# Feature Reduction and Selection

Selim Aksoy

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

GE 461, Spring 2020

# Introduction

- We are considering a scenario in which we have a set of samples where each sample is modeled by a set of features (attributes).
- In practical applications, it is not unusual to encounter problems involving hundreds or thousands 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.

# 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 accuracy affected by the dimensionality (relative to the amount of available data)?
  - How is the complexity of the model affected by the dimensionality?

# Problems of Dimensionality

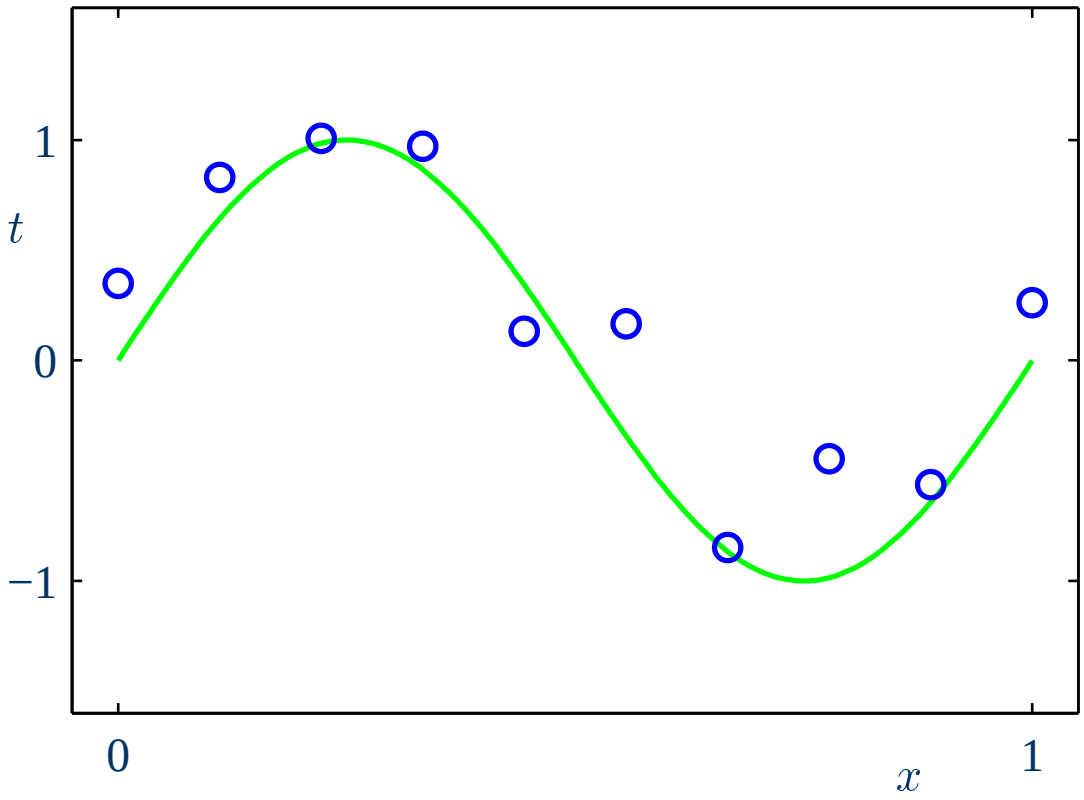
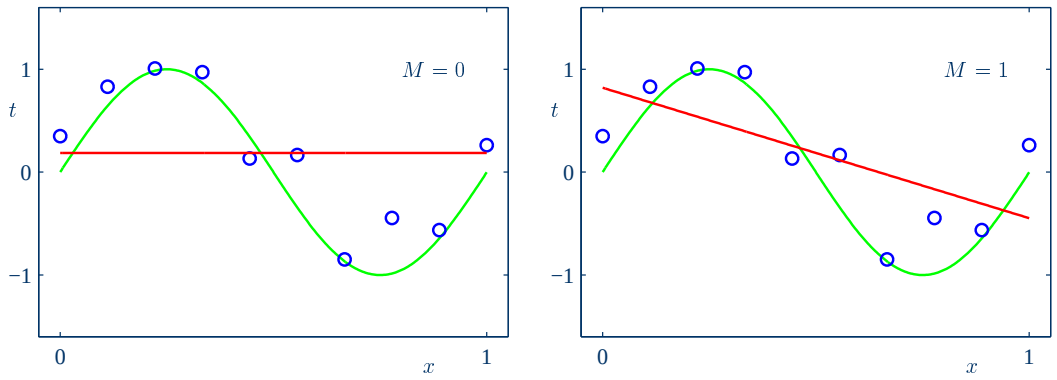



Figure 1: Regression example: plot of 10 sample points for the input variable $x$ along with the corresponding target variable $t$. Green curve is the true function that generated the data.

# Problems of Dimensionality

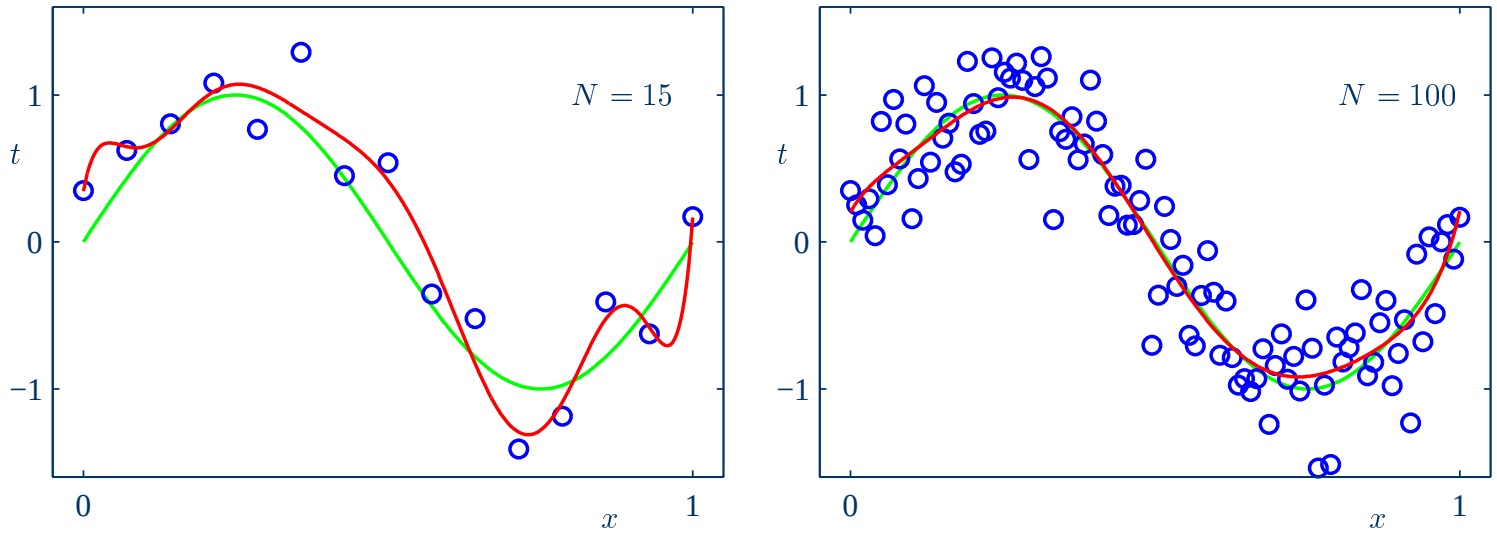



(a) 0'th order polynomial (b) 1'st order polynomial

(c) 3'rd order polynomial (d) 9'th order polynomial

Figure 2: Polynomial curve fitting: plots of polynomials having various orders, shown as red curves, fitted to the set of 10 sample points.

# Problems of Dimensionality




(a) 15 sample points (b) 100 sample points

Figure 3: Polynomial curve fitting: plots of 9'th order polynomials fitted to 15 and 100 sample points.

