Feature Reduction and Selection

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

CS 551, Spring 2005

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.
- 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 computational complexity of the classifier affected by the dimensionality?

Problems of Dimensionality

- Consider the case of two classes with multivariate Gaussian densities with the same covariance (i.e., $p(\mathbf{x}|w_j) = N(\boldsymbol{\mu_j}, \boldsymbol{\Sigma})$, $j = 1, 2$).
- The Bayes error can be computed as

$$P_e = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} e^{-u^2/2} du$$

where r^2 is the squared Mahalanobis distance

$$r^2 = (\boldsymbol{\mu_1} - \boldsymbol{\mu_2})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu_1} - \boldsymbol{\mu_2})$$

Problems of Dimensionality

- P_e approaches zero as r approaches infinity.
- Consider the special case where the features are statistically independent (i.e., $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$) where the Mahalanobis distance becomes

$$r^2 = \sum_{i=1}^{d} \left(\frac{\boldsymbol{\mu_{1i}} - \boldsymbol{\mu_{2i}}}{\sigma_i} \right)^2$$

- This shows how each feature contributes to reducing the probability of error where the most useful features have large $\boldsymbol{\mu_{1i}} - \boldsymbol{\mu_{2i}}$ relative to σ_i.
- An obvious way to reduce the error rate further is to introduce new, independent features.

Problems of Dimensionality

- In general, if the performance obtained with a given set of features is inadequate, it is natural to consider adding new features.
- If the additional features provide new information, performance will improve; otherwise, the Bayes classifier will ignore the new features (assuming the ideal case where the probabilistic structure is completely known).
- Even though increasing the number of features increases the complexity of the classifier, it may be acceptable for an improved performance.

