Feature Reduction and Selection

Selim Aksoy

Department of Computer Engineering
Bilkent University
saksoy@cs.bilkent.edu.tr

CS 551, Fall 2015

Introduction

- In practical multicategory applications, it is not unusual to encounter problems involving tens or hundreds of features.
- Intuitively, it may seem that each feature is useful for at least some of the discriminations.
- In general, if the performance obtained with a given set of features is inadequate, it is natural to consider adding new features.
- Even though increasing the number of features increases the complexity of the classifier, it may be acceptable for an improved performance.

Introduction

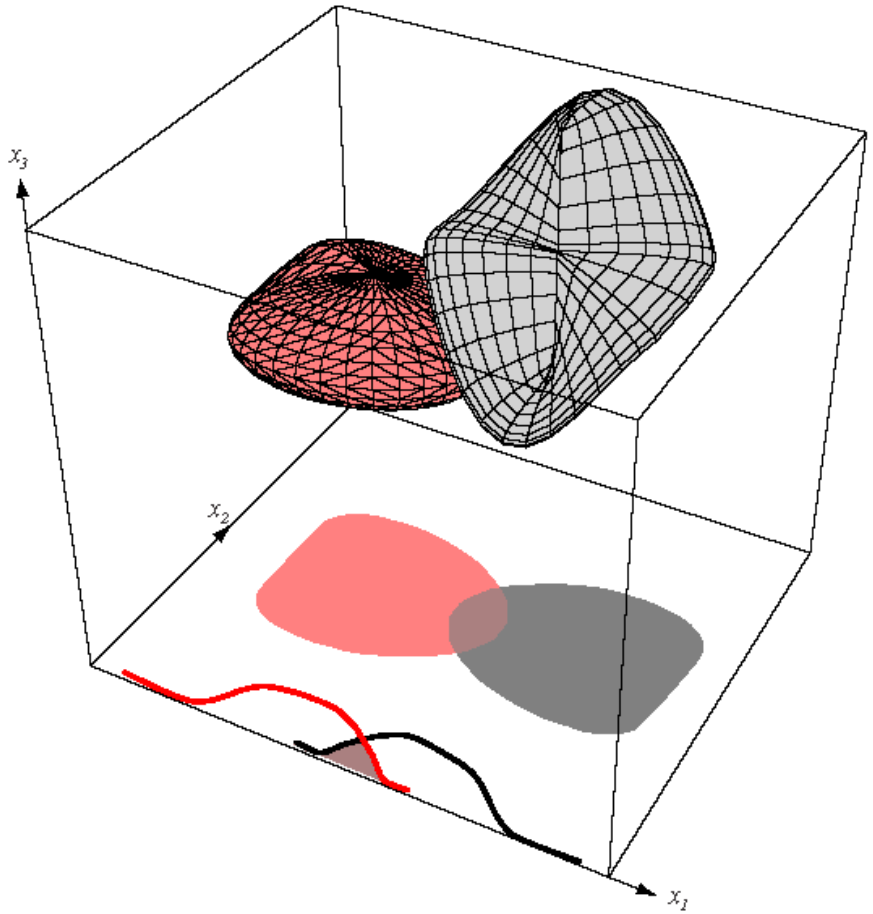


Figure 1: There is a non-zero Bayes error in the one-dimensional x_1 space or the two-dimensional x_1, x_2 space. However, the Bayes error vanishes in the x_1, x_2, x_3 space because of non-overlapping densities.

Problems of Dimensionality

- Unfortunately, it has frequently been observed in practice that, beyond a certain point, adding new features leads to worse rather than better performance.
- This is called the *curse of dimensionality*.
- There are two issues that we must be careful about:
  - How is the classification accuracy affected by the dimensionality (relative to the amount of training data)?
  - How is the complexity of the classifier affected by the dimensionality?

Problems of Dimensionality

- Potential reasons for increase in error include
  - wrong assumptions in model selection,
  - estimation errors due to the finite number of training samples for high-dimensional observations (overfitting).
- Potential solutions include
  - reducing the dimensionality,
  - simplifying the estimation.

Problems of Dimensionality

- Dimensionality can be reduced by
  - redesigning the features,
  - selecting an appropriate subset among the existing features,
  - combining existing features.
- Estimation errors can be simplified by
  - assuming equal covariance for all classes (for the Gaussian case),
  - using regularization,
  - using prior information and a Bayes estimate,
  - using heuristics such as conditional independence,
  - $\ldots$.

Problems of Dimensionality

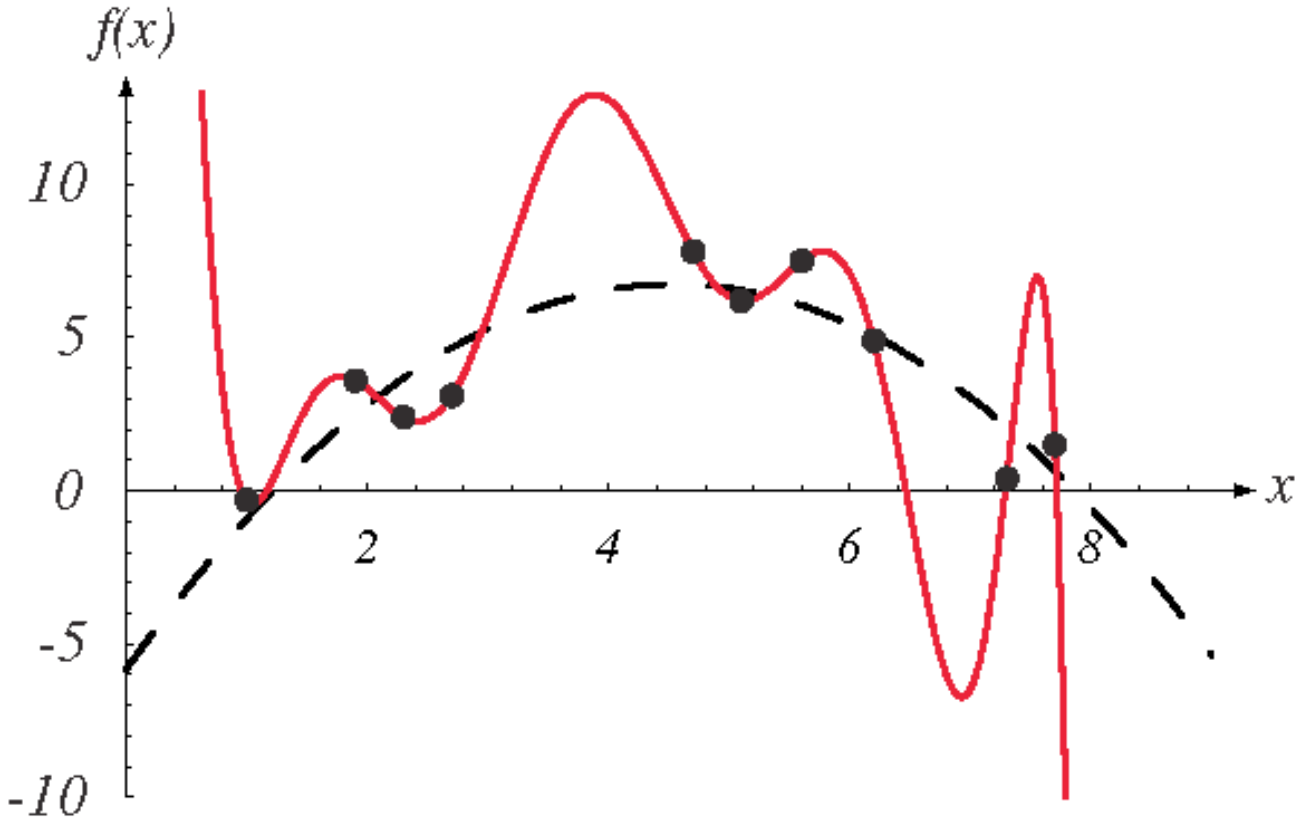


Figure 2: Problem of insufficient data is analogous to problems in curve fitting. The training data (black dots) are selected from a quadratic function plus Gaussian noise. A tenth-degree polynomial fits the data perfectly but we prefer a second-order polynomial for better generalization.

Problems of Dimensionality

- All of the commonly used classifiers can suffer from the curse of dimensionality.
- While an exact relationship between the probability of error, the number of training samples, the number of features, and the number of parameters is very difficult to establish, some guidelines have been suggested.
- It is generally accepted that using at least ten times as many training samples per class as the number of features ($n/d > 10$) is a good practice.
- The more complex the classifier, the larger should the ratio of sample size to dimensionality be.

Feature Reduction

- One way of coping with the problem of high dimensionality is to reduce the dimensionality by combining features.
- Issues in feature reduction:
  - Linear vs. non-linear transformations.
  - Use of class labels or not (depends on the availability of training data).
  - Training objective:
    - minimizing classification error (discriminative training),
    - minimizing reconstruction error (PCA),
    - maximizing class separability (LDA),
    - retaining interesting directions (projection pursuit),
    - making features as independent as possible (ICA),
    - embedding to lower dimensional manifolds (Isomap, LLE)

Feature Reduction

- Linear combinations are particularly attractive because they are simple to compute and are analytically tractable.
- Linear methods project the high-dimensional data onto a lower dimensional space.
- Advantages of these projections include
  - reduced complexity in estimation and classification,
  - ability to visually examine the multivariate data in two or three dimensions.

Feature Reduction

- Given $\mathbf{x} \in \mathbb{R}^d$, the goal is to find a linear transformation $\mathbf{A}$ that gives $\mathbf{y} = \mathbf{A}^T\mathbf{x} \in \mathbb{R}^{d'}$ where $d' < d$.
- Two classical approaches for finding optimal linear transformations are:
  - *Principal Components Analysis (PCA):* Seeks a projection that best represents the data in a least-squares sense.
  - *Linear Discriminant Analysis (LDA):* Seeks a projection that best separates the data in a least-squares sense.

Principal Components Analysis

- Given $\mathbf{x_1}, \ldots, \mathbf{x_n} \in \mathbb{R}^d$, the goal is to find a d'-dimensional subspace where the reconstruction error of $\mathbf{x_i}$ in this subspace is minimized.
- The criterion function for the reconstruction error can be defined in the least-squares sense as

$$J_{d'} = \sum_{i=1}^{n} \left\| \sum_{k=1}^{d'} \mathbf{y}_{ik} \mathbf{e_k} - \mathbf{x_i} \right\|^2$$

where $\mathbf{e_1}, \ldots, \mathbf{e}_{d'}$ are the bases for the subspace (stored as the columns of $\mathbf{A}$) and $\mathbf{y}_i$ is the projection of $\mathbf{x_i}$ onto that subspace.

Principal Components Analysis

- It can be shown that $J_{d'}$ is minimized when $\mathbf{e_1}, \ldots, \mathbf{e_{d'}}$ are the d' eigenvectors of the *scatter matrix*

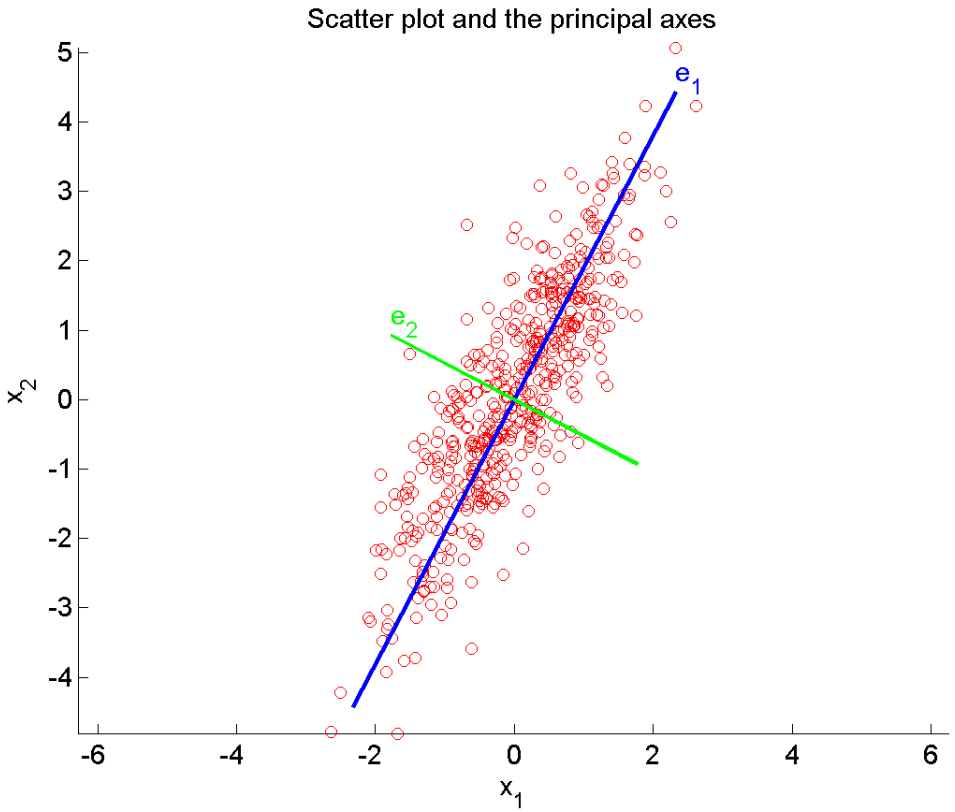
$$S = \sum_{i=1}^{n} (\mathbf{x_i} - \boldsymbol{\mu})(\mathbf{x_i} - \boldsymbol{\mu})^T$$

  having the largest eigenvalues.
- The coefficients $\mathbf{y} = (\mathbf{y}_i, \ldots, \mathbf{y}_{d'})^T$ are called the *principal components*.
- When the eigenvectors are sorted in descending order of the corresponding eigenvalues, the greatest variance of the data lies on the first principal component, the second greatest variance on the second component, etc.