# Problems of Dimensionality

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

# 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).
  - 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.

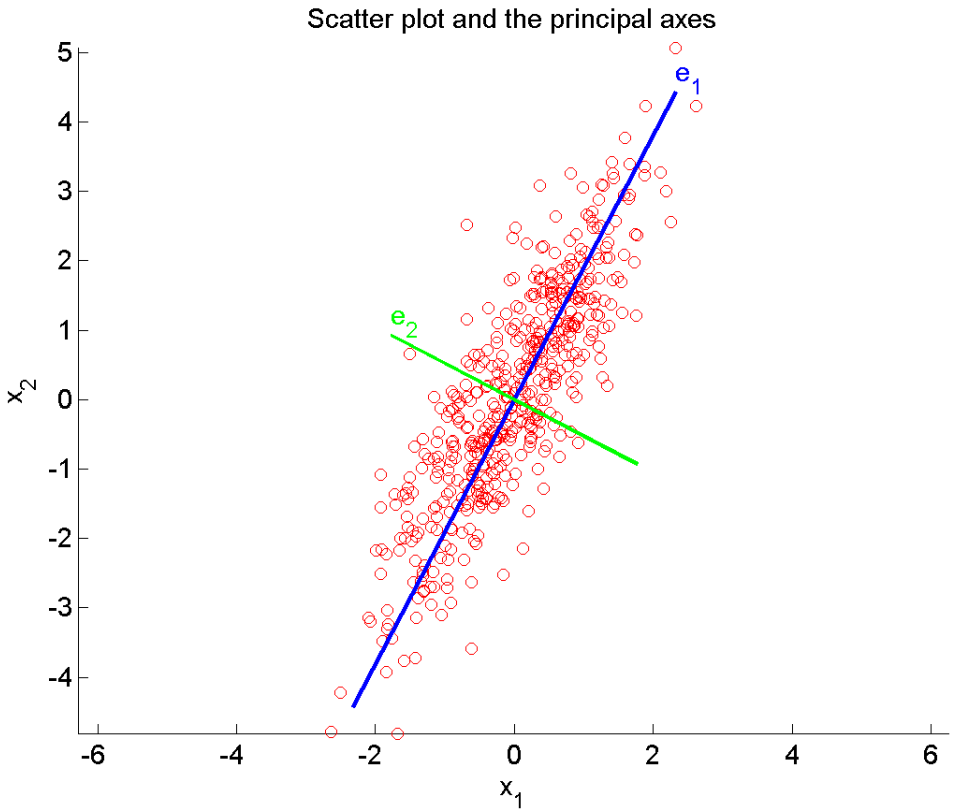
# Principal Components Analysis

- 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*

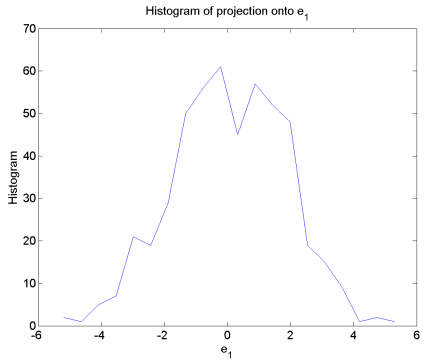
$$S = \sum_{i=1}^{n} (\mathbf{x_i} - \boldsymbol{\mu})(\mathbf{x_i} - \boldsymbol{\mu})^T$$

  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.

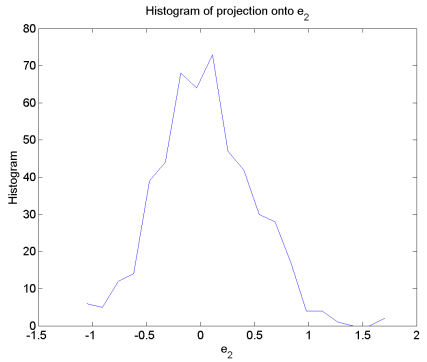
# Examples




(a) Scatter plot.




(b) Projection onto $e_1$.




(c) Projection onto $e_2$.

Figure 4: 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

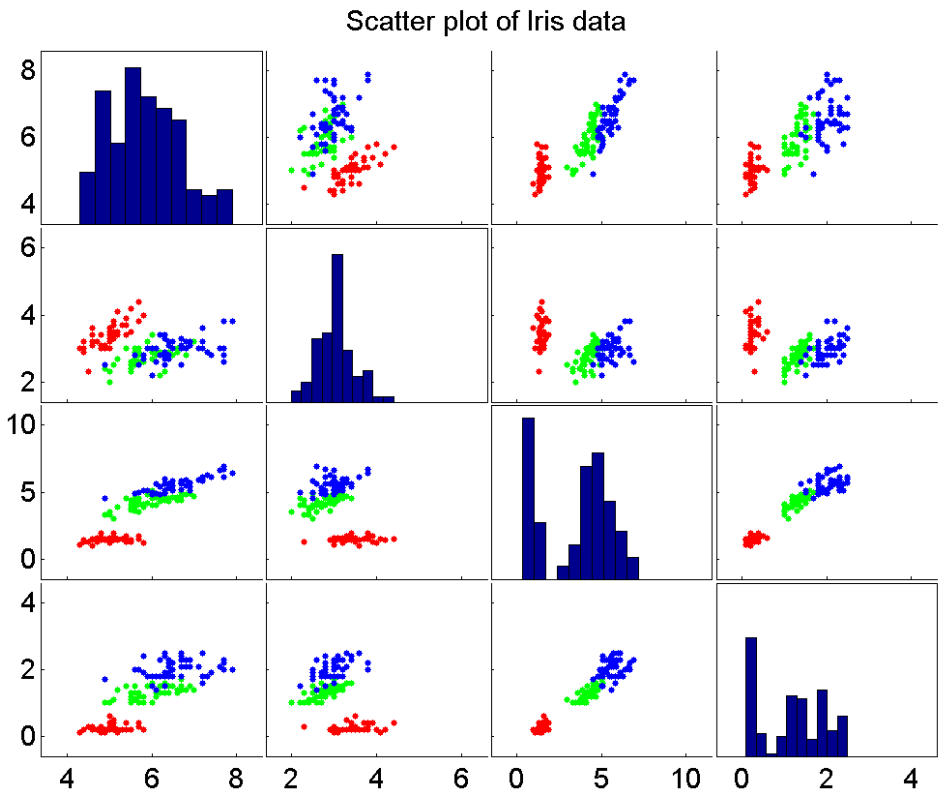



Figure 5: Scatter plot of the iris data. Diagonal cells show the histogram for each feature. Other cells show scatters of pairs of features $x_1, x_2, x_3, x_4$ in top-down and left-right order. Red, green and blue points represent samples for the setosa, versicolor and virginica classes, respectively.

# Examples

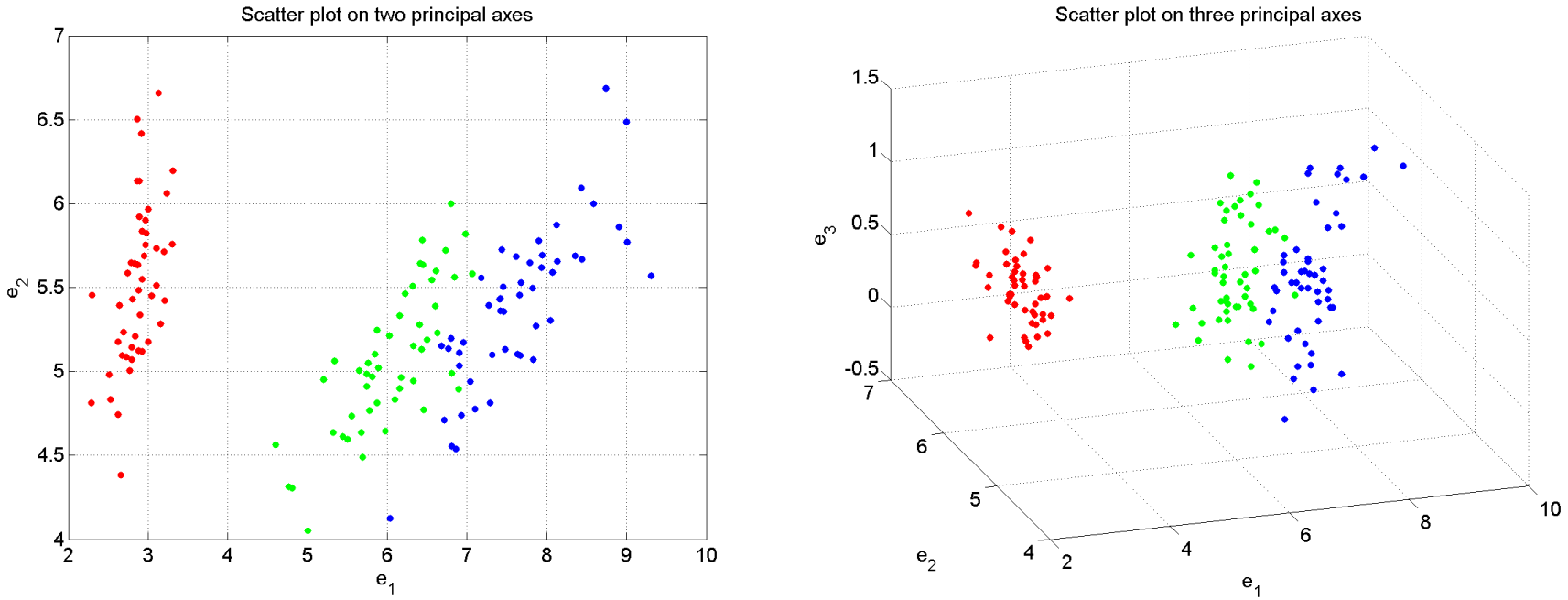



Figure 6: Scatter plot of the projection of the iris data onto the first two and the first three principal axes. Red, green and blue points represent samples for the setosa, versicolor and virginica classes, respectively.