Problems of Dimensionality

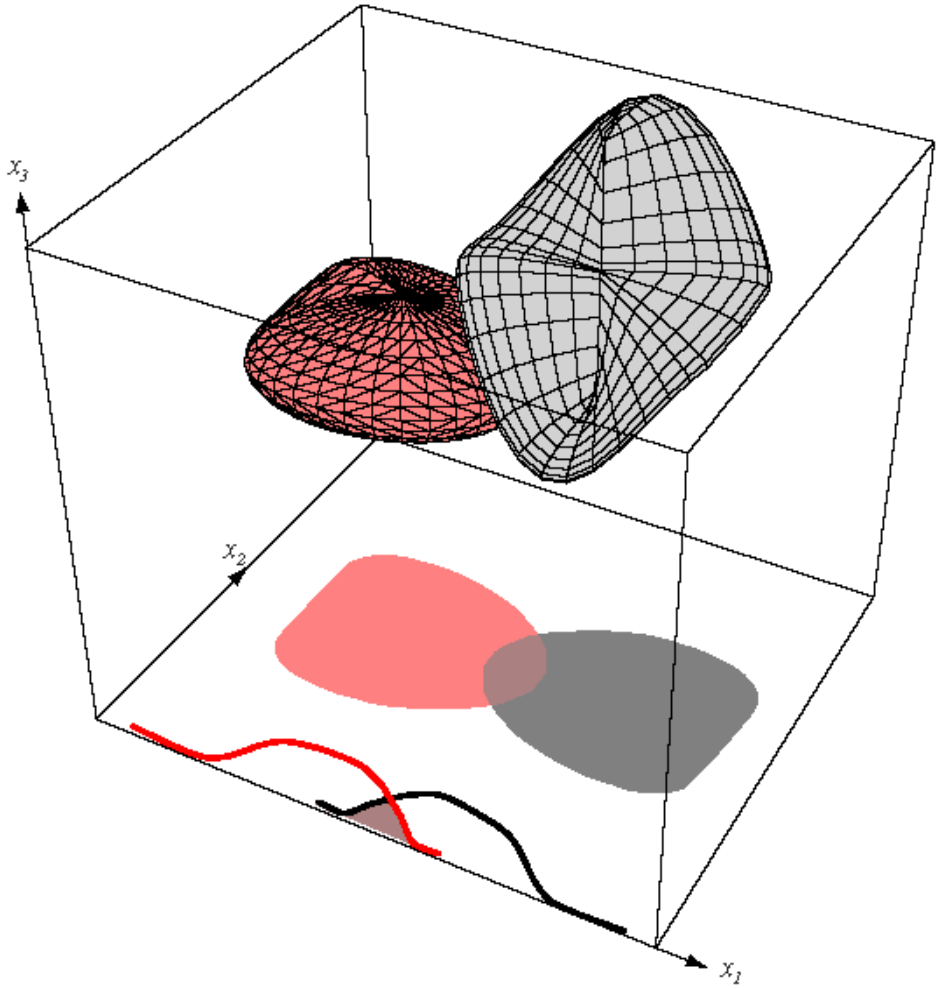


Figure 1: There is a non-zero Bayes error in the one-dimensional x_1 space or the two-dimensional x_1, x_2 space. However, the Bayes error vanishes in the x_1, x_2, x_3 space because of non-overlapping densities.

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*.
- Potential reasons include wrong assumptions in model selection or 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 prior information and a Bayes estimate
  - using heuristics such as conditional independence

