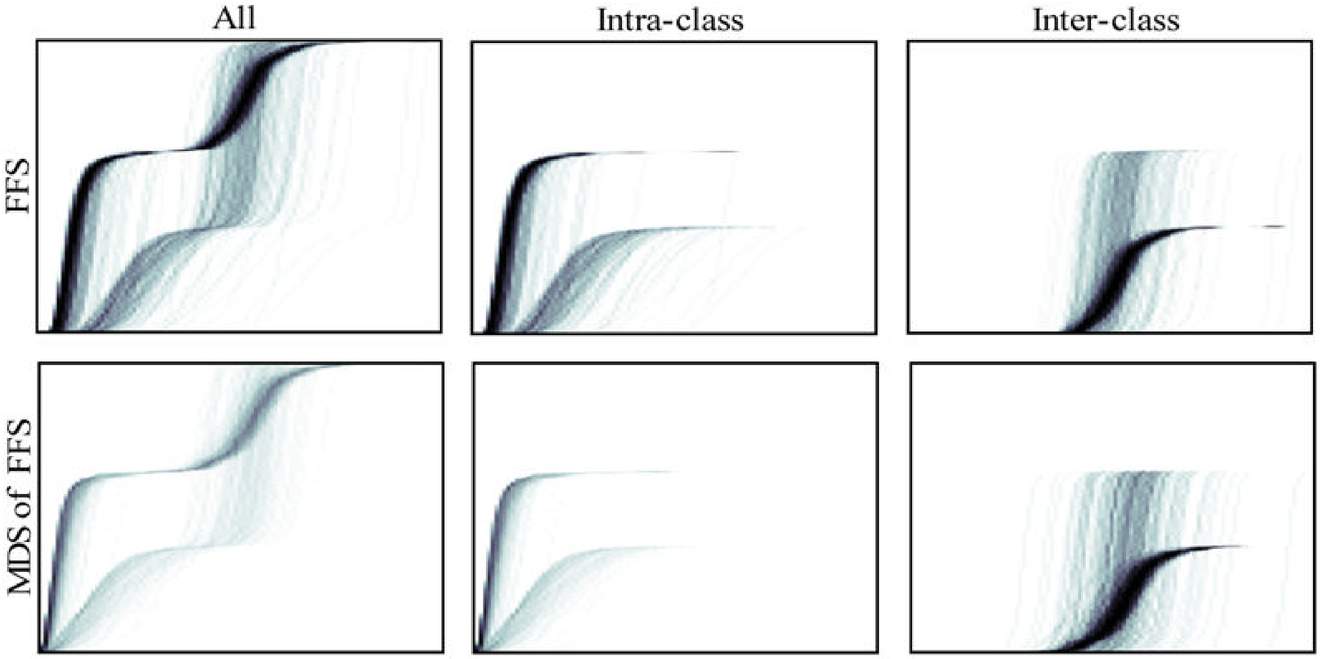

Figure 10.
K-epsilon diagrams of the force feature space of breast cancer data and its MDS projection.

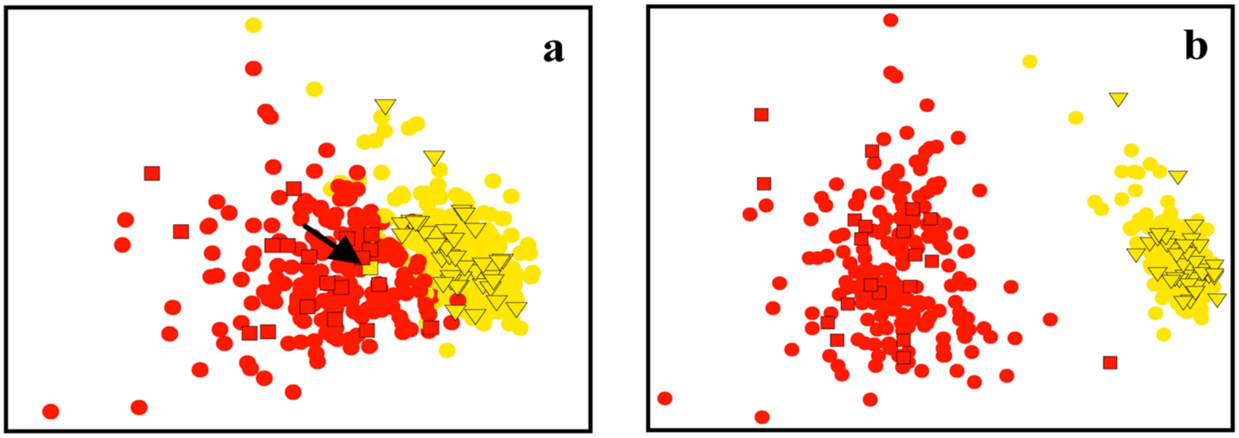

Figure 11.
MDS projection of the breast cancer dataset (a) in the original space, and (b) in the force feature space. Classifier output is denoted by shape, ground truth is denoted by color.