# 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

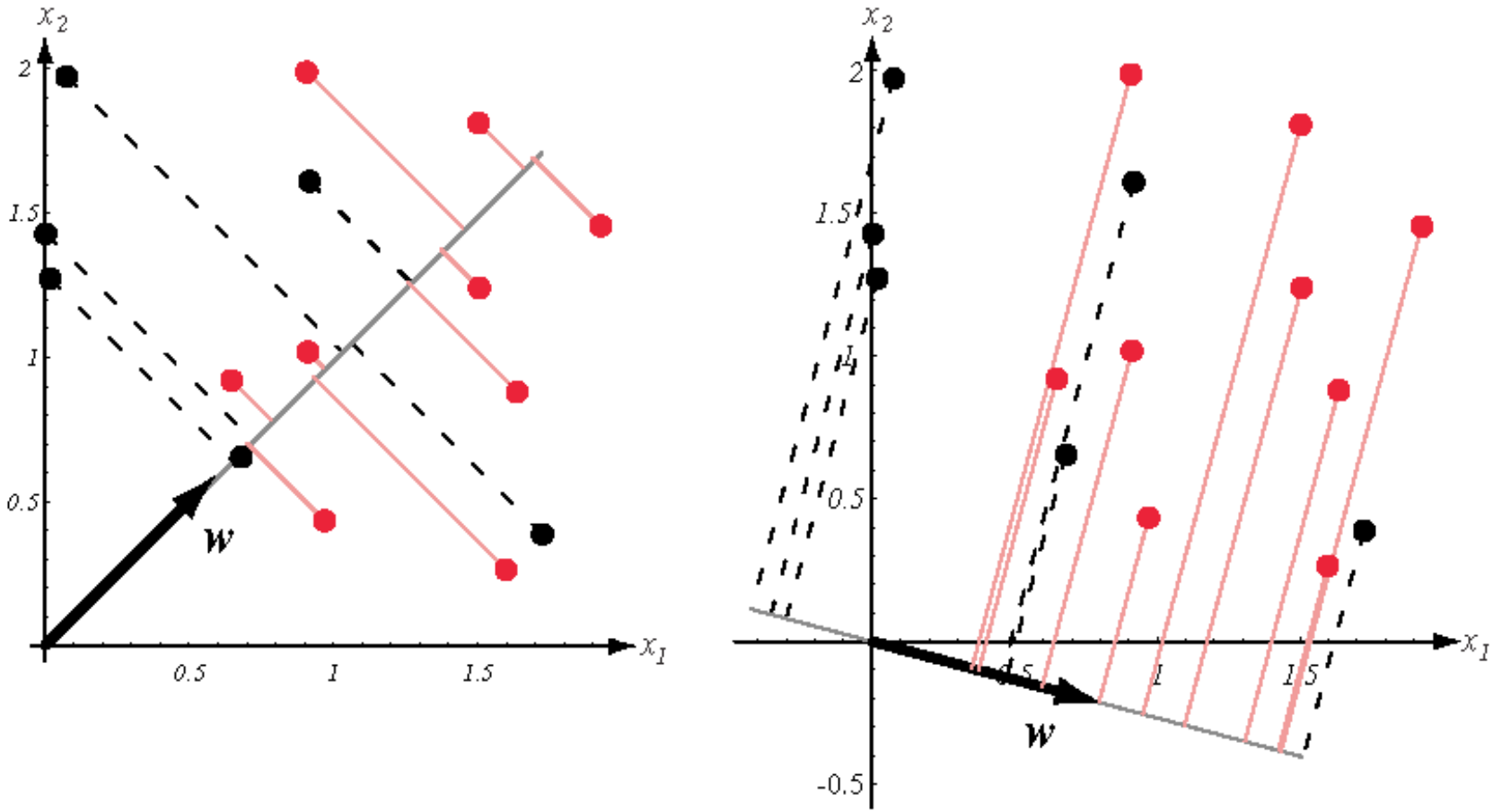



Figure 7: Projection of the same set of samples onto two different lines in the directions marked as $\mathbf{w}$. The figure on the right shows greater separation between the red and black projected points.

# 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

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

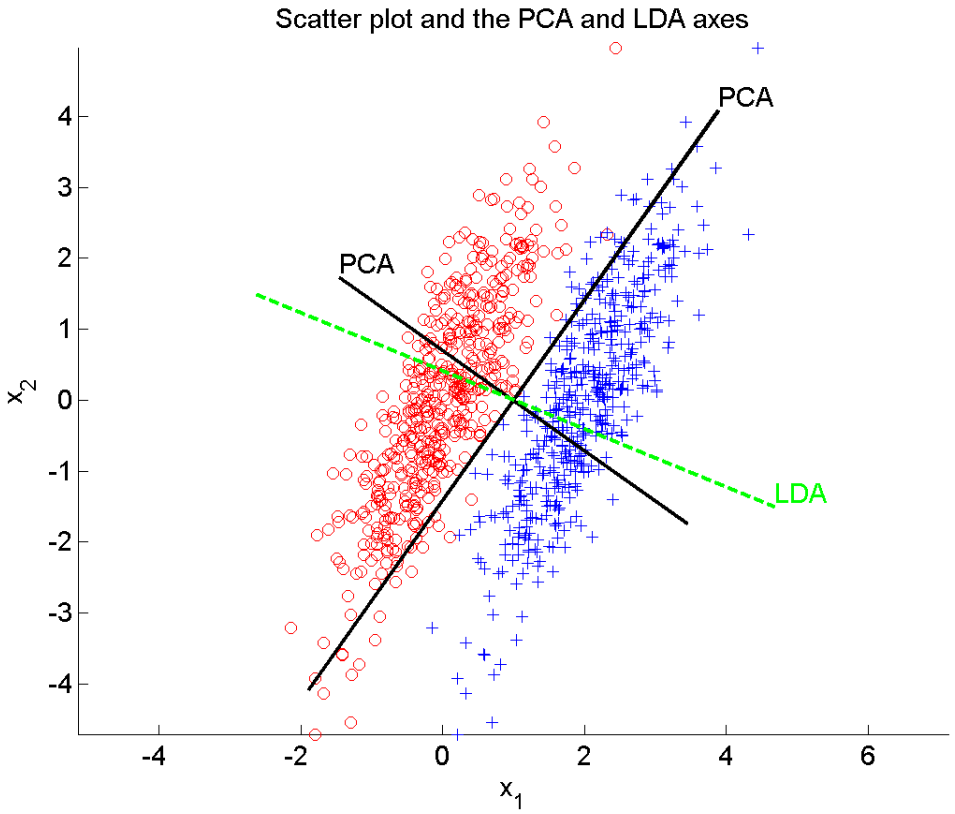
$$J(\mathbf{W}) = \frac{|\mathbf{W}^T \mathbf{S_B} \mathbf{W}|}{|\mathbf{W}^T \mathbf{S_W} \mathbf{W}|}$$

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

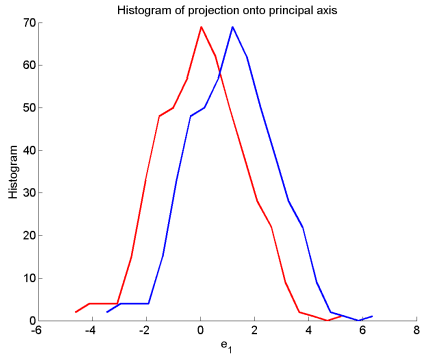
# Linear Discriminant Analysis

- 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.

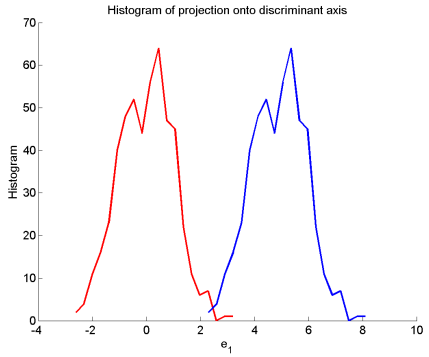
# Examples




(a) Scatter plot.



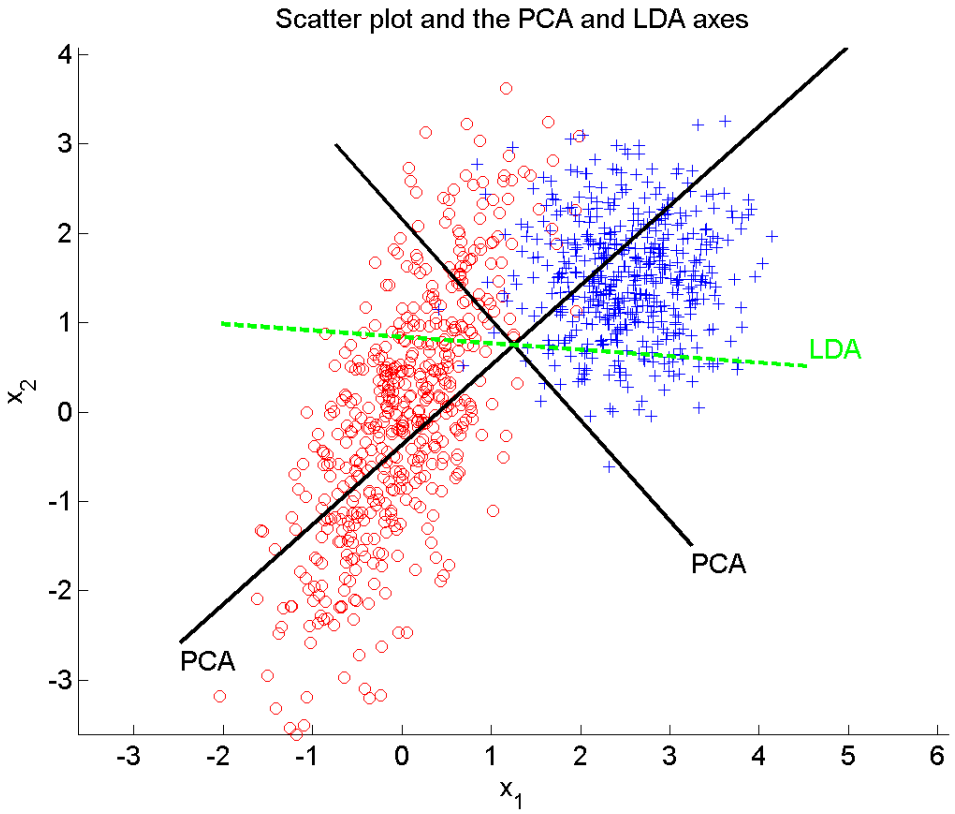

(b) Projection onto the first PCA axis.