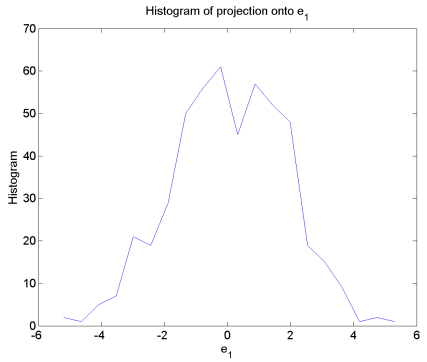
Principal Components Analysis

- Often there will be just a few large eigenvalues, and this implies that the d'-dimensional subspace contains the signal and the remaining $d - d'$ dimensions generally contain noise.
- The actual subspace where the data may lie is related to the *intrinsic dimensionality* that determines whether the given d-dimensional patterns can be described adequately in a subspace of dimensionality less than d.
- The geometric interpretation of intrinsic dimensionality is that the entire data set lies on a topological d'-dimensional hypersurface.
- Note that the intrinsic dimensionality is not the same as the linear dimensionality which is related to the number of significant eigenvalues of the scatter matrix of the data.

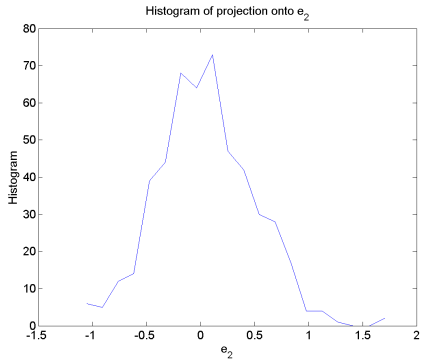
Examples



(a) Scatter plot.



(b) Projection onto e_1.



(c) Projection onto e_2.

Figure 3: Scatter plot (red dots) and the principal axes for a bivariate sample. The blue line shows the axis e_1 with the greatest variance and the green line shows the axis e_2 with the smallest variance. Features are now uncorrelated.

Examples

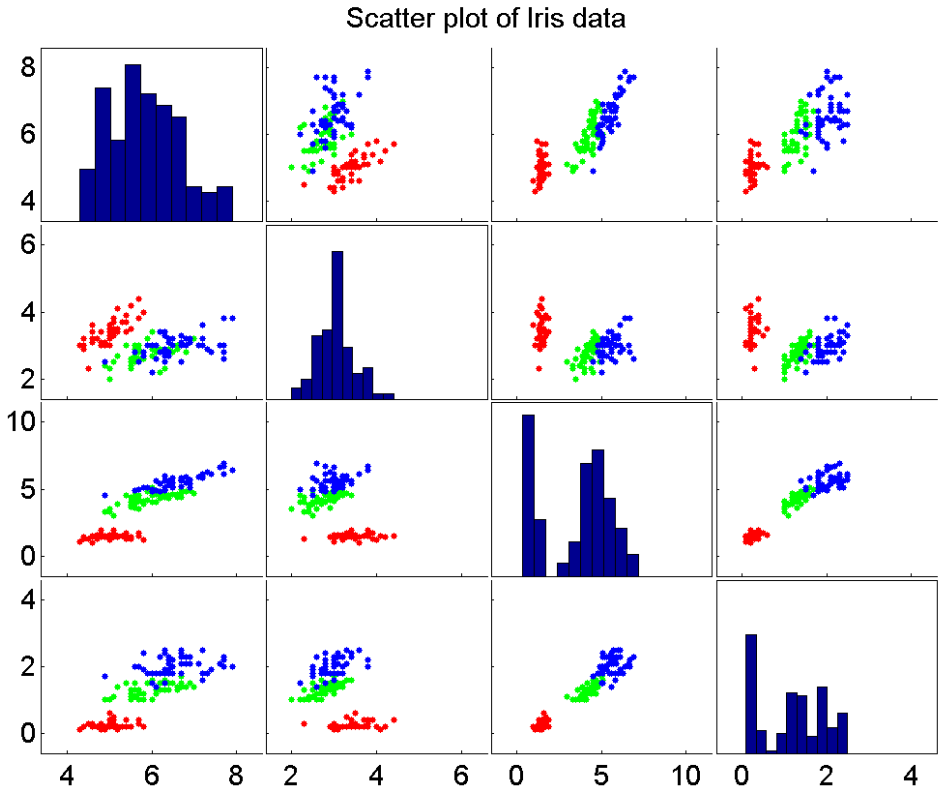


Figure 4: Scatter plot of the iris data. Diagonal cells show the histogram for each feature. Other cells show scatters of pairs of features x_1, x_2, x_3, x_4 in top-down and left-right order. Red, green and blue points represent samples for the setosa, versicolor and virginica classes, respectively.

Examples

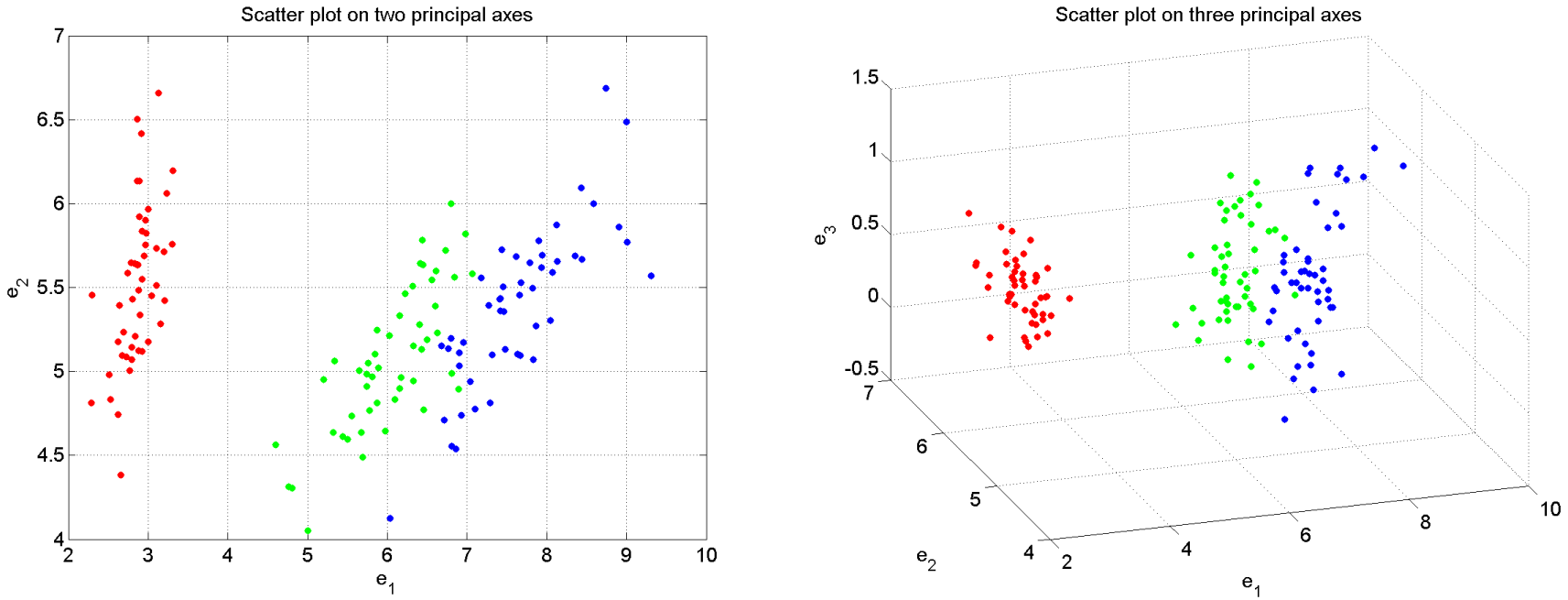


Figure 5: Scatter plot of the projection of the iris data onto the first two and the first three principal axes. Red, green and blue points represent samples for the setosa, versicolor and virginica classes, respectively.

Linear Discriminant Analysis

- Whereas PCA seeks directions that are efficient for representation, discriminant analysis seeks directions that are efficient for discrimination.
- Given $\mathbf{x_1}, \ldots, \mathbf{x_n} \in \mathbb{R}^d$ divided into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$ corresponding to the classes w_1 and w_2, respectively, the goal is to find a projection onto a line defined as

$$y = \mathbf{w}^T \mathbf{x}$$

where the points corresponding to $\mathcal{D}_1$ and $\mathcal{D}_2$ are well separated.

Linear Discriminant Analysis

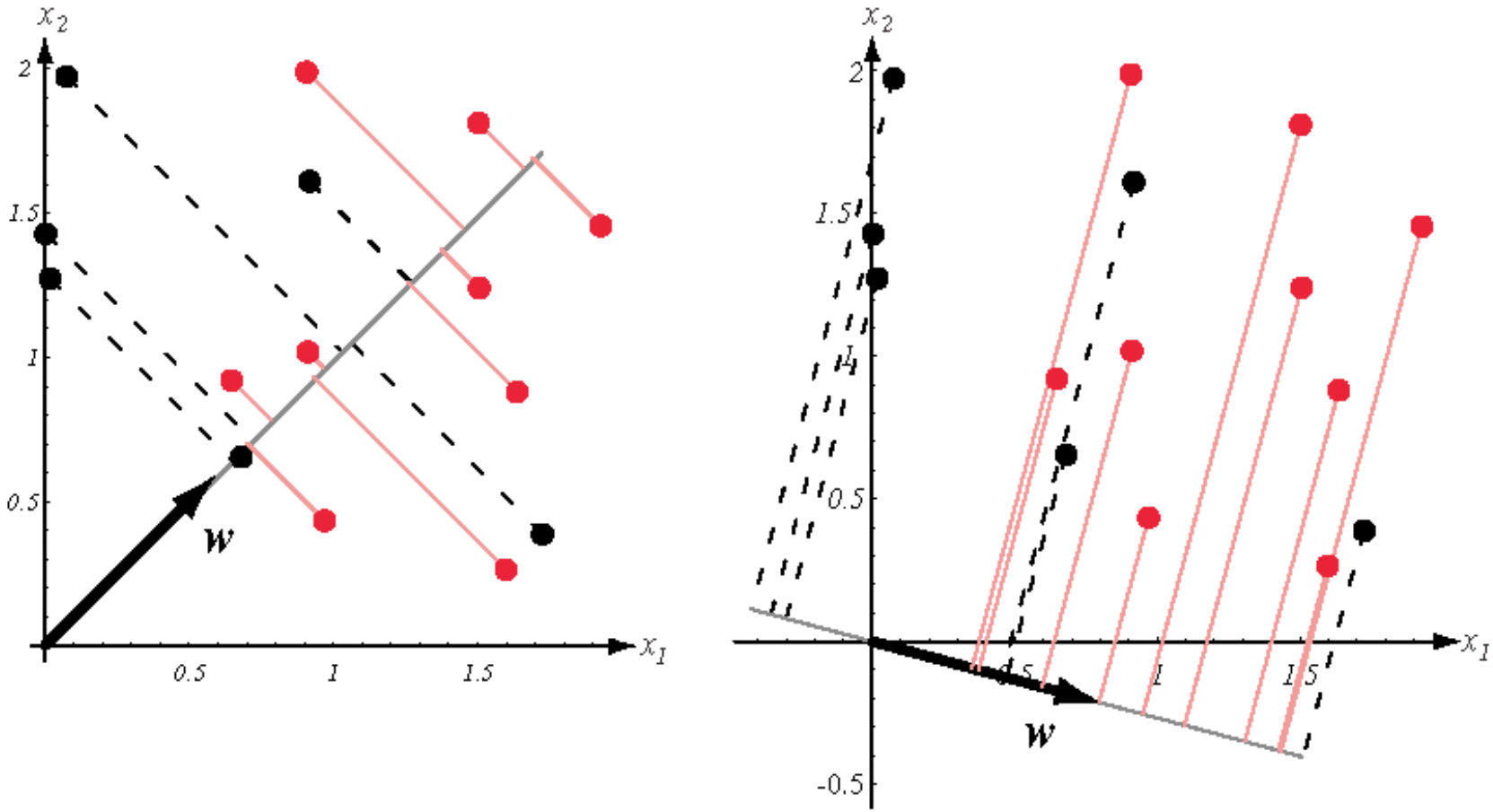


Figure 6: Projection of the same set of samples onto two different lines in the directions marked as $\mathbf{w}$. The figure on the right shows greater separation between the red and black projected points.

Linear Discriminant Analysis

- The criterion function for the best separation can be defined as

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

  where $\tilde{m}_i = \frac{1}{\#\mathcal{D}_i} \sum_{y \in w_i} y$ is the sample mean and $\tilde{s}_i^2 = \sum_{y \in w_i} (y - \tilde{m}_i)^2$ is the scatter for the projected samples labeled w_i.

- This is called the *Fisher's linear discriminant* with the geometric interpretation that the best projection makes the difference between the means as large as possible relative to the variance.