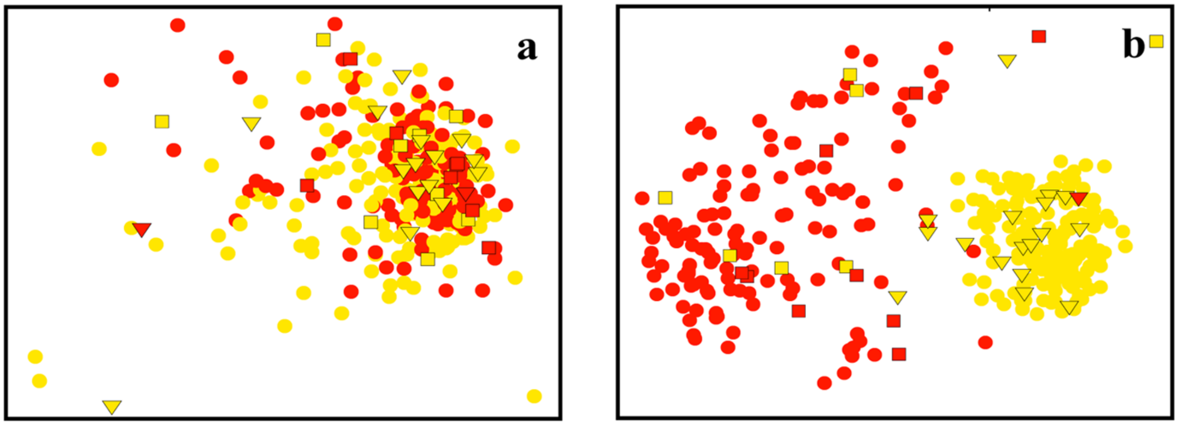

Figure 12.
MDS projection of the liver dataset with KNN (a) in the original space, and (b) in the force feature space.

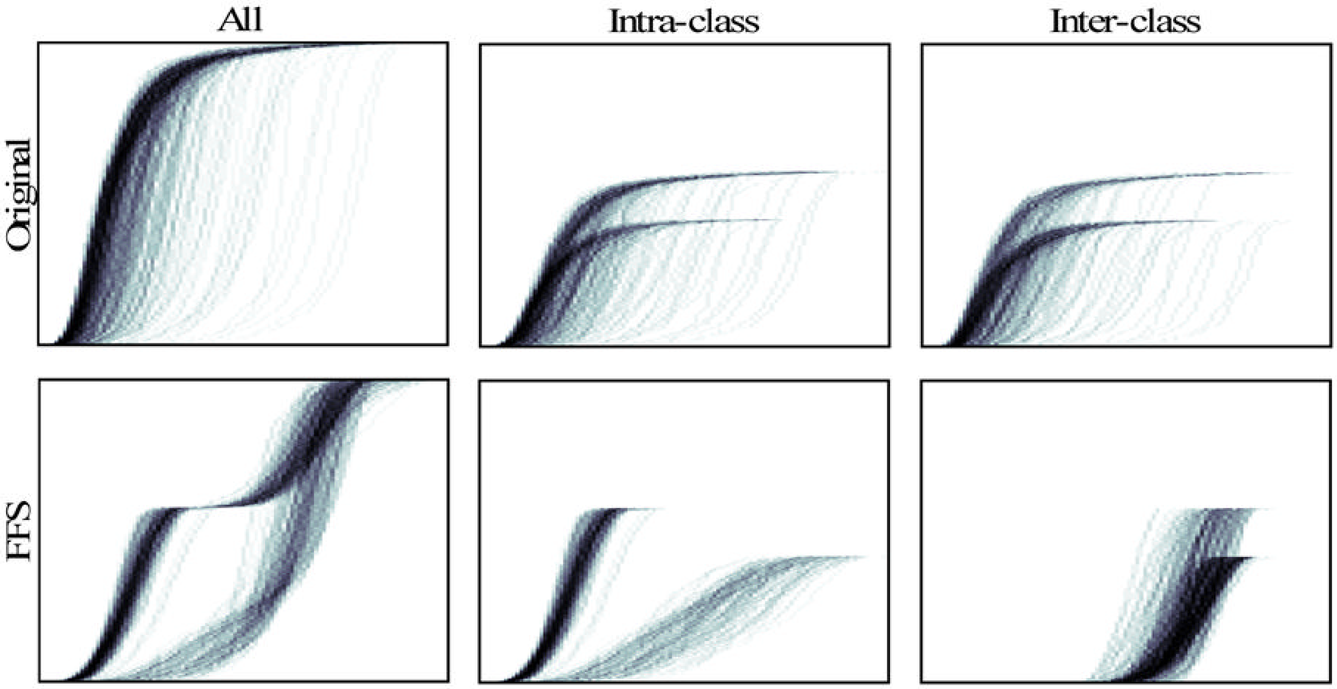

Figure 13.

K-epsilon diagrams of the liver dataset. Top: original. Bottom: lifted to force feature space.

Table 1

Parameter description

	Parameters						
	α_R	λ_A	λ_R	Δ	k	β	γ
Breast	0.7	3	3	0.12	8	1	3
Cleveland	0.7	3	3	0.14	8	1	3
Echo	0.7	3	3	0.22	8	1	3
Ionosphere	0.7	2	5	0.185	8	1	3
Liver	0.7	1	4	0.22	8	1	2.5
Monks1	0.25	2.2	3.5	0.11	3	0.5	0.3
Monks2	0.55	2.5	3	0.99	1	0.7	1
Monks3	0.2	2.5	3	0.1	12	0.3	0.3

Table 2

Comparison of classifier correct rate

	KNN		SVM	
	Org.	**FFS**	**Org.**	**FFS**
Breast	95.2	**97.3**	**97.2**	**97.2**
Cleveland	75.5	**81.7**	82.7	**83.0**
Echo	88.9	**92.1**	**97.2**	96.0
Ionosphere	86.6	**93.8**	87.6	**90.5**
Liver	62.6	**66.4**	**69.0**	67.7
Monks1	27.0	**40.3**	**42.6**	35.9
Monks2	62.6	**63.9**	49.8	**62.9**
Monks3	14.7	**38.6**	**44.2**	38.2