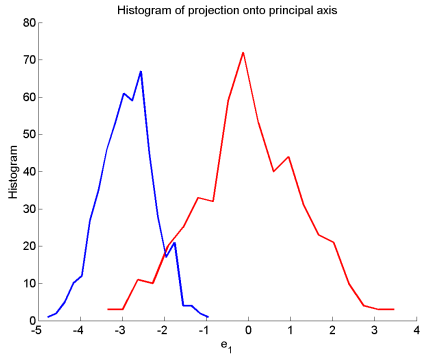

(c) Projection onto the first LDA axis.

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.

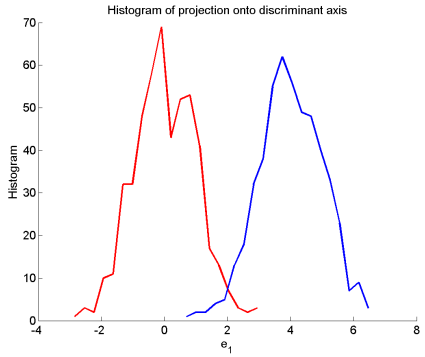
# Examples




(a) Scatter plot.




(b) Projection onto the first PCA axis.




(c) Projection onto the first LDA axis.

Figure 9: 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

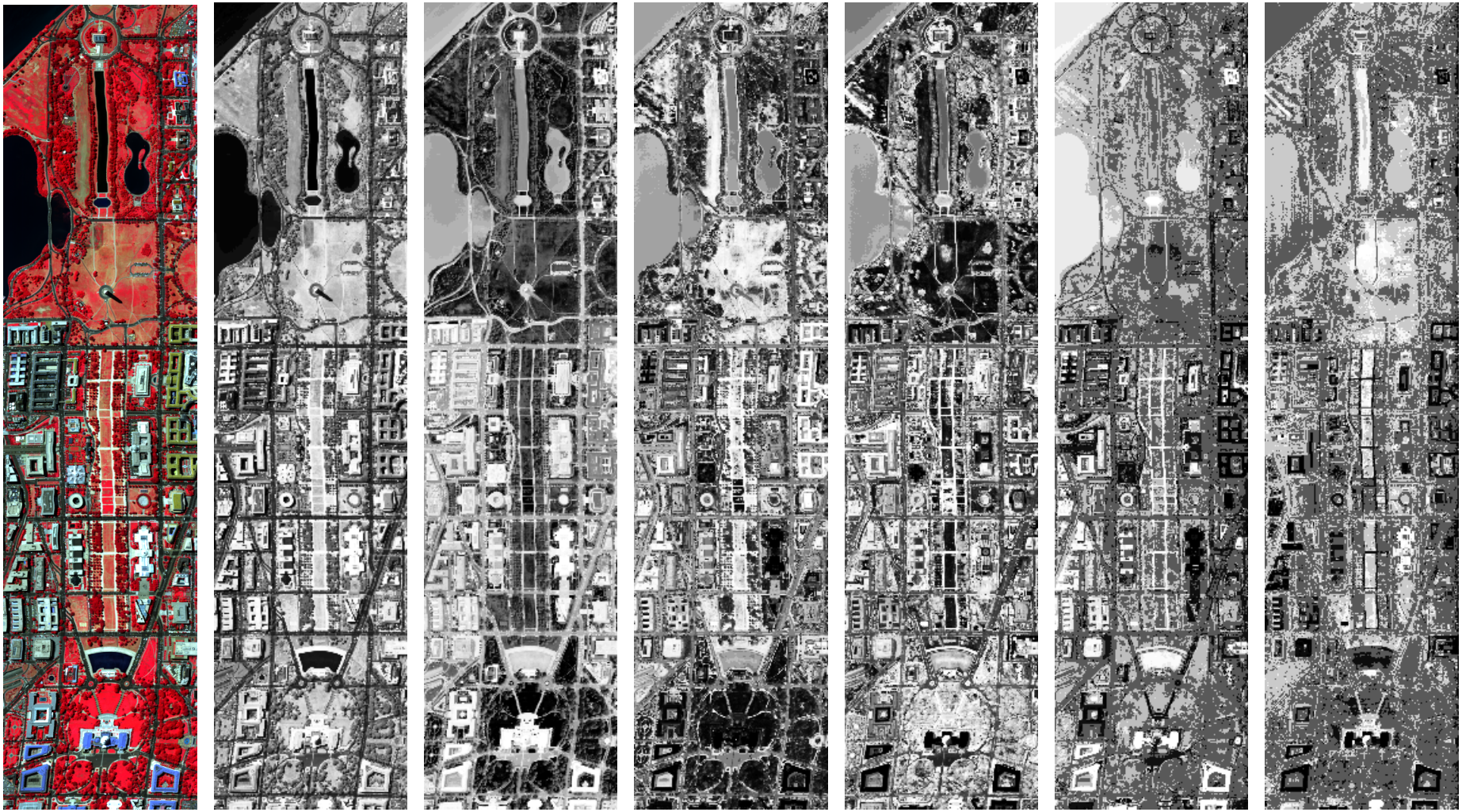


Figure 10: A hyperspectral image and the first six PCA bands (after projection).

# Examples

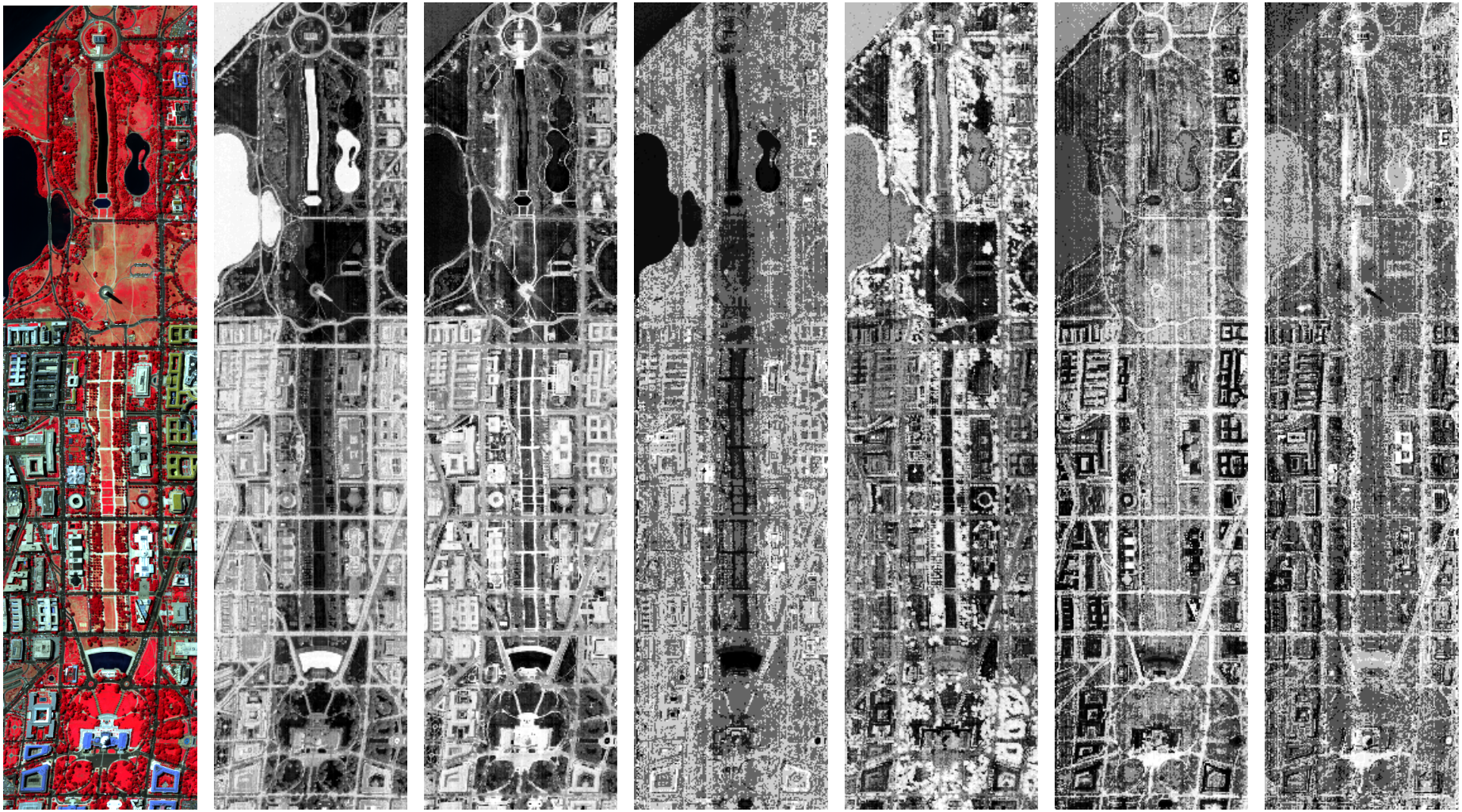


Figure 11: A hyperspectral image and the six LDA bands (after projection).