Problems of Dimensionality

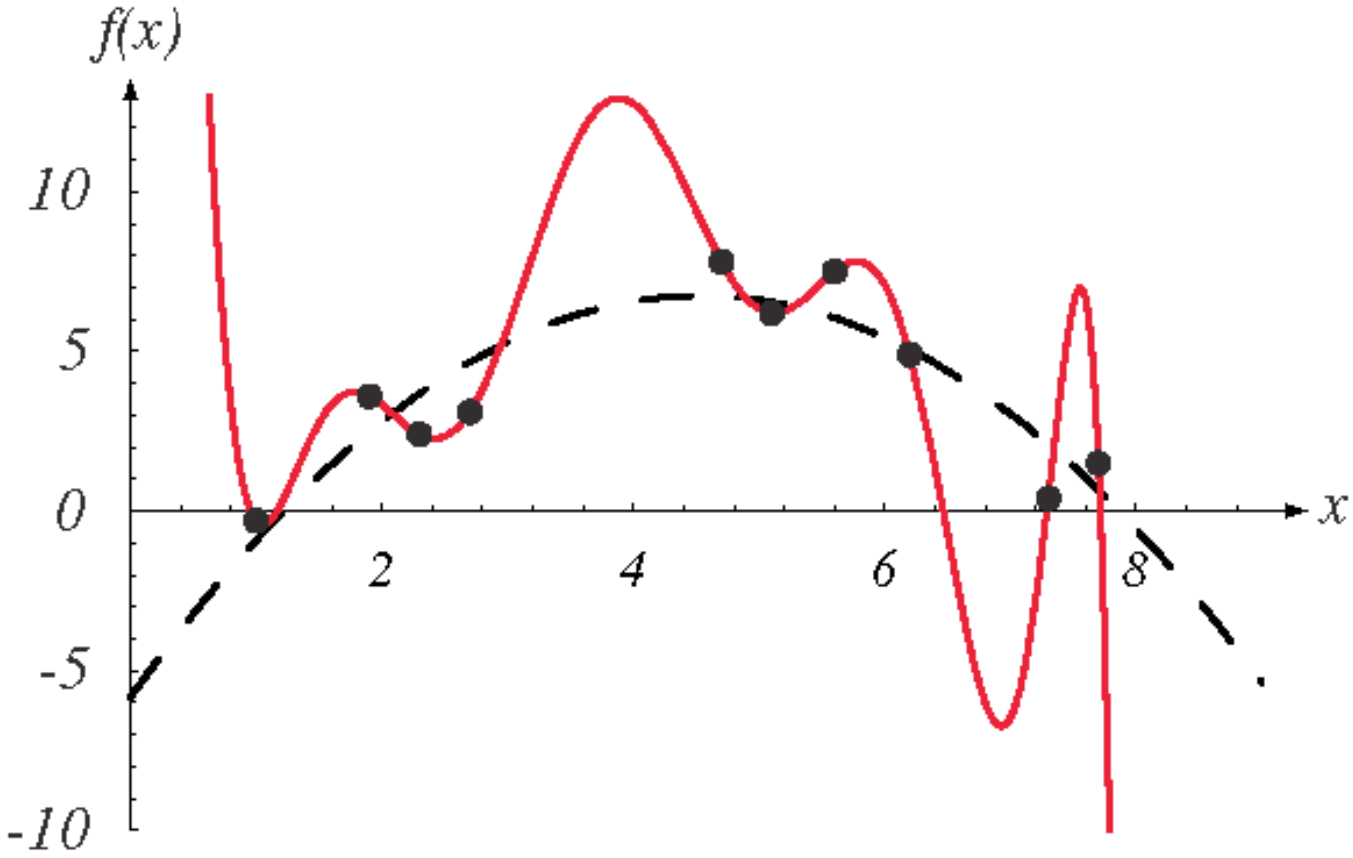


Figure 2: Problem of insufficient data is analogous to problems in curve fitting. The training data (black dots) are selected from a quadratic function plus Gaussian noise. A tenth-degree polynomial fits the data perfectly but we prefer a second-order polynomial for better generalization.

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 approach for 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)

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*

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

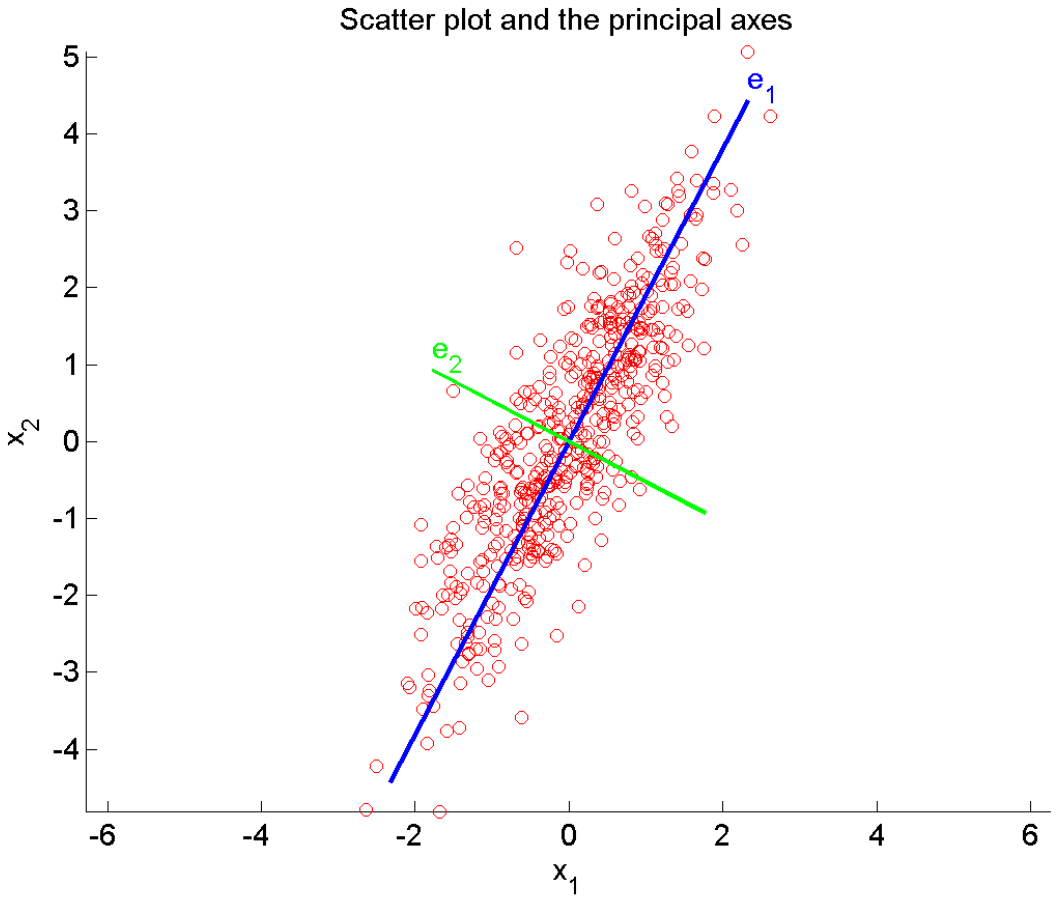
having the largest eigenvalues.