Linear Discriminant Analysis

- To compute the optimal $\mathbf{w}$, we define the *scatter matrices* $\mathbf{S}_i$

$$\mathbf{S}_i = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T \text{ where } \mathbf{m}_i = \frac{1}{\#\mathcal{D}_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x},$$

the *within-class scatter matrix* $\mathbf{S_W}$

$$\mathbf{S_W} = \mathbf{S_1} + \mathbf{S_2},$$

and the *between-class scatter matrix* $\mathbf{S_B}$

$$\mathbf{S_B} = (\mathbf{m_1} - \mathbf{m_2})(\mathbf{m_1} - \mathbf{m_2})^T.$$

Linear Discriminant Analysis

- Then, the criterion function becomes

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S_B} \mathbf{w}}{\mathbf{w}^T \mathbf{S_W} \mathbf{w}}$$

and the optimal $\mathbf{w}$ can be computed as

$$\mathbf{w} = \mathbf{S_W^{-1}}(\mathbf{m_1} - \mathbf{m_2}).$$

- Note that, $\mathbf{S_W}$ is symmetric and positive semidefinite, and it is usually nonsingular if $n > d$. $\mathbf{S_B}$ is also symmetric and positive semidefinite, but its rank is at most 1.

Linear Discriminant Analysis

- Generalization to c classes involves $c-1$ discriminant functions where the projection is from a d-dimensional space to a $(c-1)$-dimensional space ($d \geq c$).
- The scatter matrices $\mathbf{S}_i$ are computed as

$$\mathbf{S}_i = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T \text{ where } \mathbf{m}_i = \frac{1}{\#\mathcal{D}_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}.$$

- The within-class scatter matrix $\mathbf{S}_\mathbf{W}$ is computed as

$$\mathbf{S}_\mathbf{W} = \sum_{i=1}^{c} \mathbf{S}_i.$$

Linear Discriminant Analysis

- The between-class scatter matrix $\mathbf{S_B}$ is computed as

$$\mathbf{S_B} = \sum_{i=1}^{c} (\#\mathcal{D}_i)(\mathbf{m_i} - \mathbf{m})(\mathbf{m_i} - \mathbf{m})^T$$

  where $\mathbf{m} = \frac{1}{n} \sum_{\mathbf{x}} \mathbf{x}$ is the total mean vector.

- Then, the criterion function becomes

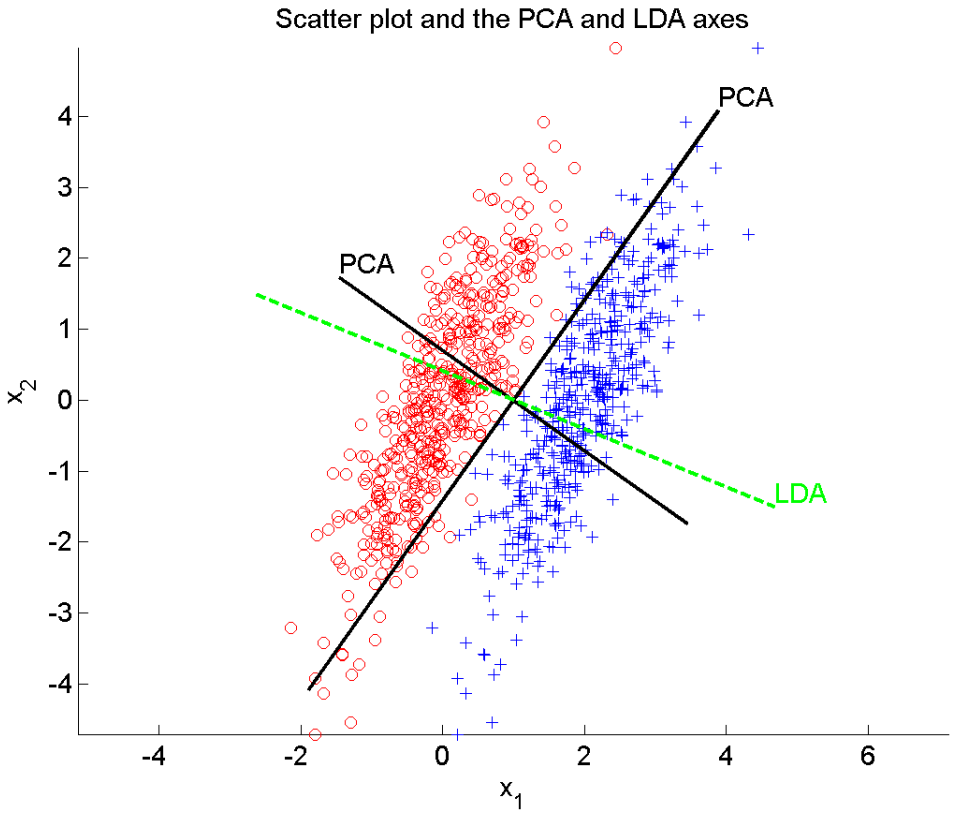
$$J(\mathbf{W}) = \frac{|\mathbf{W}^T \mathbf{S_B} \mathbf{W}|}{|\mathbf{W}^T \mathbf{S_W} \mathbf{W}|}$$

  where $\mathbf{W}$ is the d-by-$(c-1)$ transformation matrix and $|\cdot|$ represents the determinant.

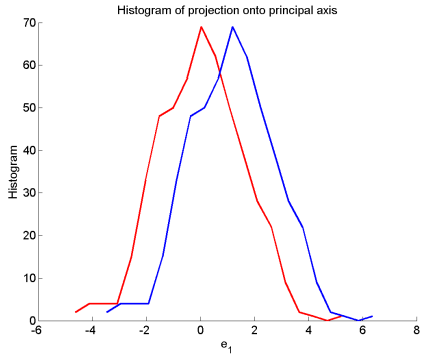
Linear Discriminant Analysis

- It can be shown that $J(\mathbf{W})$ is maximized when the columns of $\mathbf{W}$ are the eigenvectors of $\mathbf{S}_\mathbf{W}^{-1}\mathbf{S}_\mathbf{B}$ having the largest eigenvalues.
- Because $\mathbf{S}_\mathbf{B}$ is the sum of c matrices of rank one or less, and because only $c-1$ of these are independent, $\mathbf{S}_\mathbf{B}$ is of rank $c-1$ or less. Thus, no more than $c-1$ of the eigenvalues are nonzero.
- Once the transformation from the d-dimensional original feature space to a lower dimensional subspace is done using PCA or LDA, parametric or non-parametric methods can be used to train Bayesian classifiers.

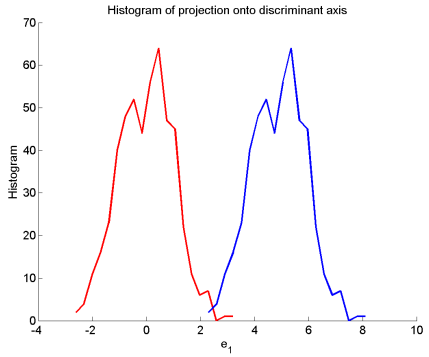
Examples



(a) Scatter plot.



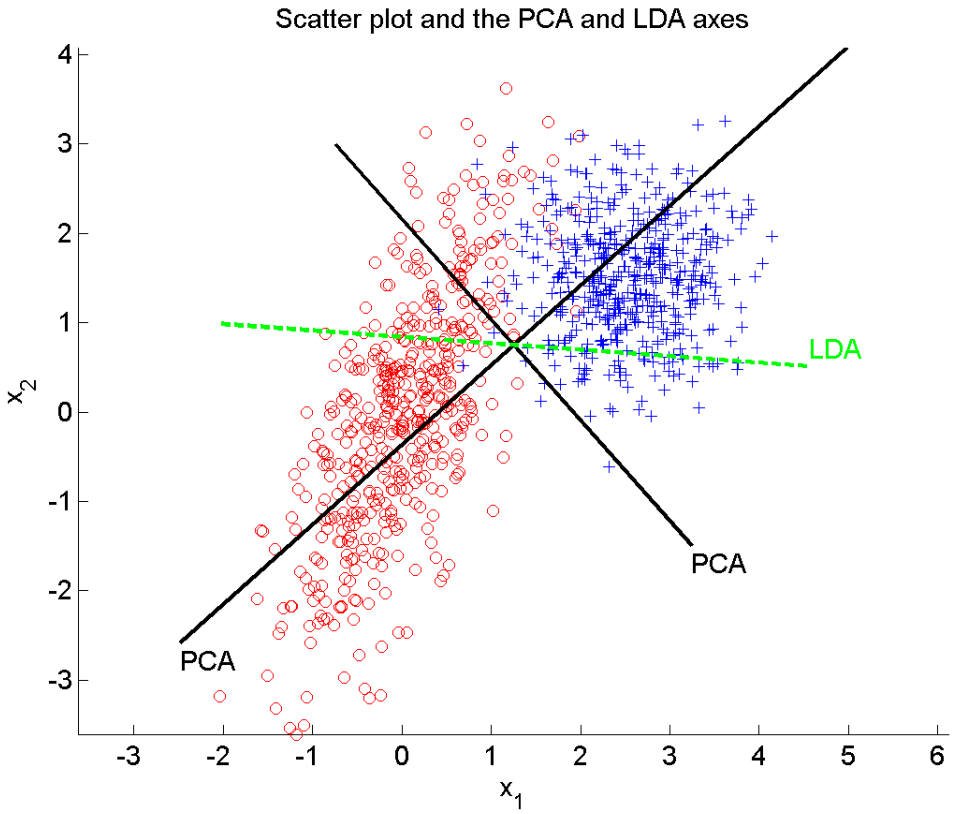
(b) Projection onto the first PCA axis.



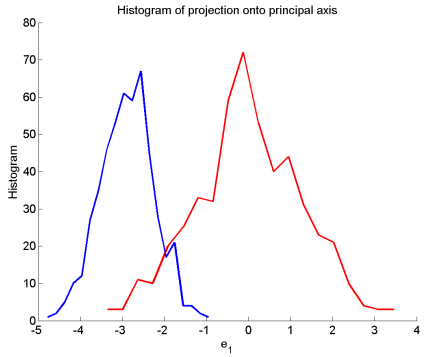
(c) Projection onto the first LDA axis.

Figure 7: Scatter plot and the PCA and LDA axes for a bivariate sample with two classes. Histogram of the projection onto the first LDA axis shows better separation than the projection onto the first PCA axis.

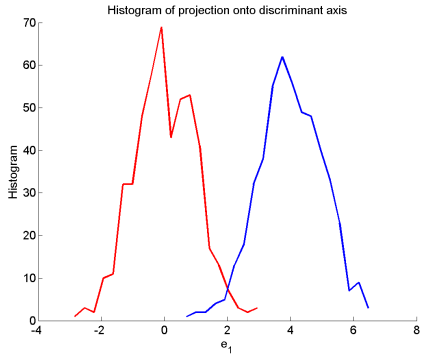
Examples



(a) Scatter plot.



(b) Projection onto the first PCA axis.



(c) Projection onto the first LDA axis.

Figure 8: Scatter plot and the PCA and LDA axes for a bivariate sample with two classes. Histogram of the projection onto the first LDA axis shows better separation than the projection onto the first PCA axis.

Examples

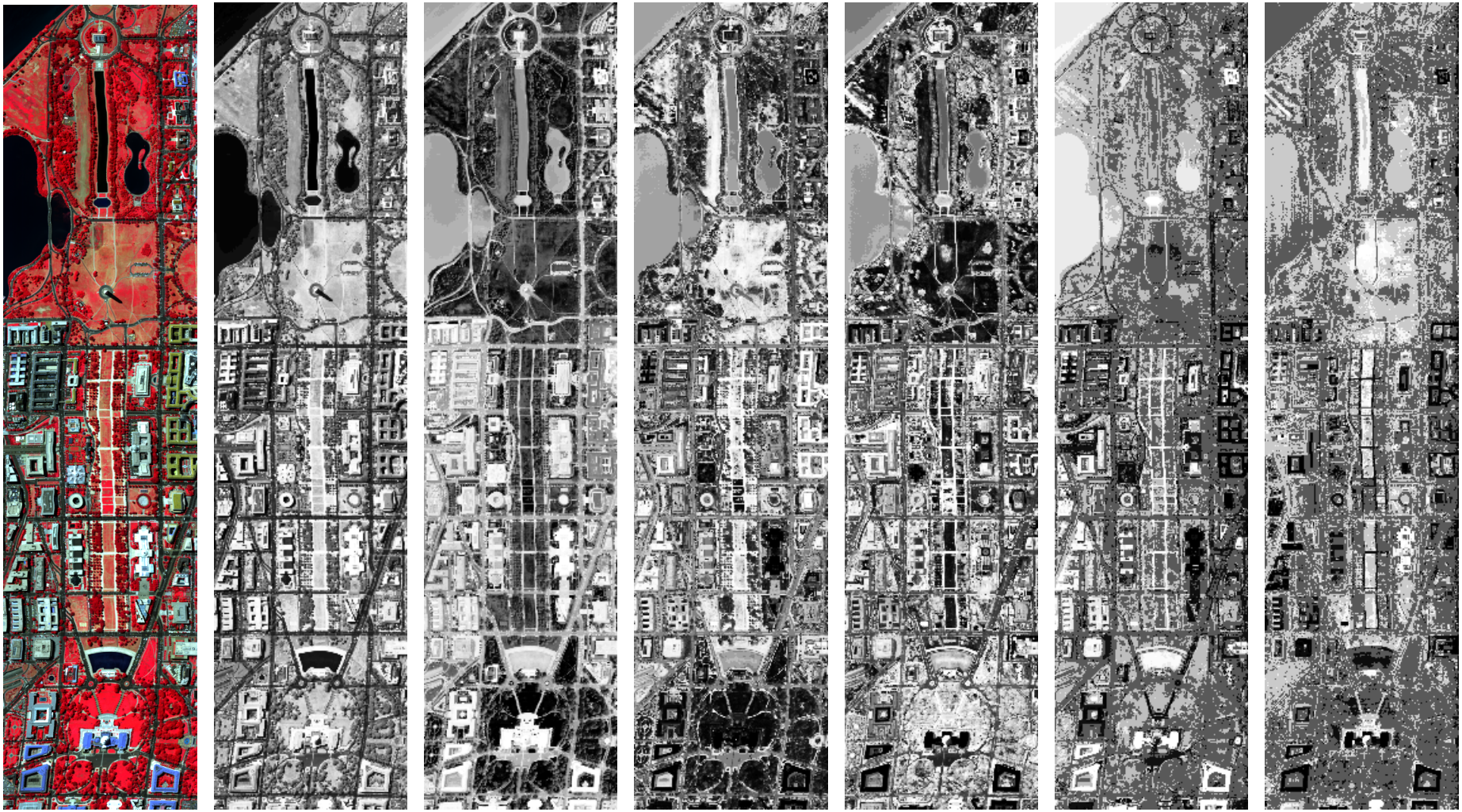


Figure 9: A satellite image and the first six PCA bands (after projection). Histogram equalization was applied to all images for better visualization.