# Examples

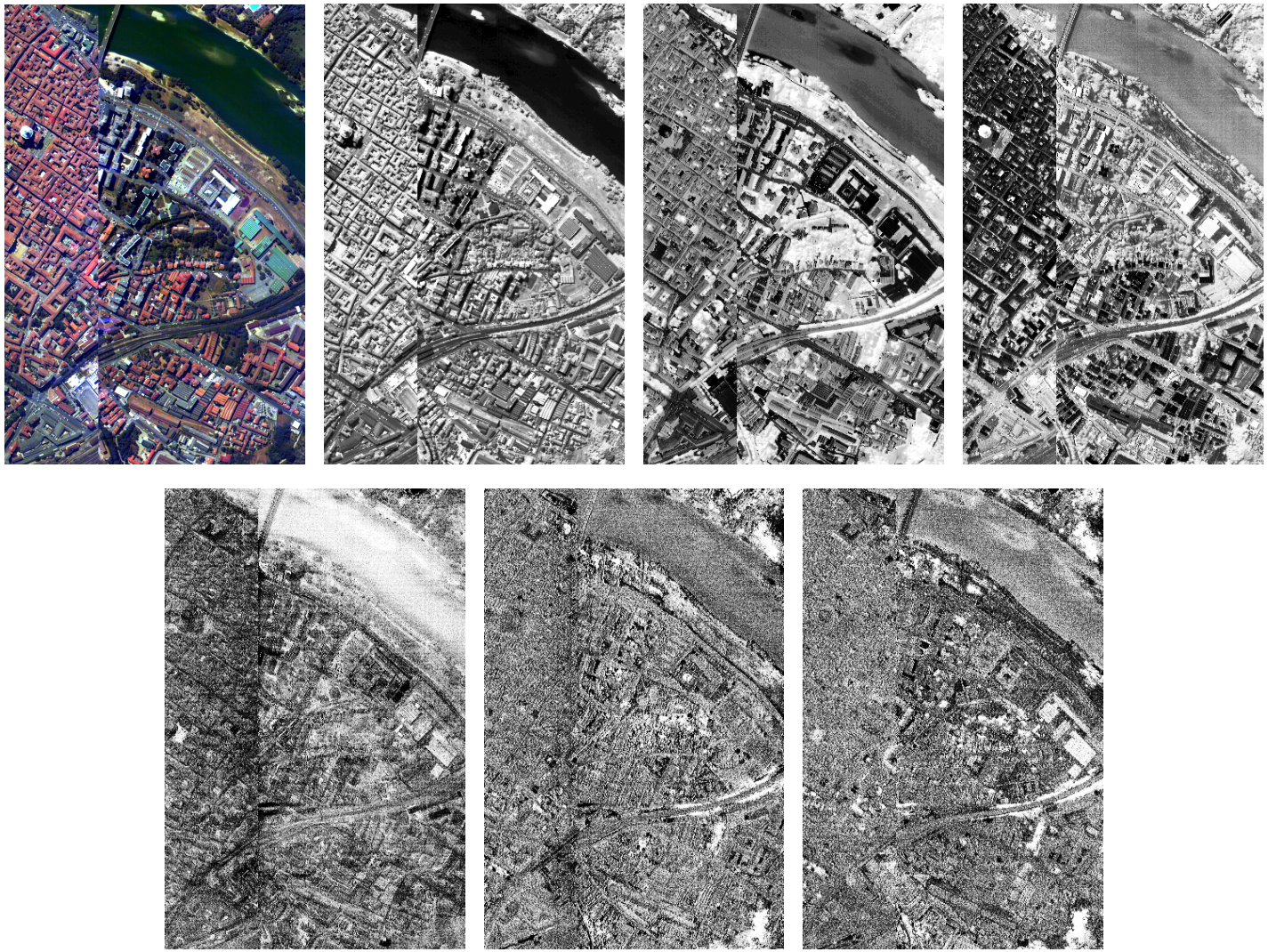


Figure 12: A hyperspectral image and the first six PCA bands (after projection).

# Examples

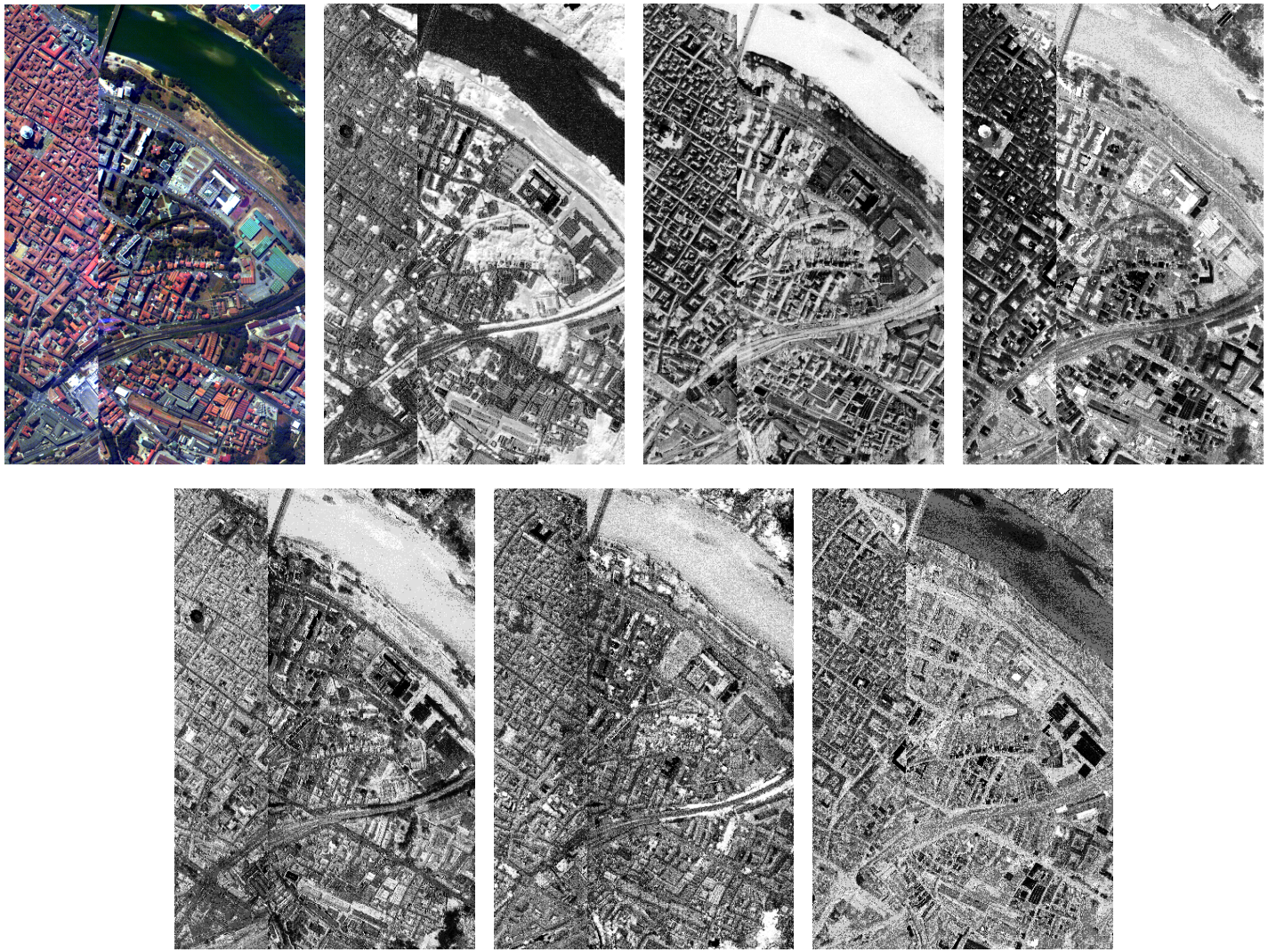


Figure 13: A hyperspectral image and the six LDA bands (after projection)