- 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.

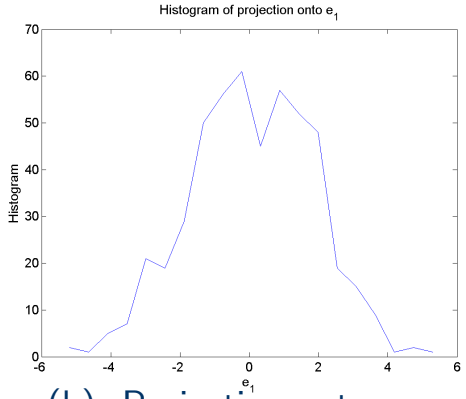
Principal Components Analysis

- 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 covariance matrix of the data.

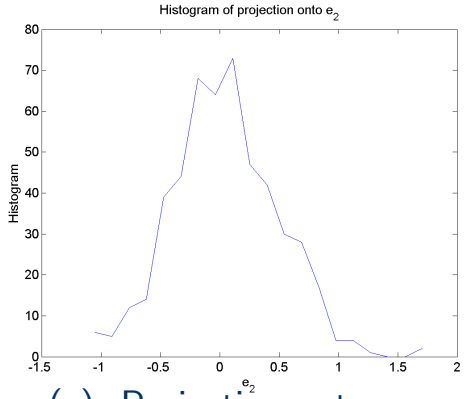
Examples



(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

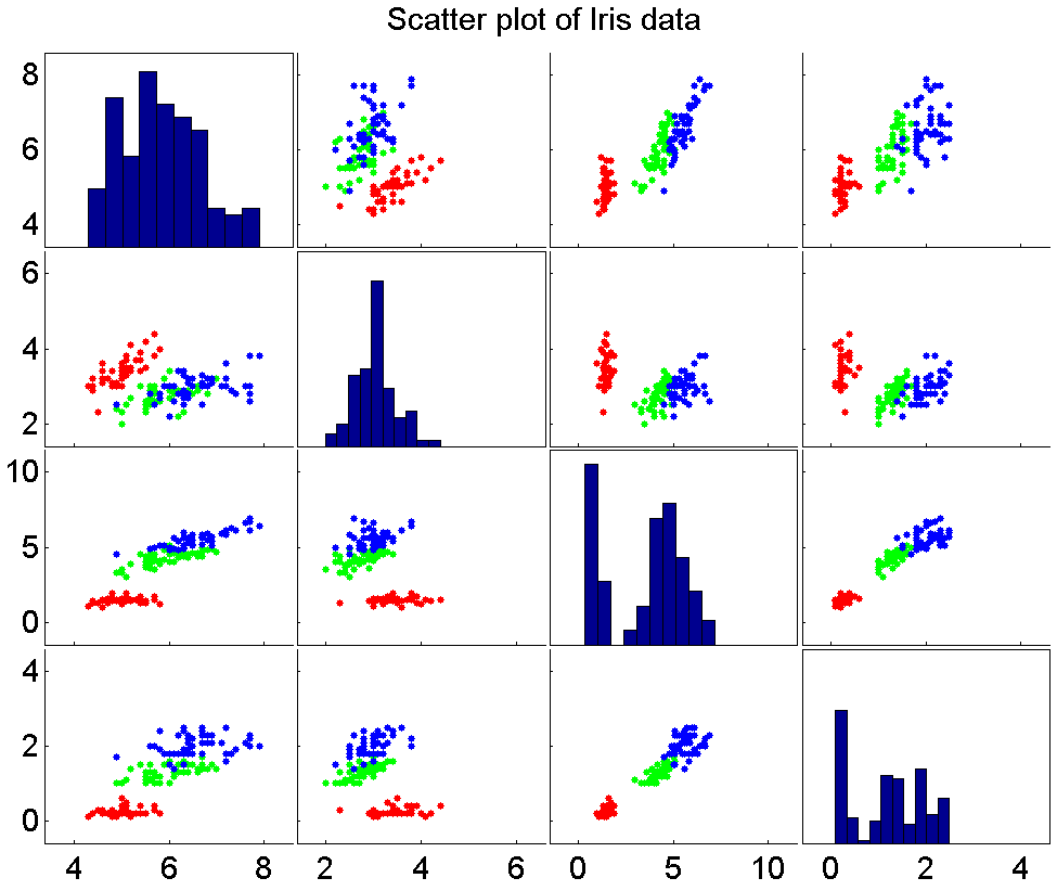


Figure 4: Scatter plot of the iris data. Diagonal cells show the histogram for each feature. Other cells