Examples

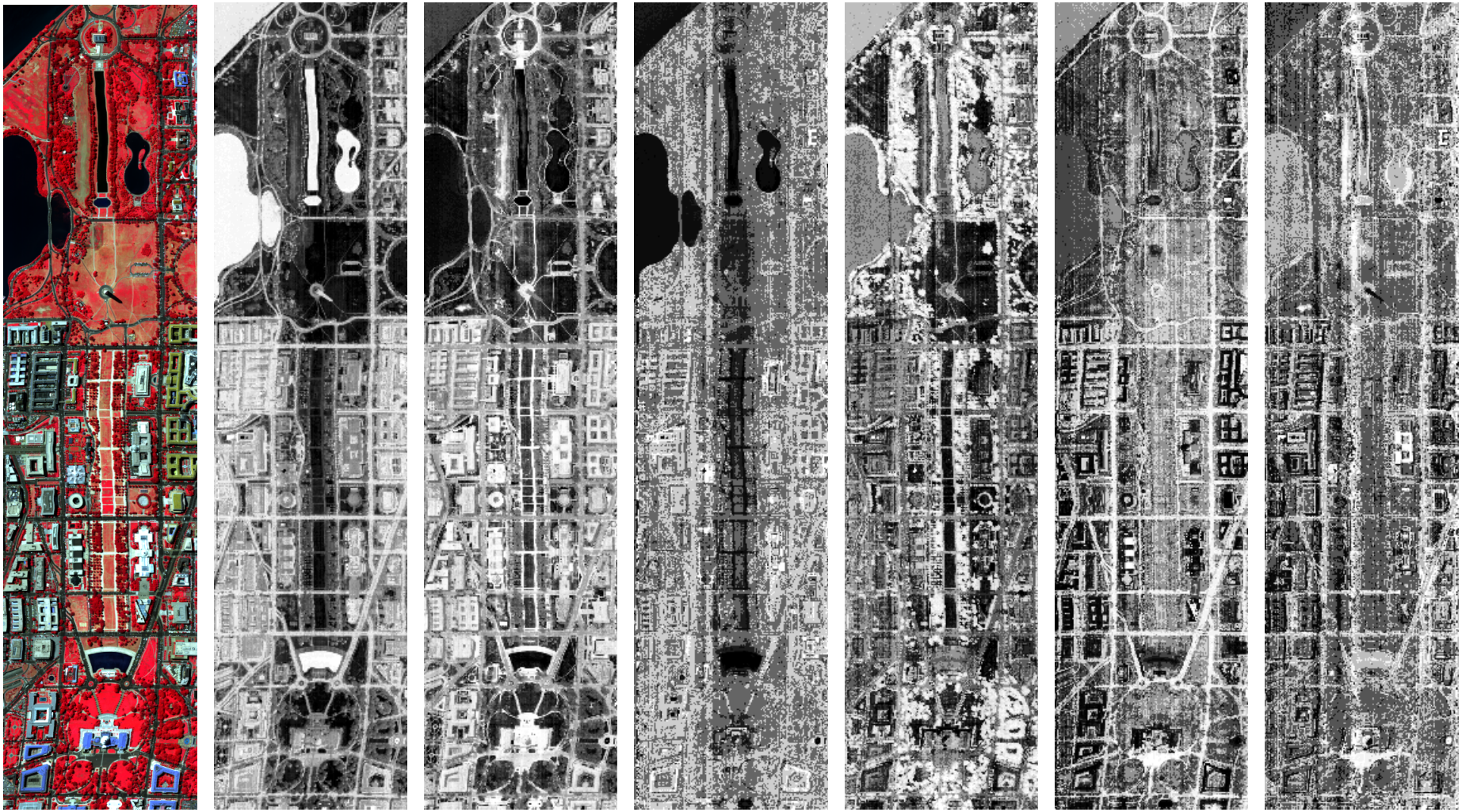


Figure 10: A satellite image and the six LDA bands (after projection). Histogram equalization was applied to all images for better visualization.

Examples

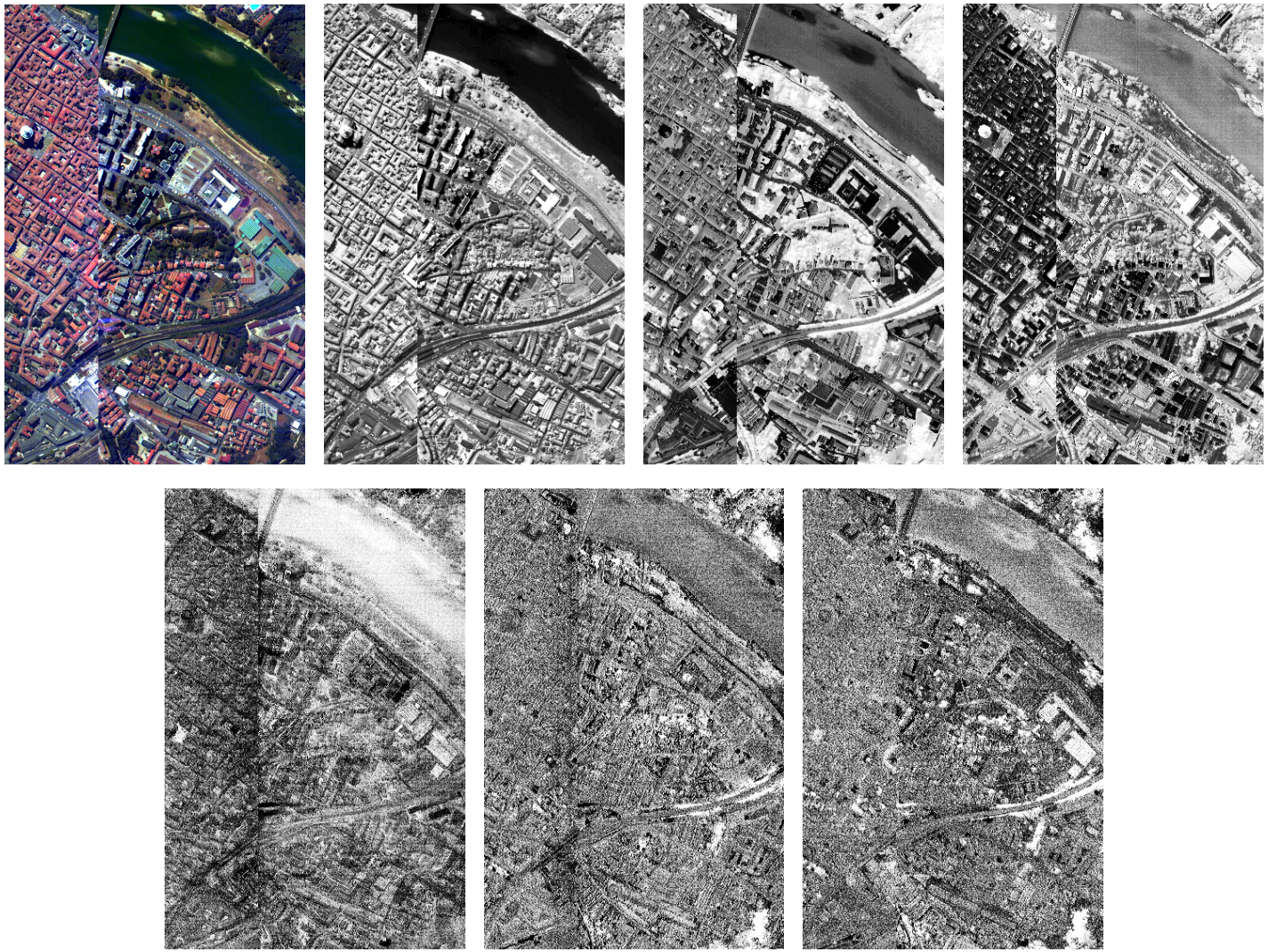


Figure 11: A satellite image and the first six PCA bands (after projection) Histogram equalization was applied to all images for better visualization.

Examples

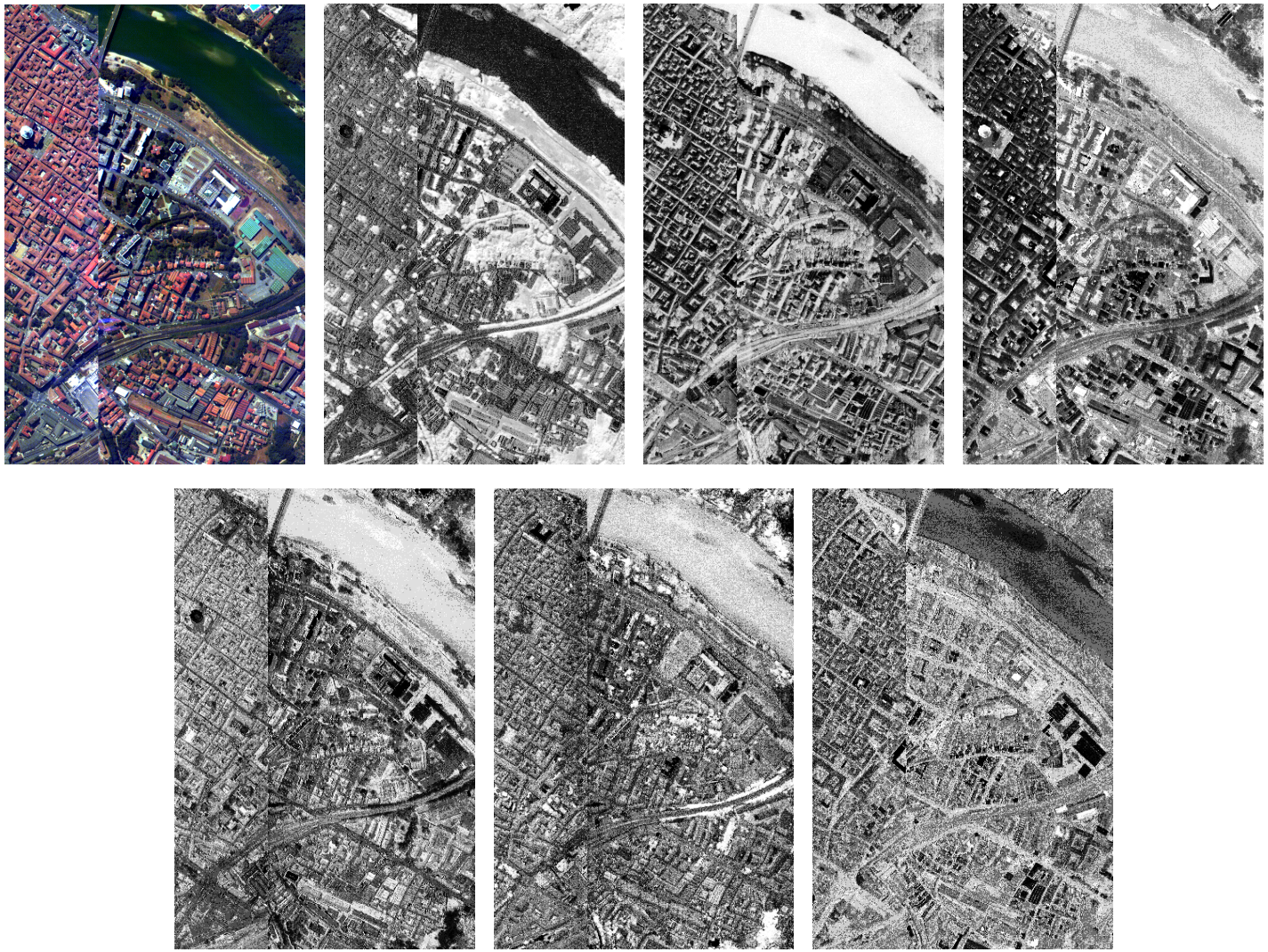


Figure 12: A satellite image and the six LDA bands (after projection). Histogram equalization was applied to all images for better visualization.