# Examples

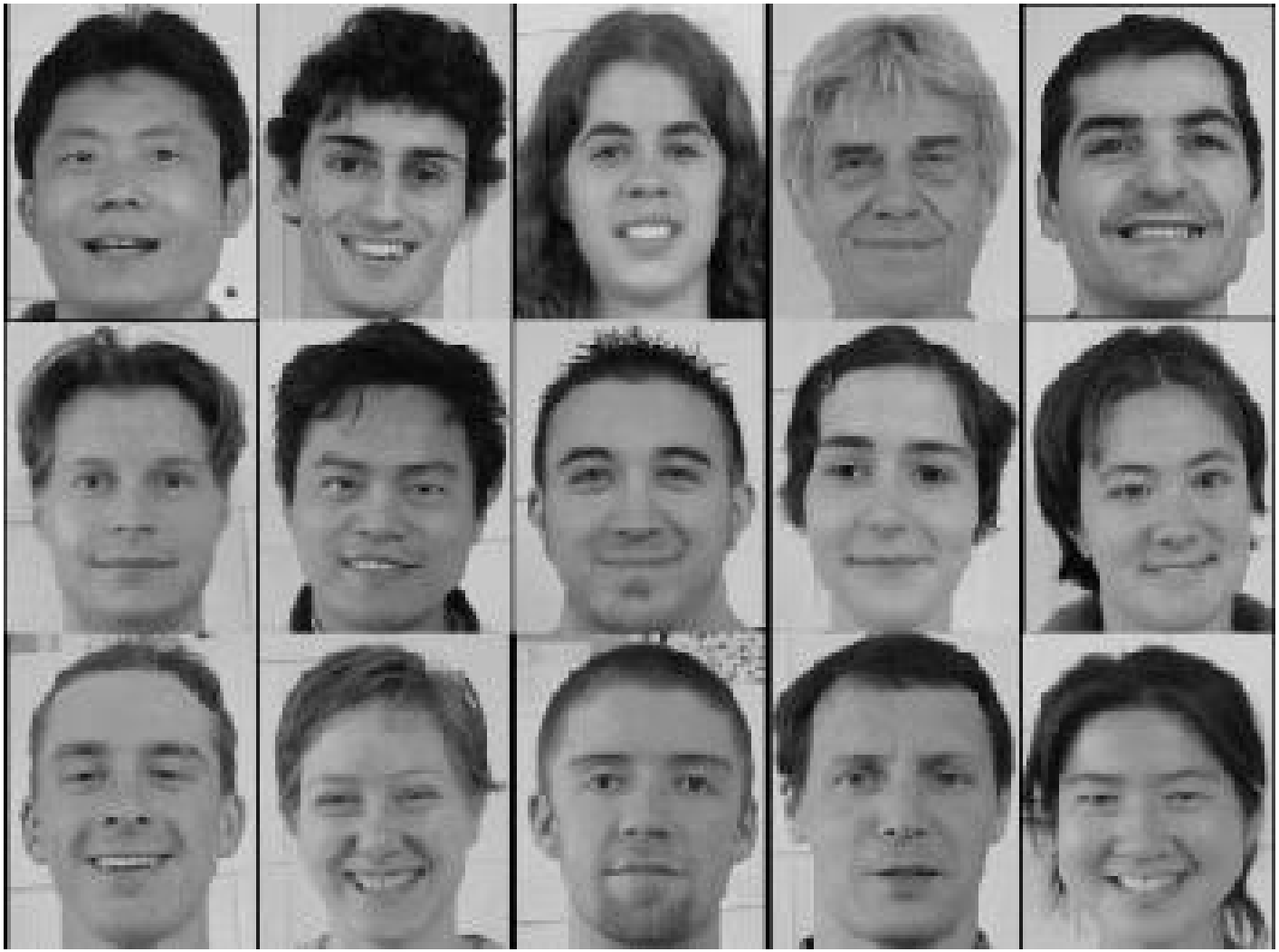


Figure 14: Example face images. (Taken from http://www.geop.ubc.ca/CDSST/eigenfaces.html.)

# Examples

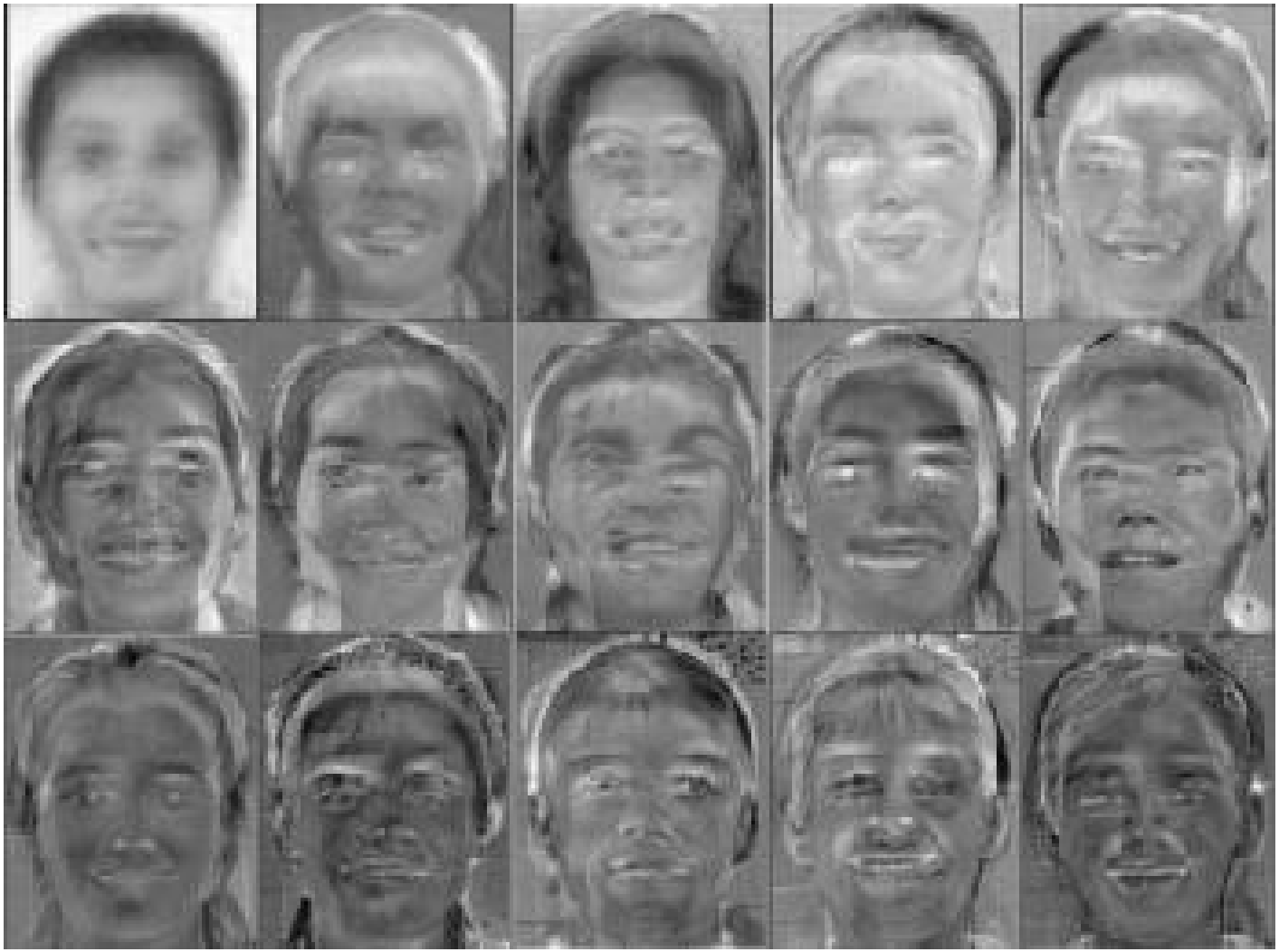


Figure 15: Eigenvectors (principal axes) of the face images (often referred to as eigenfaces).

# Isometric Feature Mapping

- The isometric feature mapping (Isomap) algorithm combines the major algorithmic features of PCA and multi-dimensional scaling with the flexibility to learn a broad class of nonlinear manifolds.
- A manifold is a topological space that locally resembles Euclidean space near each point.
- The approach seeks to preserve the intrinsic geometry of the data, as captured in the geodesic manifold distances between all pairs of data points.
- The essential point is to estimate the geodesic distance between faraway points, given only input-space distances.

# Isometric Feature Mapping

- 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

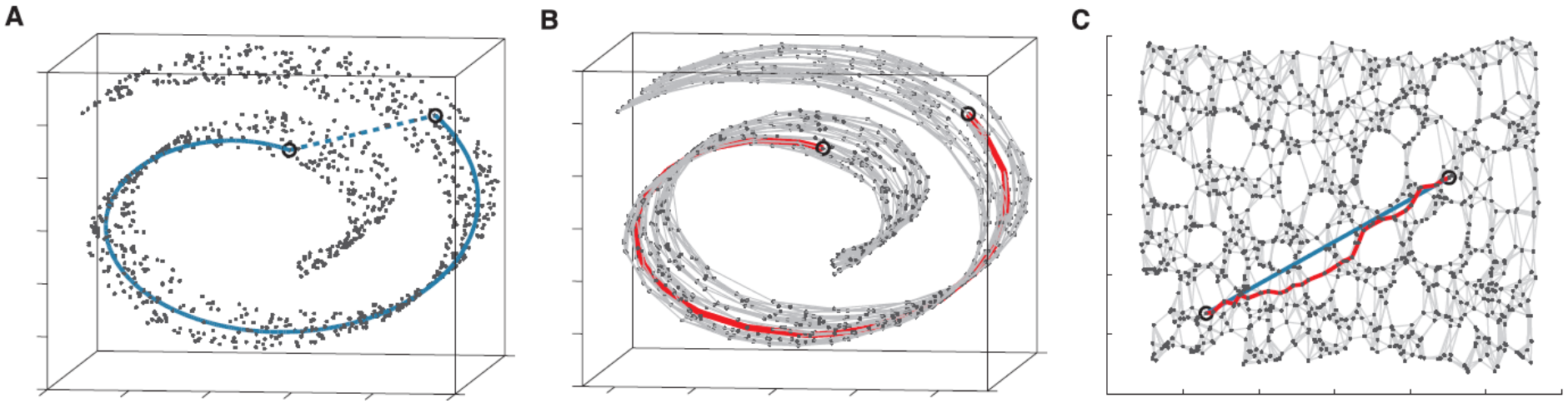



Figure 16: The "Swiss roll" data set. (A) The Euclidean distance between two points in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (B) The neighborhood graph $G$ constructed with the closest $7$ neighbors 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).