show scatters of pairs of features x_1, x_2, x_3, x_4 in top-down and left-right order. Red, green and blue points represent samples for the setosa, versicolor and virginica classes, respectively.

Examples

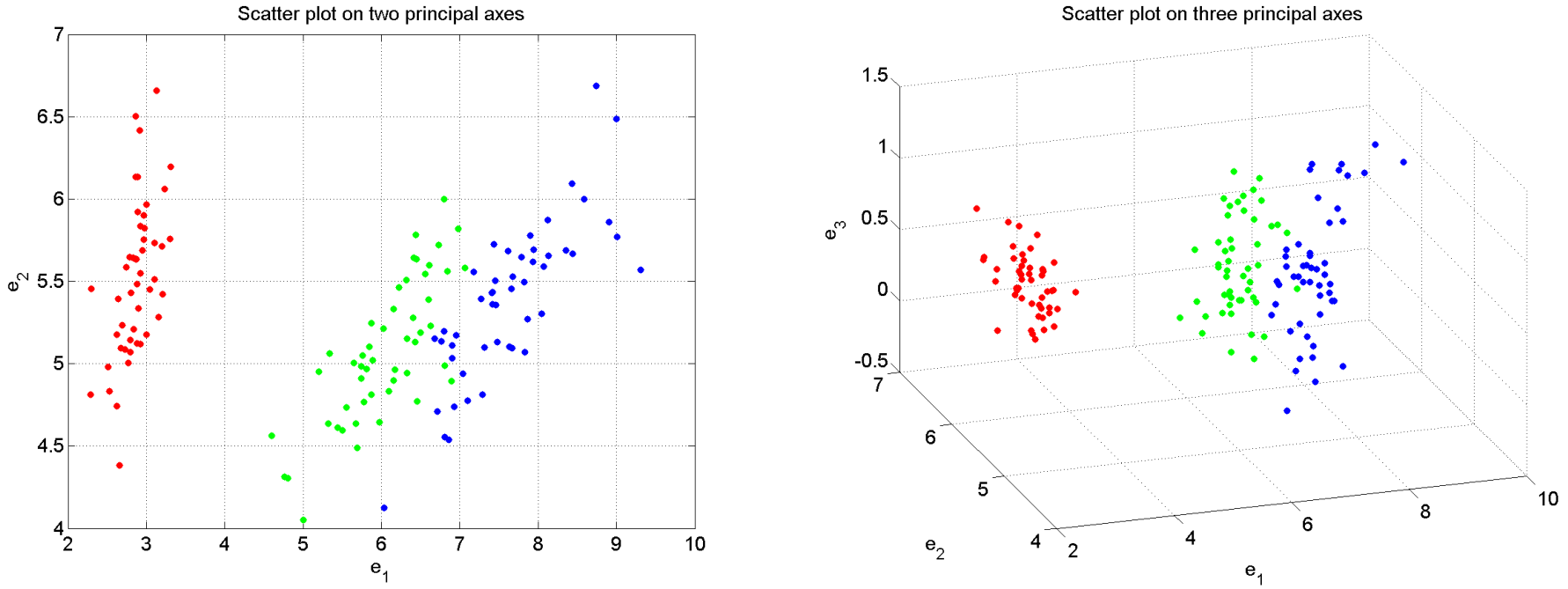


Figure 5: Scatter plot of the projection of the iris data onto the first two and the first three principal axes. Red, green and blue points represent samples for the setosa, versicolor and virginica classes, respectively.