Examples

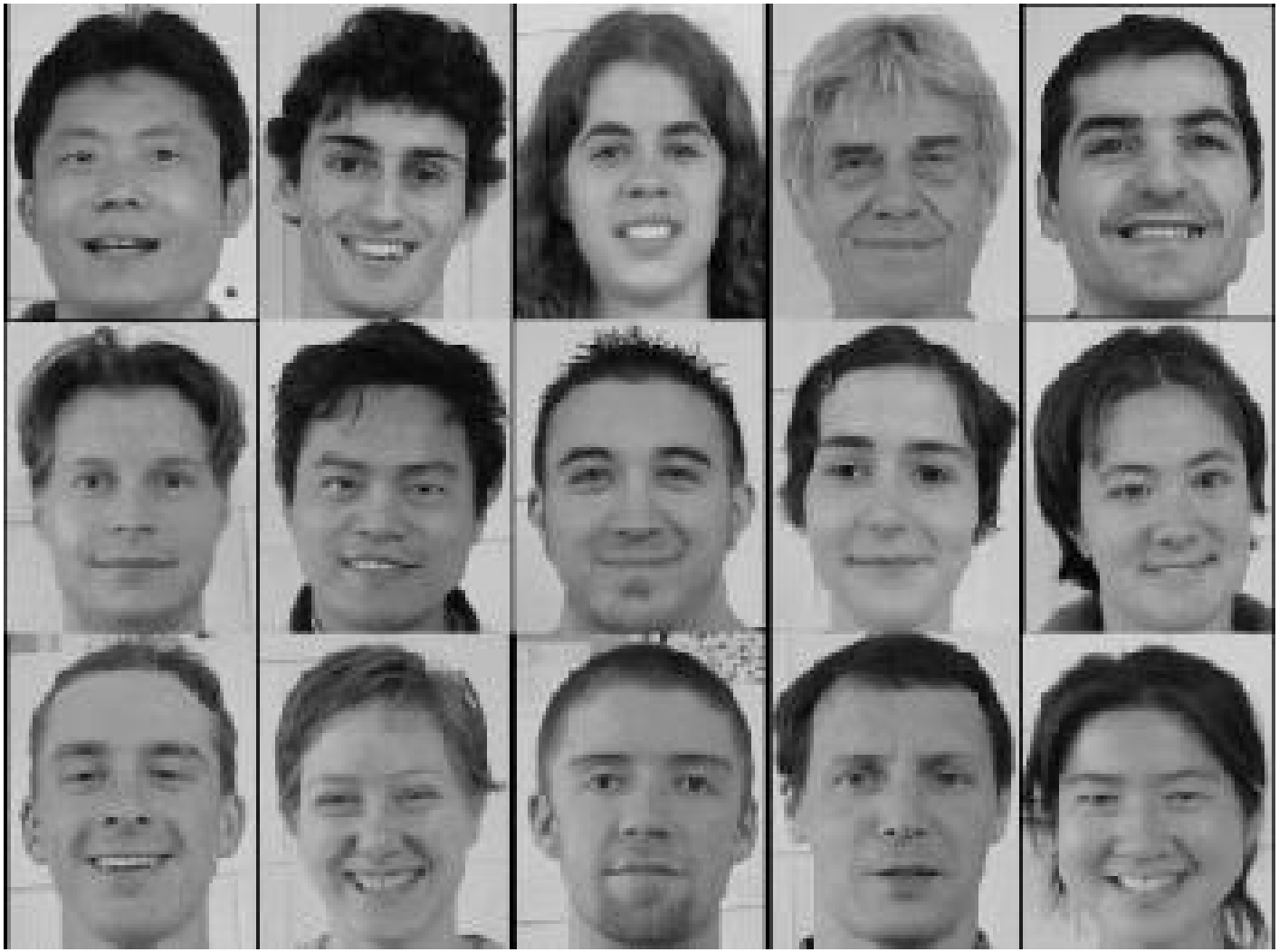


Figure 13: Example face images. (Taken from http://www.geop.ubc.ca/CDSST/eigenfaces.html.)

Examples

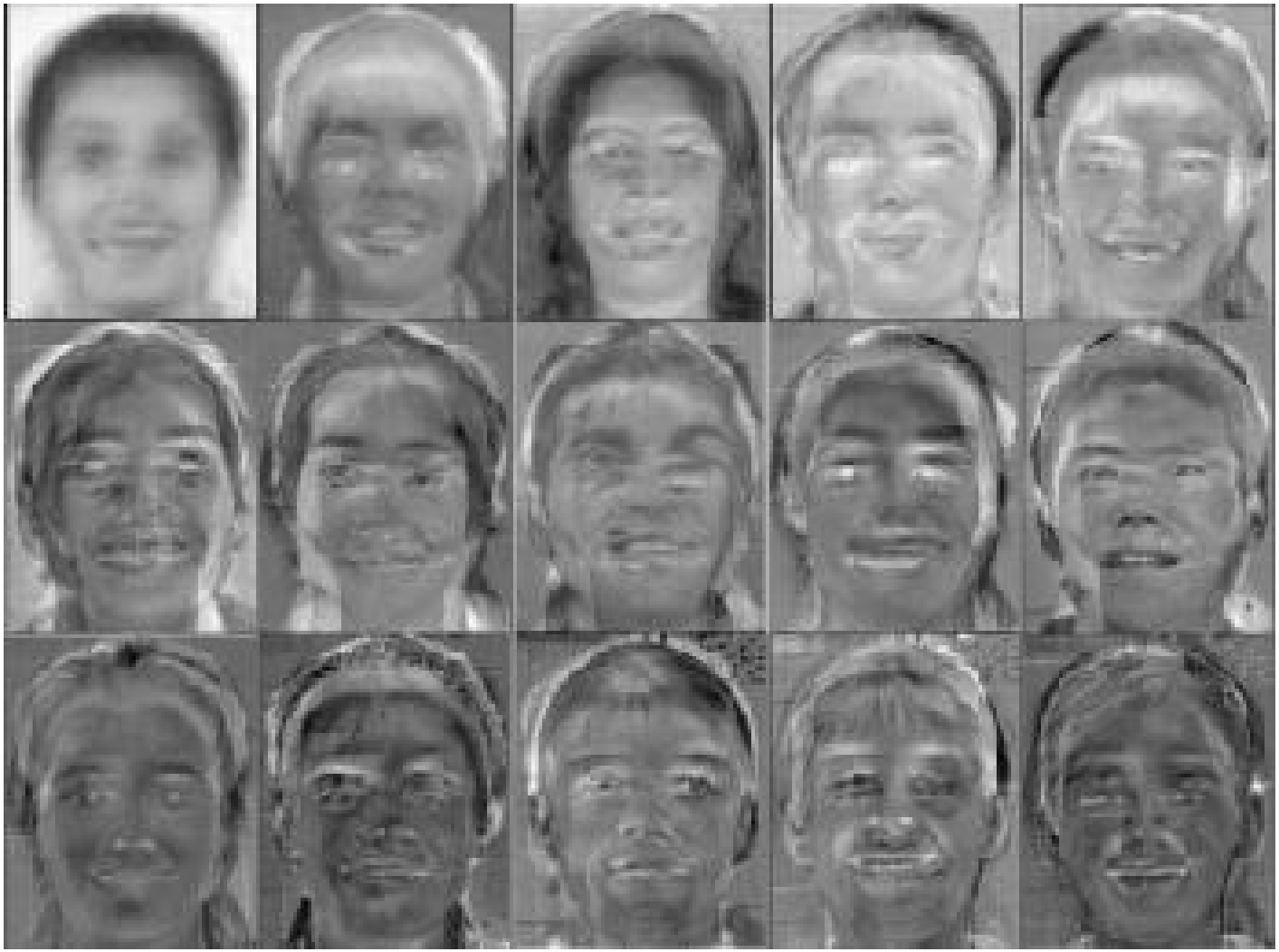


Figure 14: Eigenvectors (principal axes) of the face images (often referred to as eigenfaces).

Isometric Feature Mapping

- The isometric feature mapping (Isomap) algorithm combines the major algorithmic features of PCA and MDS (multi-dimensional scaling)
  - computational efficiency,
  - global optimality, and
  - asymptotic convergence guarantees

  with the flexibility to learn a broad class of nonlinear manifolds.
- The approach seeks to preserve the intrinsic geometry of the data, as captured in the geodesic manifold distances between all pairs of data points.

Isometric Feature Mapping

- The essential point is to estimate the geodesic distance between faraway points, given only input-space distances.
- For neighboring points, input-space distance provides a good approximation.
- For faraway points, geodesic distance can be approximated by adding up a sequence of short hops between neighboring points.
- These approximations are computed efficiently by finding shortest paths in a graph with edges connecting neighboring data points.

Isometric Feature Mapping

- The Isomap algorithm has three steps.
- The first step determines which points are neighbors on the manifold M, based on the distances $d_X(i, j)$ between pairs of points i, j in the input space X.
- A sparse graph G is defined over all data points by connecting points i and j if they are closer than ϵ (ϵ-Isomap) or if i is one of the K nearest neighbors of j (K-Isomap), and the edge weights are set as $d_X(i, j)$.

Isometric Feature Mapping

- In the second step, Isomap estimates the geodesic distances $d_M(i, j)$ between all pairs of points on the manifold M by computing their shortest path distances $d_G(i, j)$ in the graph G.
- The final step applies classical multi-dimensional scaling to the matrix of graph distances $\mathbf{D}_G = \{d_G(i, j)\}$, constructing an embedding of the data in a d-dimensional Euclidean space Y that best preserves the manifold's estimated intrinsic geometry.

Isometric Feature Mapping

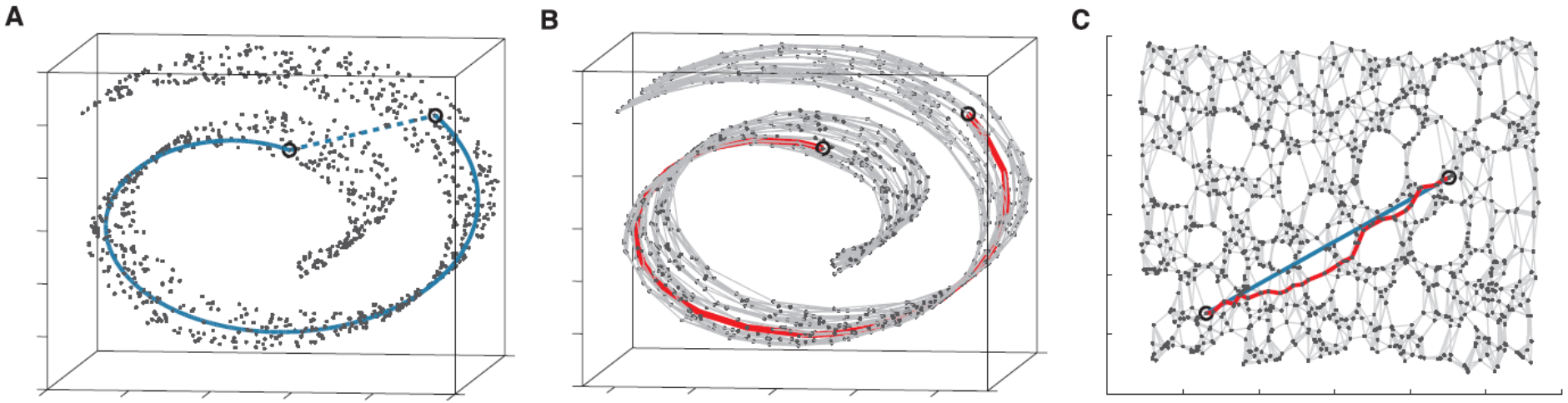


Figure 15: The "Swiss roll" data set. (A) The Euclidean distance between two points in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (B) The neighborhood graph G constructed with $K = 7$ allows an approximation (red segments) to the true geodesic path with the shortest path in G. (C) The two-dimensional embedding recovered by Isomap preserves the shortest path distances in the neighborhood graph. Straight lines in the embedding (blue) now represent cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).

Isometric Feature Mapping

- The coordinate vectors $\mathbf{y}_i$ for points in Y are chosen to minimize the cost function

$$E = \|\tau(\mathbf{D}_G) - \tau(\mathbf{D}_Y)\|_{L^2}$$

  where $\mathbf{D}_Y$ denotes the matrix of Euclidean distances $\{d_Y(i,j) = \|\mathbf{y}_i - \mathbf{y}_j\|\}$ and $\|\mathbf{A}\|_{L^2}$ the L^2 matrix norm $\sqrt{\sum_{i,j} \mathbf{A}_{ij}^2}$.
- The τ operator is defined as $\tau(\mathbf{D}) = -\mathbf{HSH}/2$, where $\mathbf{S}$ is the matrix of squared distances $\{\mathbf{S}_{ij} = \mathbf{D}_{ij}^2\}$, and $\mathbf{H}$ is the centering matrix $\{\mathbf{H}_{ij} = \delta_{ij} - 1/N\}$.
- The global minimum of the cost function is achieved by setting the coordinates $\mathbf{y}_i$ to the top d eigenvectors of the matrix $\tau(\mathbf{D}_G)$.