# Examples

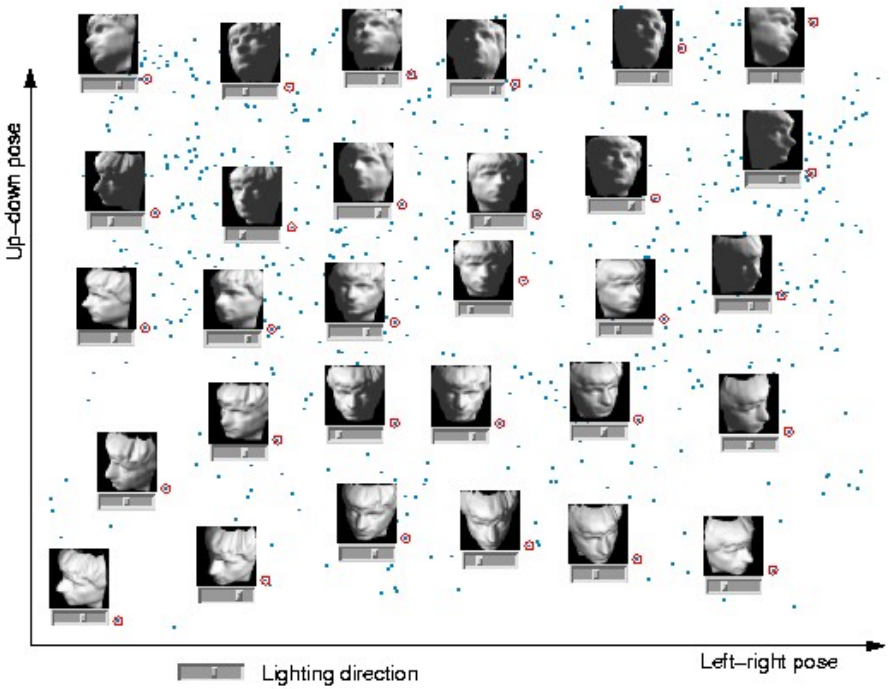



Figure 17: The input consists of $4096$-dimensional vectors, representing the brightness values of $64 \times 64$ pixel images of a face rendered with different poses and lighting directions. A two-dimensional projection is shown with horizontal sliders (under the images) representing the third dimension. 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

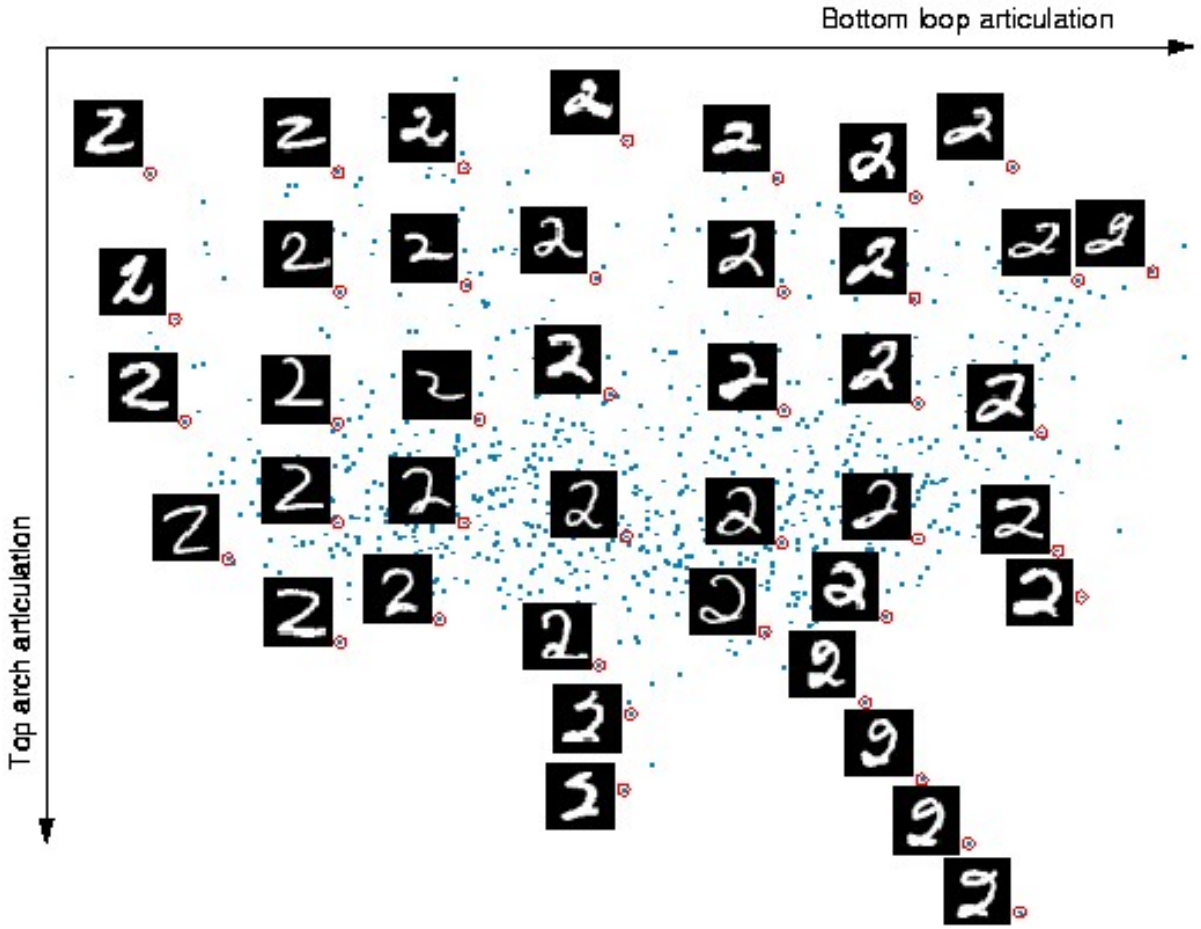



Figure 18: Isomap applied to handwritten “2”s. The two most significant dimensions in the Isomap embedding articulate the major features of the “2”: bottom loop ($x$ axis) and top arch ($y$ axis).

# 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 characterization of local geometry in the original data space is expected to be equally valid for local patches on the manifold.
- Therefore, each high-dimensional observation is mapped to a low-dimensional vector representing global internal coordinates on the manifold.

# Locally Linear Embedding

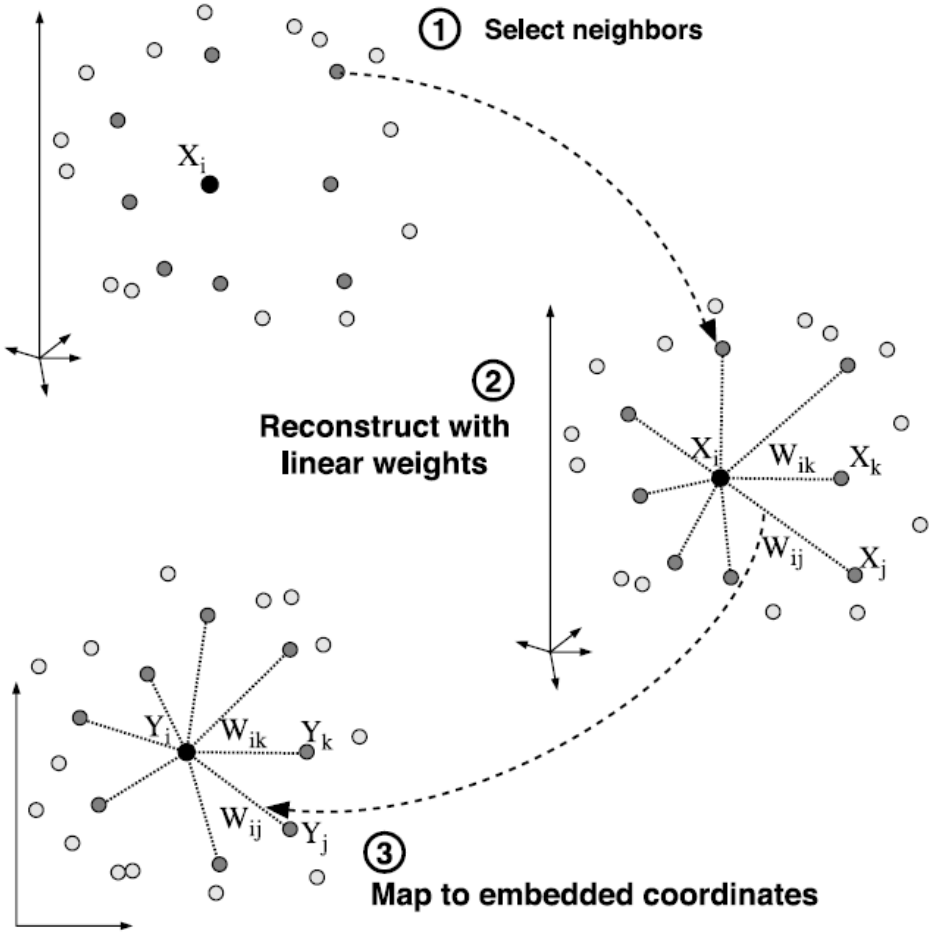



Figure 19: Steps of LLE. (1) Assign neighbors to data point $\mathbf{x}_i$. (2) Compute the weights $\mathbf{W}_{ij}$ that best reconstruct $\mathbf{x}_i$ from its neighbors. (3) Compute the low-dimensional embedding vectors $\mathbf{y}_i$ best reconstructed by $\mathbf{W}_{ij}$.