Isometric Feature Mapping

- Pros:
  - A noniterative, polynomial time procedure with a guarantee of global optimality.
  - A guarantee of asymptotic convergence to the true structure for manifolds whose intrinsic geometry is that of a convex region of Euclidean space.
  - Single free parameter (ϵ or K).
- Cons:
  - Sensitive to noise.
  - Computationally expensive (dense matrix eigenreduction).

Examples

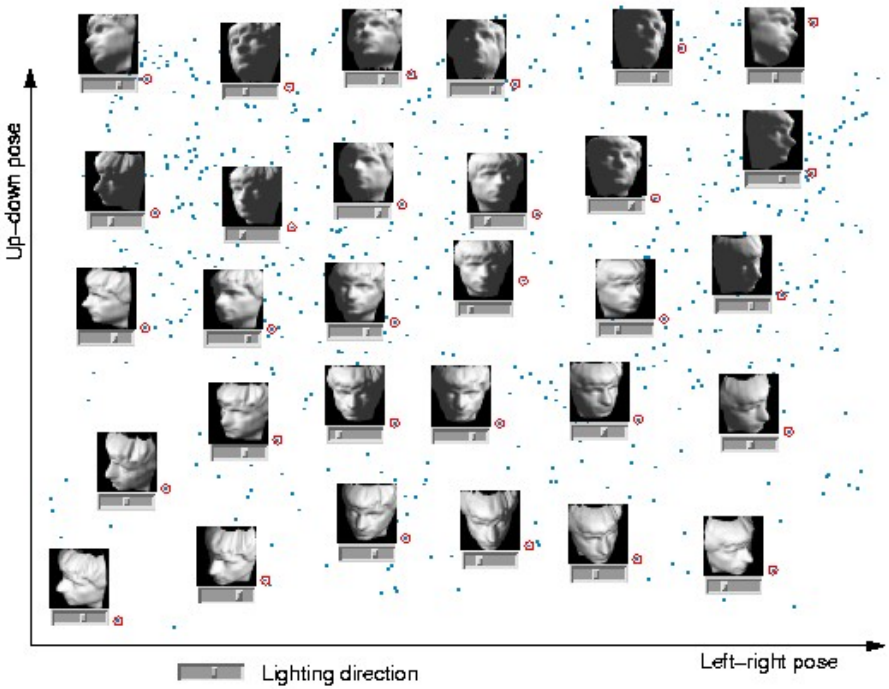


Figure 16: The input consists of 4096-dimensional vectors, representing the brightness values of 64×64 pixel images of a face rendered with different poses and lighting directions. A two-dimensional projection is shown with horizontal sliders (under the images) representing the third dimension. Each coordinate axis of the embedding correlates highly with one degree of freedom underlying the original data: left-right pose (x axis), up-down pose (y axis), and lighting direction (slider position).

Examples

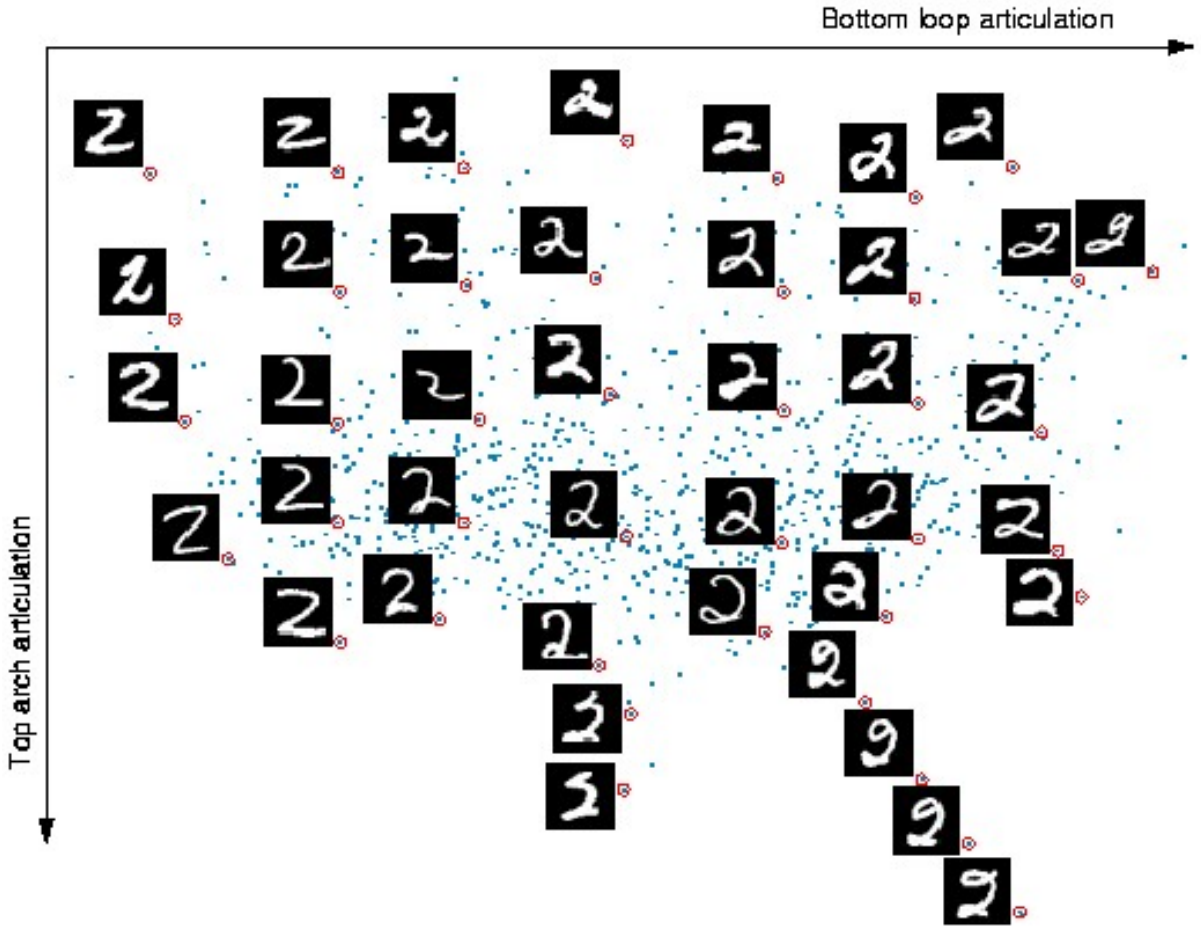


Figure 17: ϵ-Isomap applied to handwritten "2"s. The two most significant dimensions in the Isomap embedding articulate the major features of the "2": bottom loop (x axis) and top arch (y axis).

Examples

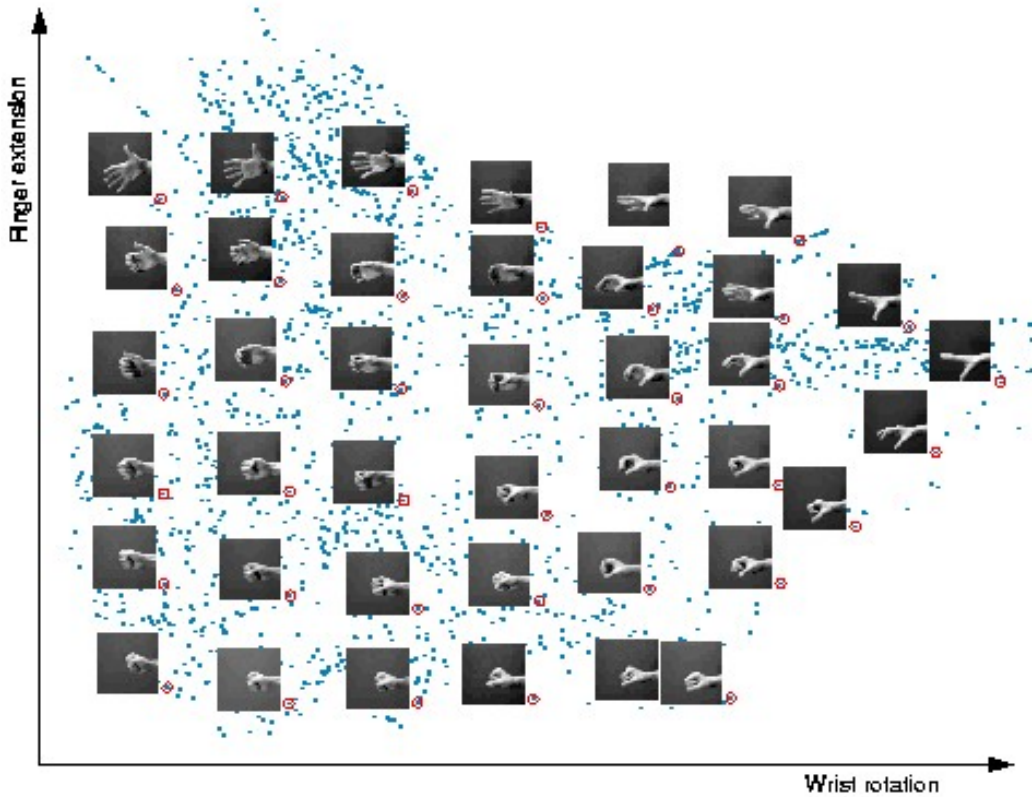


Figure 18: Isomap ($K = 6$) applied to 64×64 images of a hand in different configurations. The images were generated by making a series of opening and closing movements of the hand at different wrist orientations. The recovered coordinate axes map approximately onto the distinct underlying degrees of freedom: wrist rotation (x axis) and finger extension (y axis).

Locally Linear Embedding

- The locally linear embedding (LLE) algorithm is based on simple geometric intuitions.
- Suppose that the data consist of N real-valued vectors $\mathbf{x}_i$, each of dimensionality d, sampled from some underlying manifold.
- Provided there is sufficient data (such that the manifold is well-sampled), each data point and its neighbors are expected to lie on or close to a locally linear patch of the manifold.

Locally Linear Embedding

- The local geometry of these patches is characterized by linear coefficients that reconstruct each data point from its neighbors.
- The reconstruction errors are measured by the cost function

$$\varepsilon(\mathbf{W}) = \sum_i \left\| \mathbf{x}_i - \sum_j \mathbf{W}_{ij} \mathbf{x}_j \right\|^2$$

  which adds up the squared distances between all data points and their reconstructions.
- The weights $\mathbf{W}_{ij}$ summarize the contribution of the j'th data point to the i'th reconstruction.

Locally Linear Embedding

- To compute the weights $\mathbf{W}_{ij}$, the cost function is minimized subject to two constraints:
  - each data point $\mathbf{x}_i$ is reconstructed only from its neighbors, enforcing $\mathbf{W}_{ij} = 0$ if $\mathbf{x}_j$ does not belong to the set of neighbors of $\mathbf{x}_i$,
  - the rows of the weight matrix sum to one: $\sum_j \mathbf{W}_{ij} = 1$.
- The optimal weights $\mathbf{W}_{ij}$ subject to these constraints are found by solving a least-squares problem.

Locally Linear Embedding

- The constrained weights that minimize these reconstruction errors obey an important symmetry: for any particular data point, they are invariant to rotations, rescalings, and translations of that data point and its neighbors.
- Suppose that the data lie on or near a smooth nonlinear manifold of lower dimensionality $d' \ll d$.
- By design, the reconstruction weights $\mathbf{W}_{ij}$ reflect intrinsic geometric properties of the data that are invariant to such transformations.
- Therefore, their characterization of local geometry in the original data space is expected to be equally valid for local patches on the manifold.

Locally Linear Embedding

- LLE constructs a neighborhood-preserving mapping based on this idea.
- In the final step of the algorithm, each high-dimensional observation $\mathbf{x}_i$ is mapped to a low-dimensional vector $\mathbf{y}_i$ representing global internal coordinates on the manifold.
- This is done by choosing d'-dimensional coordinates $\mathbf{y}_i$ to minimize the embedding cost function

$$\Phi(\mathbf{Y}) = \sum_i \left\| \mathbf{y}_i - \sum_j \mathbf{W}_{ij}\mathbf{y}_j \right\|^2 .$$

Locally Linear Embedding

- This cost function, like the previous one, is based on locally linear reconstruction errors, but here the weights $\mathbf{W}_{ij}$ are fixed while optimizing the coordinates $\mathbf{y}_i$.
- The cost function can be minimized by solving a sparse eigenvalue problem whose bottom d' nonzero eigenvectors provide an ordered set of orthogonal coordinates centered on the origin.

Locally Linear Embedding

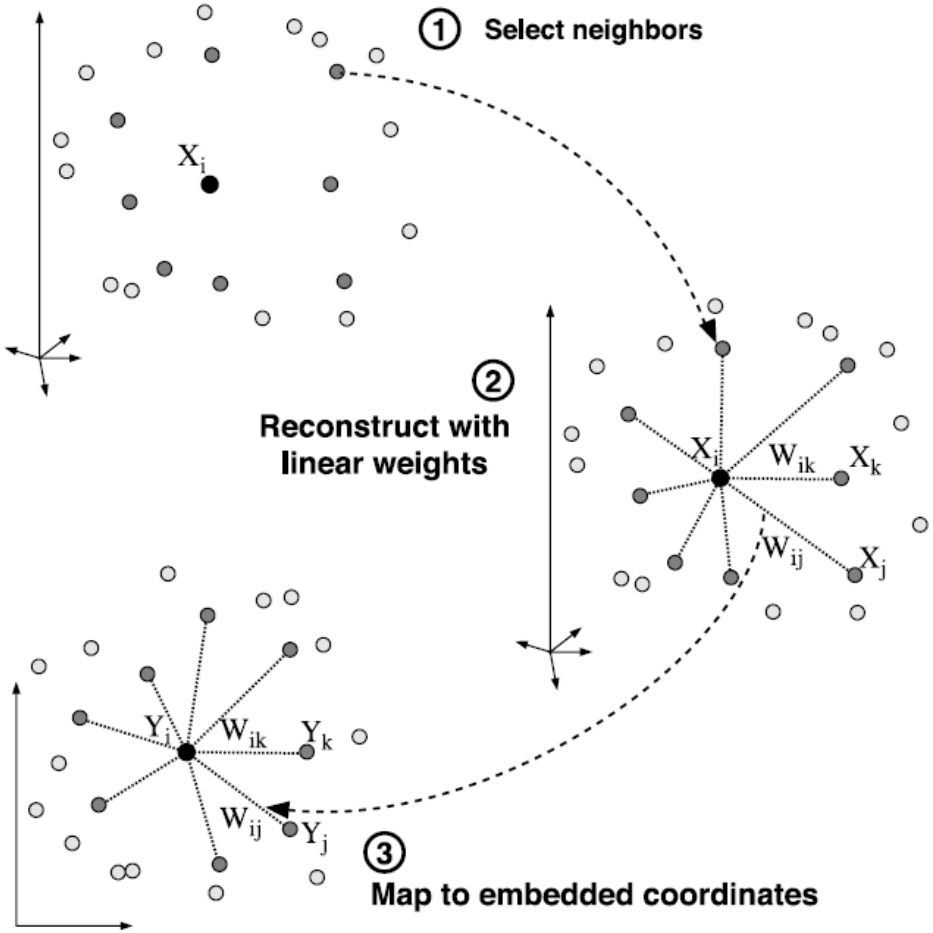


Figure 19: Steps of LLE. (1) Assign neighbors to data point $\mathbf{x}_i$. (2) Compute the weights $\mathbf{W}_{ij}$ that best reconstruct $\mathbf{x}_i$ from its neighbors. (3) Compute the low-dimensional embedding vectors $\mathbf{y}_i$ best reconstructed by $\mathbf{W}_{ij}$.

Locally Linear Embedding

- Pros:
  - Globally optimal result.
  - Single free parameter (number of neighbors, K).
  - Simple linear algebra operations using sparse matrices.
- Cons:
  - Sensitive to noise.
  - No theoretical guarantees.

Examples

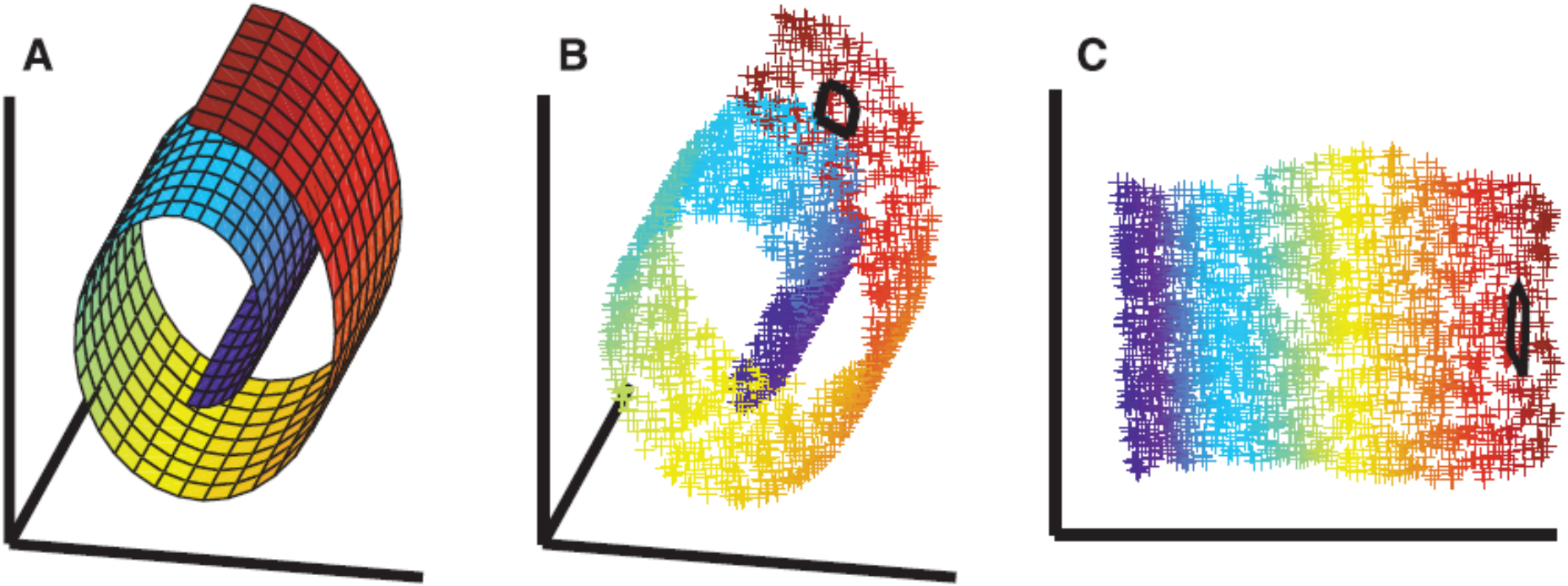


Figure 20: The “Swiss roll” data set. The color coding illustrates the neighborhood-preserving mapping discovered by LLE. Black outlines in (B) and (C) show the neighborhood of a single point.

Examples

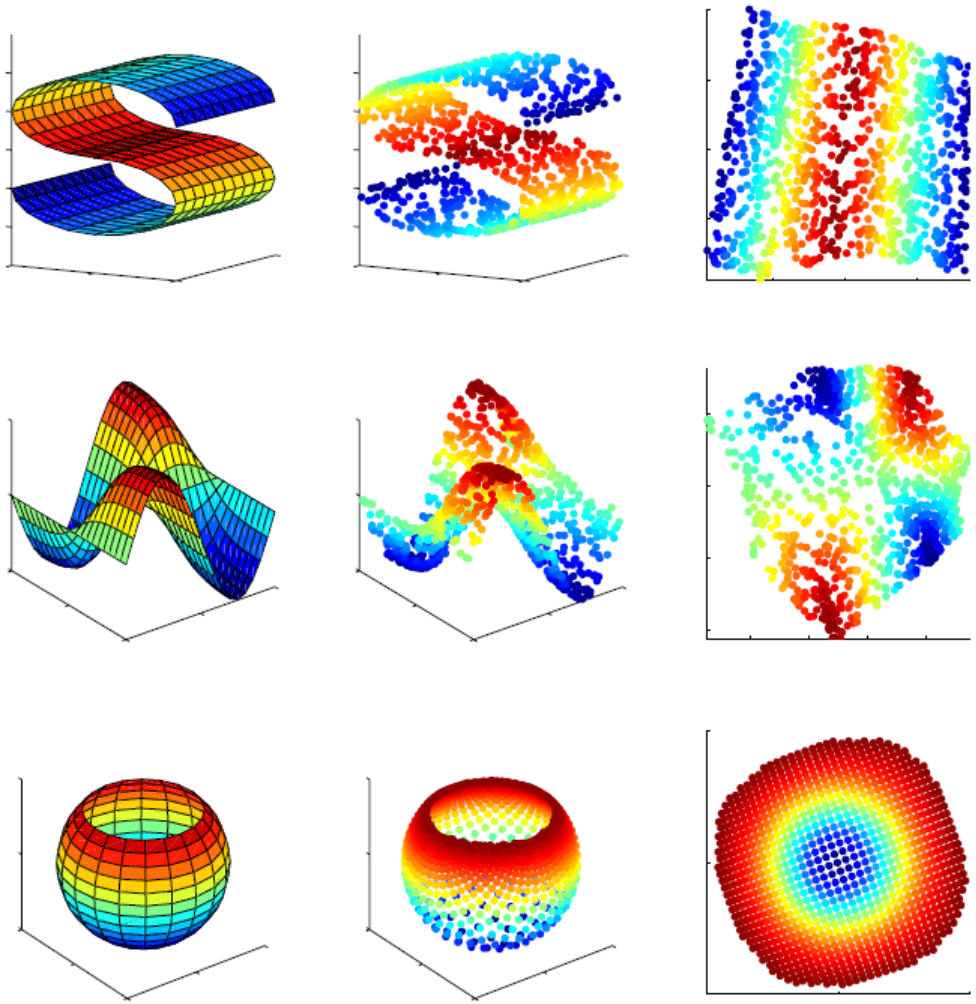


Figure 21: Other examples for three-dimensional data sampled from two dimensional manifolds.

Examples

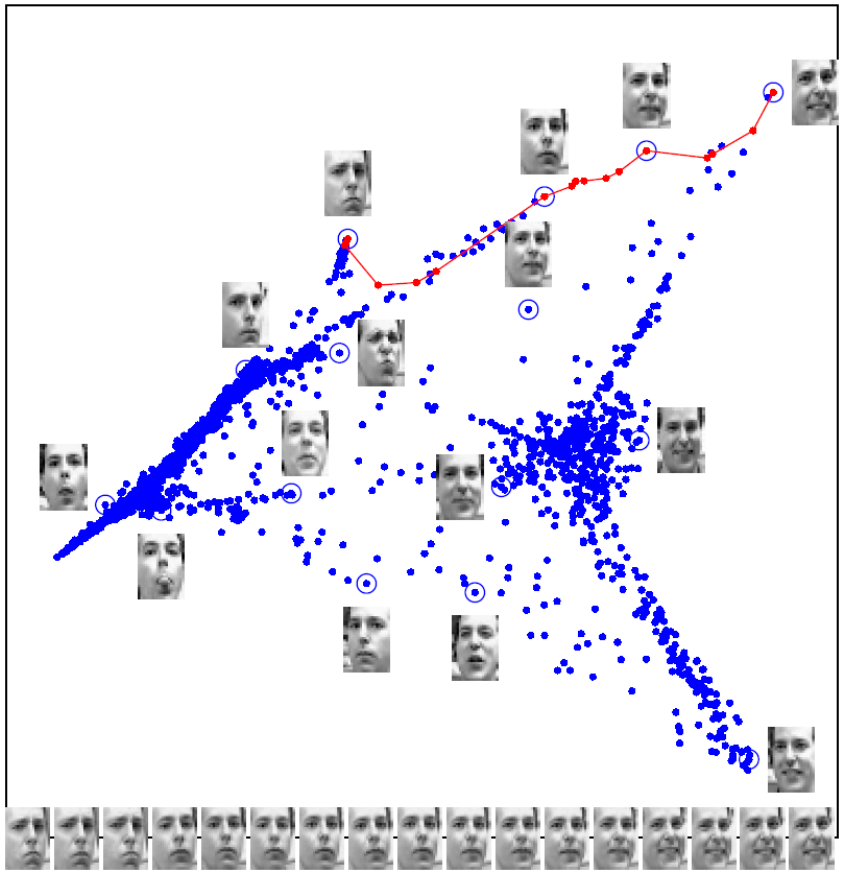


Figure 22: Images of faces, digitized at 20×28 pixels, mapped into the embedding space described by the first two coordinates of LLE. The bottom images correspond to points along the top-right path (linked by solid red line), illustrating one particular mode of variability in pose and expression.

Examples

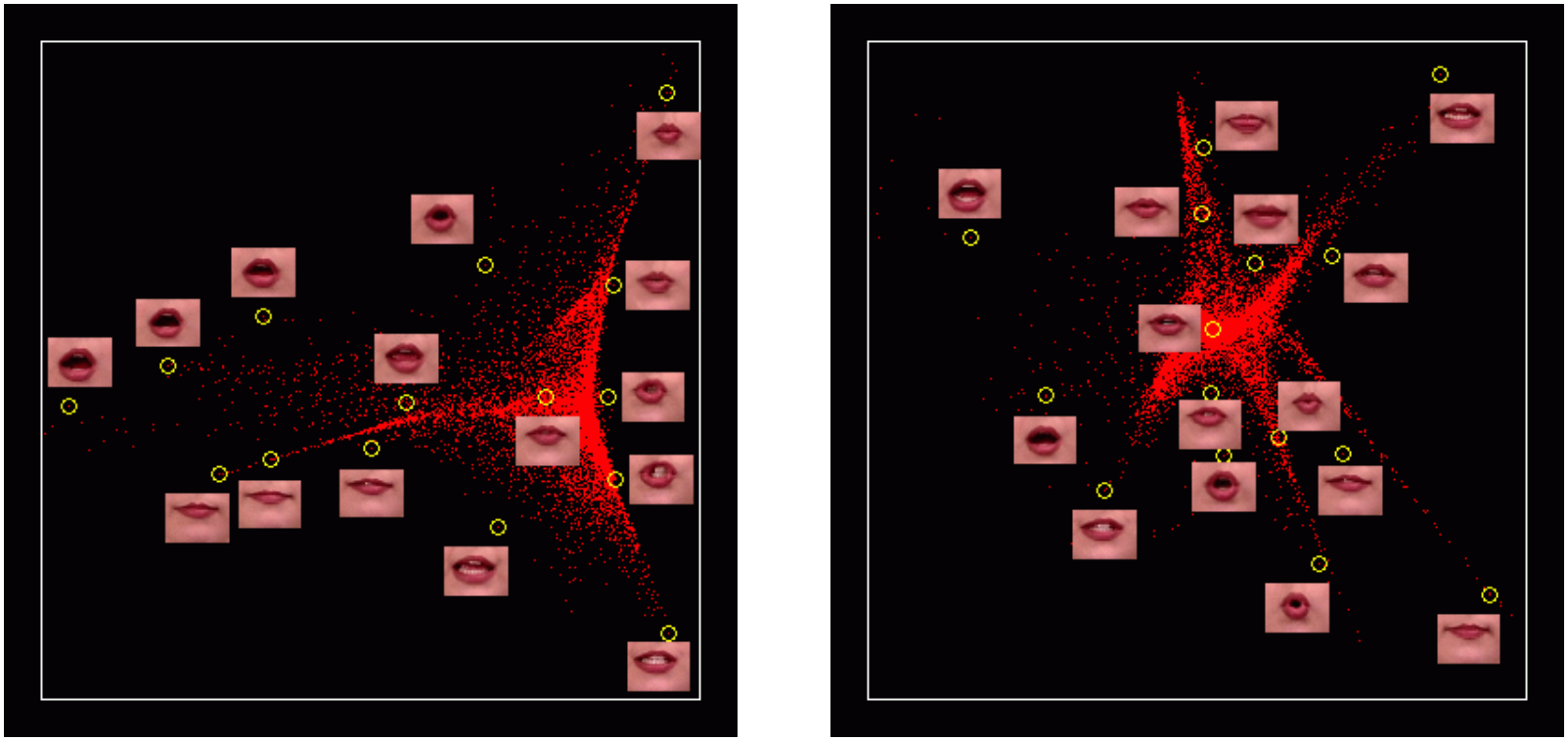


Figure 23: Images of lips, mapped into the embedding space described by the first two coordinates of LLE.