# Examples

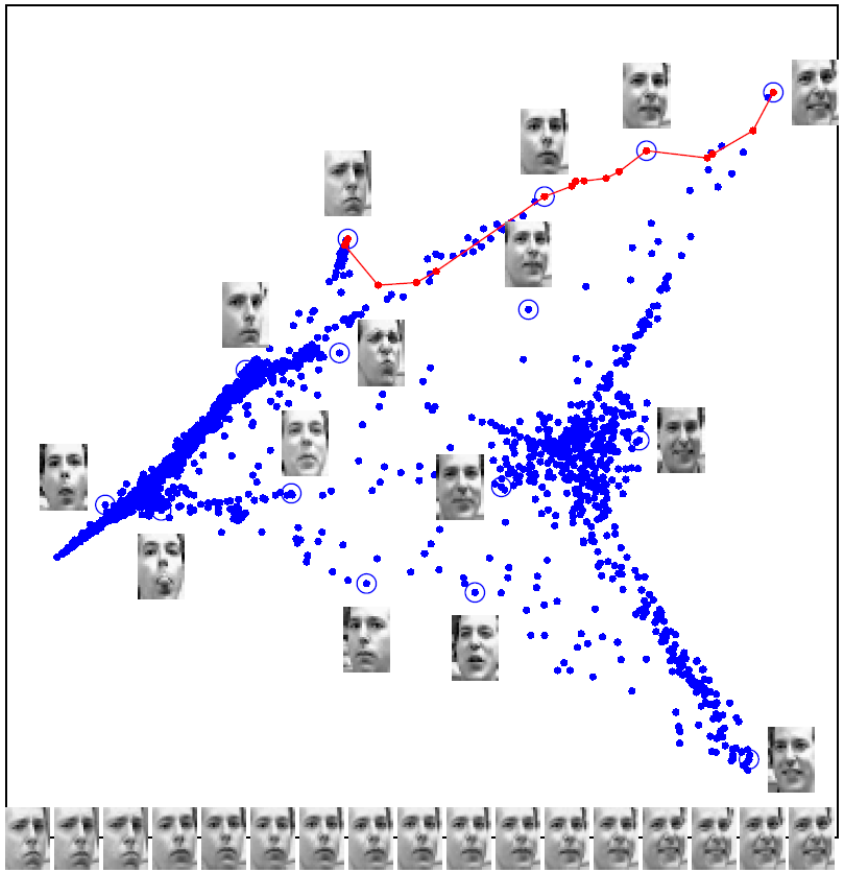


Figure 20: Images of faces, digitized at $20 \times 28$ pixels, mapped into the embedding space described by the first two coordinates of LLE. The bottom images correspond to points along the top-right path (linked by solid red line), illustrating one particular mode of variability in pose and expression.

# 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

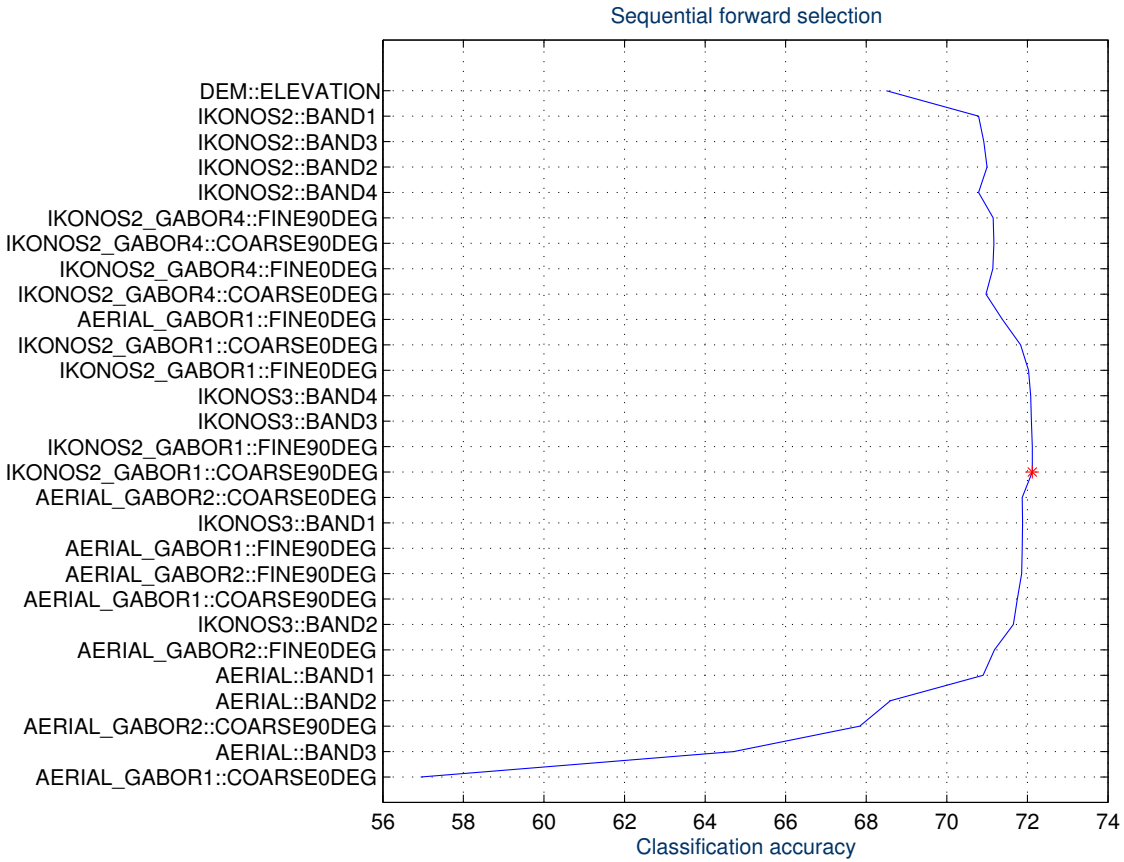



Figure 21: Results of sequential forward feature selection for classification of a satellite image using 28 features. $x$-axis shows the classification accuracy (%) and $y$-axis shows the features added at each iteration (the first iteration is at the bottom). The highest accuracy value is shown with a star.

# Examples

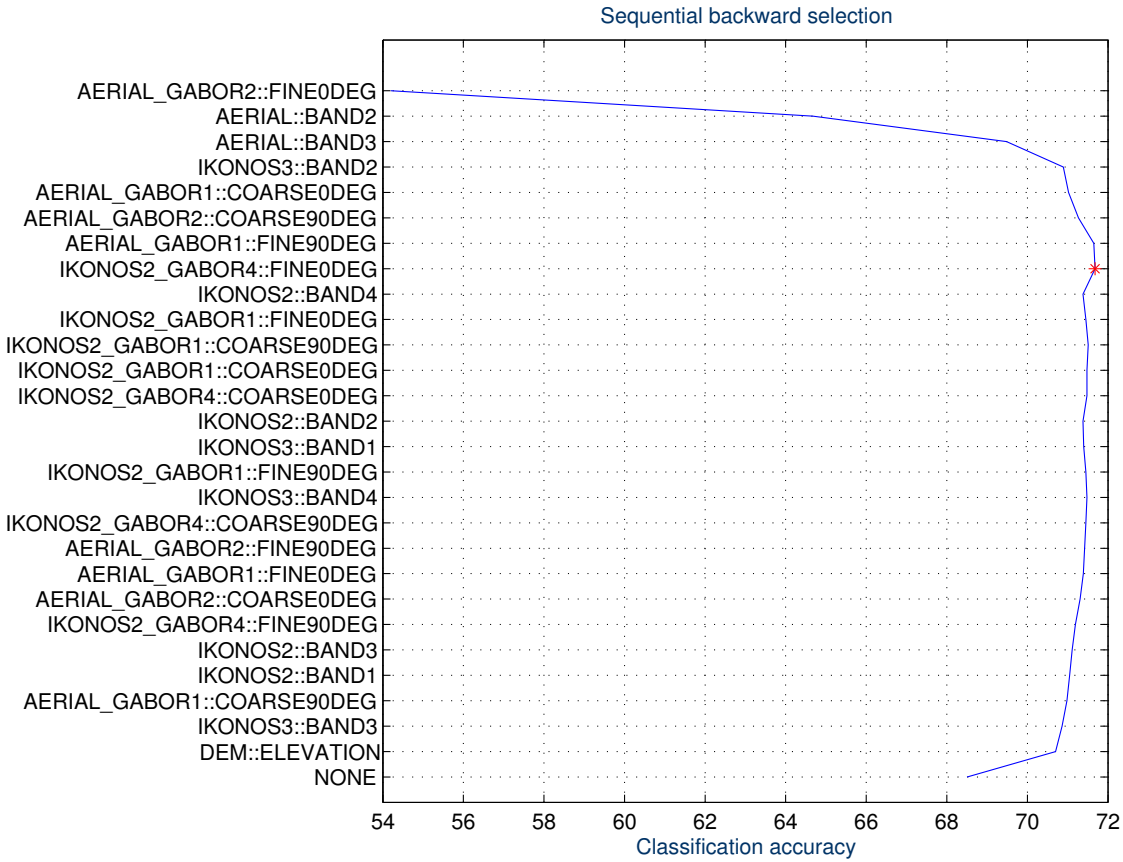



Figure 22: Results of sequential backward feature selection for classification of a satellite image using 28 features. $x$-axis shows the classification accuracy (%) and $y$-axis shows the features removed at each iteration (the first iteration is at the bottom). The highest accuracy value is shown with a star.

# 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.