Feature Reduction

Table 1: Feature reduction methods.

Method	Property	Comments
Principal Component Analysis (PCA)	Linear map; fast; eigenvector-based.	Traditional, eigenvector based method, also known as Karhunen-Loève expansion; good for Gaussian data.
Linear Discriminant Analysis	Supervised linear map; fast; eigenvector-based.	Better than PCA for classification; limited to $(c-1)$ components with non-zero eigenvalues.
Projection Pursuit	Linear map; iterative; non-Gaussian.	Mainly used for interactive exploratory data-analysis.
Independent Component Analysis (ICA)	Linear map, iterative, non-Gaussian.	Blind source separation, used for de-mixing non-Gaussian distributed sources (features).
Kernel PCA	Nonlinear map; eigenvector-based.	PCA-based method, using a kernel to replace inner products of pattern vectors.
PCA Network	Linear map; iterative.	Auto-associative neural network with linear transfer functions and just one hidden layer.
Nonlinear PCA	Linear map; non-Gaussian criterion; usually iterative	Neural network approach, possibly used for ICA.
Nonlinear auto-associative network	Nonlinear map; non-Gaussian criterion; iterative.	Bottleneck network with several hidden layers; the nonlinear map is optimized by a nonlinear reconstruction; input is used as target.
Multidimensional scaling (MDS), and Sammon's projection	Nonlinear map; iterative.	Often poor generalization; sample size limited; noise sensitive; mainly used for 2-dimensional visualization.
Self-Organizing Map (SOM)	Nonlinear; iterative.	Based on a grid of neurons in the feature space; suitable for extracting spaces of low dimensionality.

Feature Selection

- An alternative to feature reduction that uses linear or non-linear combinations of features is feature selection that reduces dimensionality by selecting subsets of existing features.
- The first step in feature selection is to define a criterion function that is often a function of the classification error.
- Note that, the use of classification error in the criterion function makes feature selection procedures dependent on the specific classifier used.

Feature Selection

- The most straightforward approach would require
  - examining all $\binom{d}{m}$ possible subsets of size m,
  - selecting the subset that performs the best according to the criterion function.
- The number of subsets grows combinatorially, making the exhaustive search impractical.
- Iterative procedures are often used but they cannot guarantee the selection of the optimal subset.

Feature Selection

- *Sequential forward selection:*
  - First, the best single feature is selected.
  - Then, pairs of features are formed using one of the remaining features and this best feature, and the best pair is selected.
  - Next, triplets of features are formed using one of the remaining features and these two best features, and the best triplet is selected.
  - This procedure continues until all or a predefined number of features are selected.

Feature Selection

- *Sequential backward selection:*
  - First, the criterion function is computed for all d features.
  - Then, each feature is deleted one at a time, the criterion function is computed for all subsets with $d-1$ features, and the worst feature is discarded.
  - Next, each feature among the remaining $d-1$ is deleted one at a time, and the worst feature is discarded to form a subset with $d-2$ features.
  - This procedure continues until one feature or a predefined number of features are left.

Examples

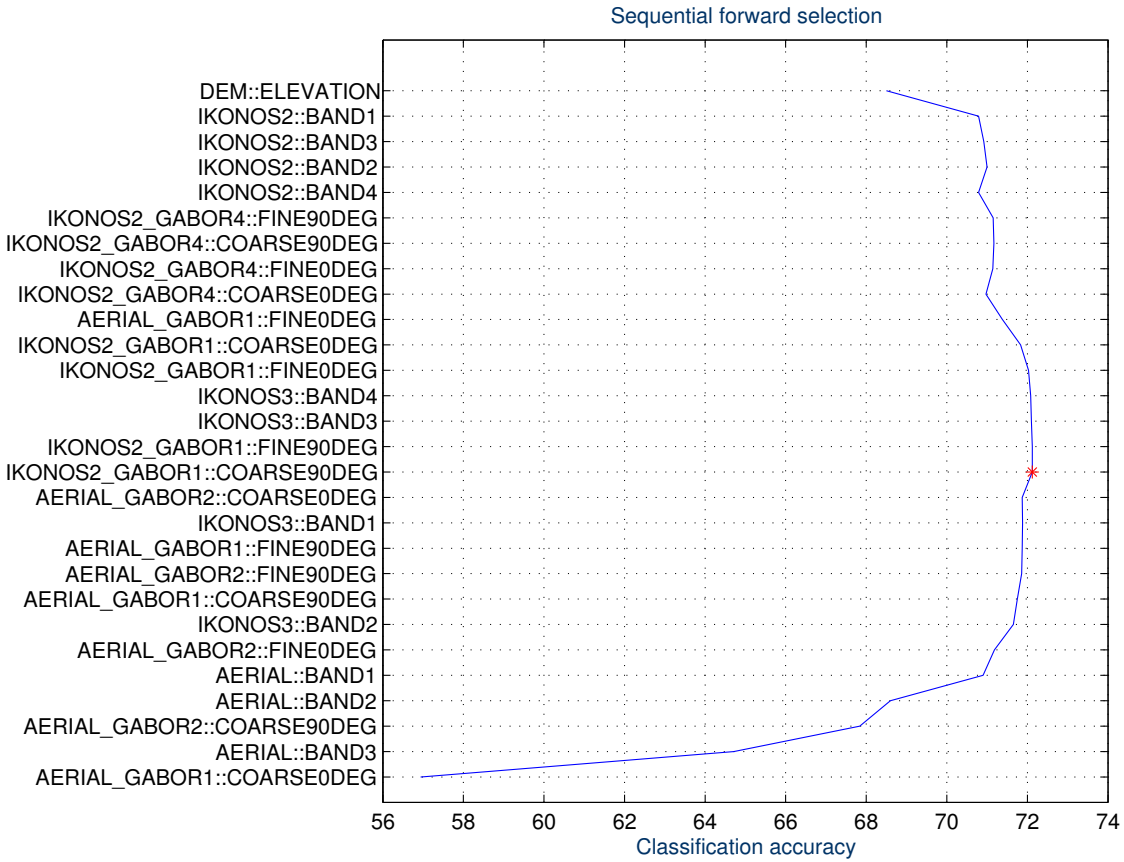


Figure 24: Results of sequential forward feature selection for classification of a satellite image using 28 features. x-axis shows the classification accuracy (%) and y-axis shows the features added at each iteration (the first iteration is at the bottom). The highest accuracy value is shown with a star.

Examples

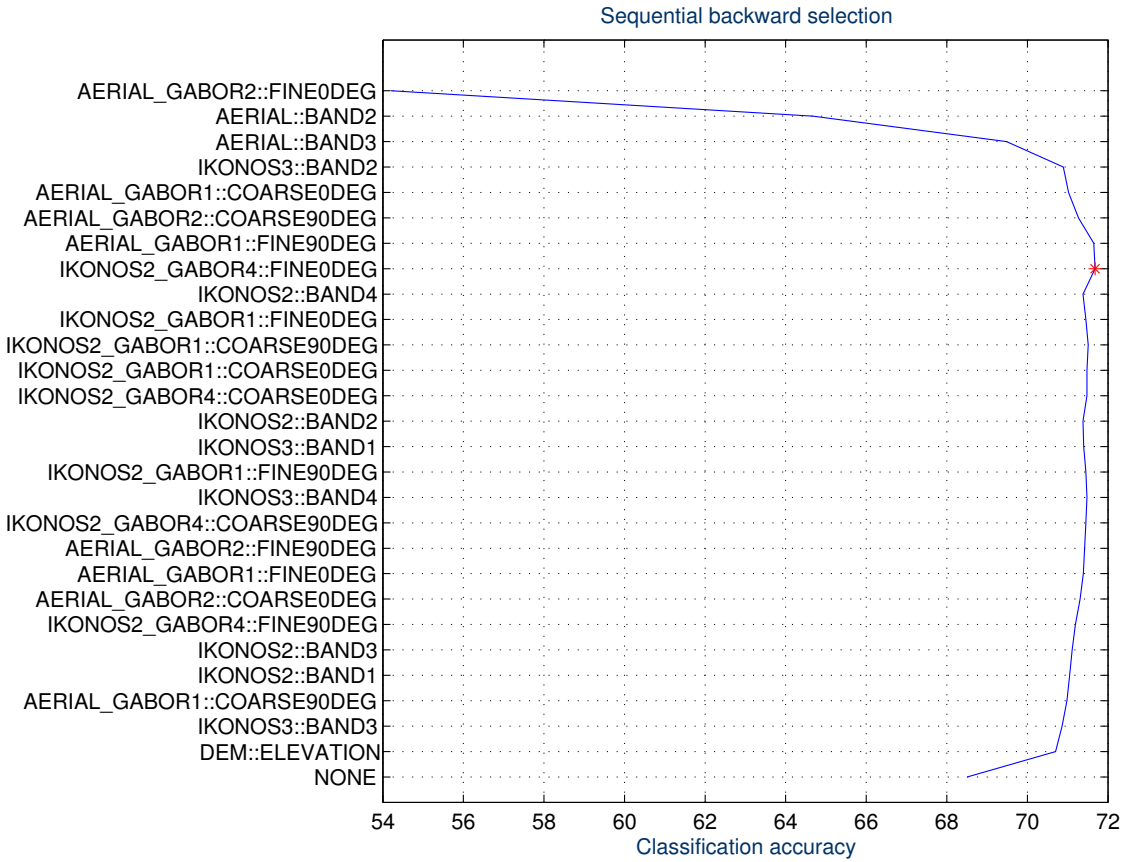


Figure 25: Results of sequential backward feature selection for classification of a satellite image using 28 features. x-axis shows the classification accuracy (%) and y-axis shows the features removed at each iteration (the first iteration is at the bottom). The highest accuracy value is shown with a star.

Feature Selection

Table 2: Feature selection methods.

Method	Property	Comments
Exhaustive Search	Evaluate all $\binom{d}{m}$ possible subsets.	Guaranteed to find the optimal subset; not feasible for even moderately large values of m and d.
Branch-and-Bound Search	Uses the well-known branch-and-bound search method; only a fraction of all possible feature subsets need to be enumerated to find the optimal subset.	Guaranteed to find the optimal subset provided the criterion function satisfies the monotonicity property; the worst-case complexity of this algorithm is exponential.
Best Individual Features	Evaluate all the m features individually; select the best m individual features.	Computationally simple; not likely to lead to an optimal subset.
Sequential Forward Selection (SFS)	Select the best single feature and then add one feature at a time which in combination with the selected features maximizes the criterion function.	Once a feature is retained, it cannot be discarded; computationally attractive since to select a subset of size 2, it examines only $(d-1)$ possible subsets.
Sequential Backward Selection (SBS)	Start with all the d features and successively delete one feature at a time.	Once a feature is deleted, it cannot be brought back into the optimal subset; requires more computation than sequential forward selection.
"Plus l-take away r" Selection	First enlarge the feature subset by l features using forward selection and then delete r features using backward selection.	Avoids the problem of feature subset "nesting" encountered in SFS and SBS methods; need to select values of l and $r (l > r)$.
Sequential Forward Floating Search (SFFS) and Sequential Backward Floating Search (SBFS)	A generalization of "plus-l take away-r" method; the values of l and r are determined automatically and updated dynamically.	Provides close to optimal solution at an affordable computational cost.

Summary

- The choice between feature reduction and feature selection depends on the application domain and the specific training data.
- Feature selection leads to savings in computational costs and the selected features retain their original physical interpretation.
- Feature reduction with transformations may provide a better discriminative ability but these new features may not have a clear physical meaning.