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

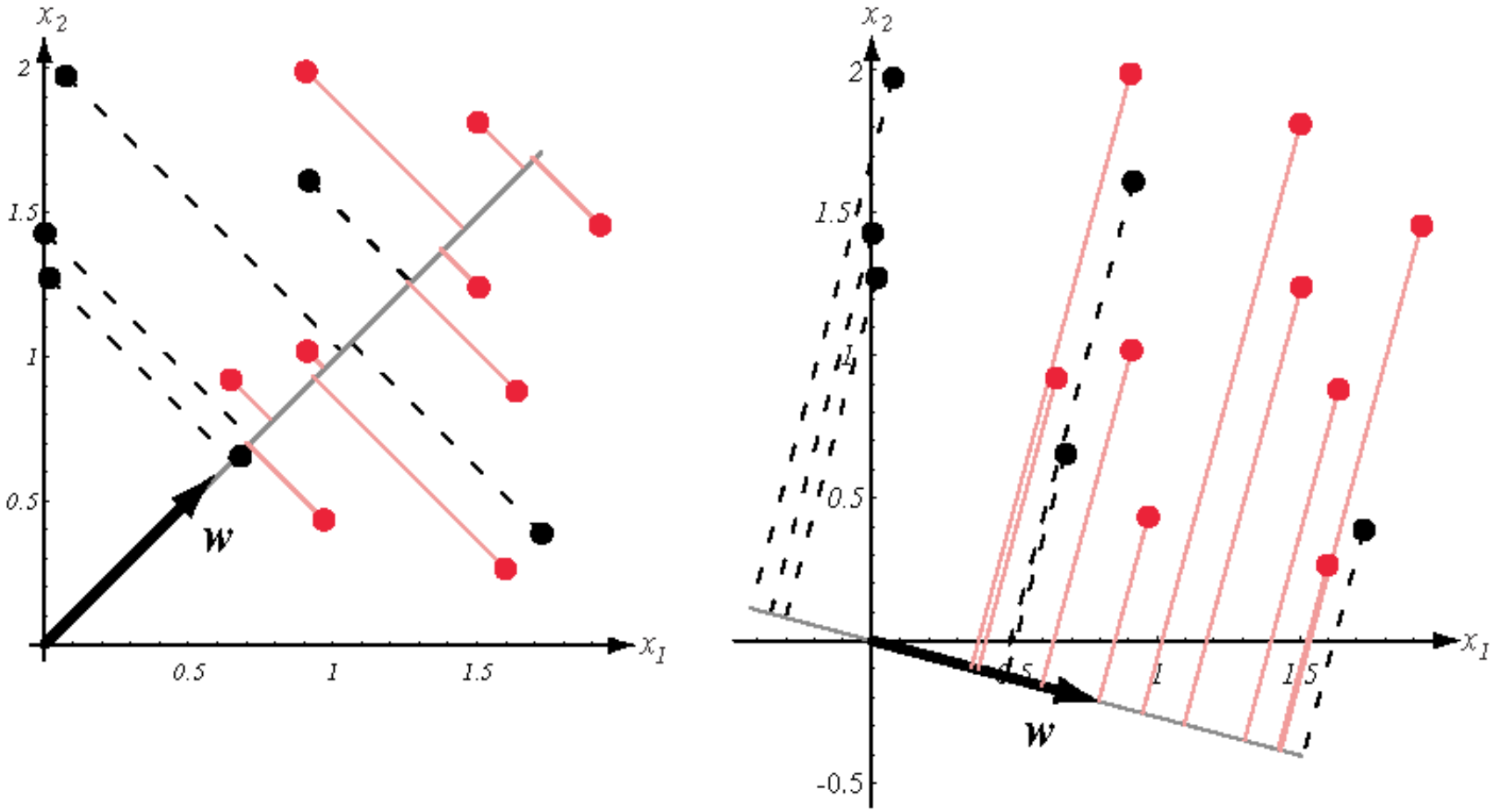


Figure 6: Projection of the same set of samples onto two different lines in the directions marked as $\mathbf{w}$. The figure on the right shows greater separation between the red and black projected points.

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 $\mathbf{S_W}$ is computed as

$$\mathbf{S_W} = \sum_{i=1}^{c} \mathbf{S}_i$$

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

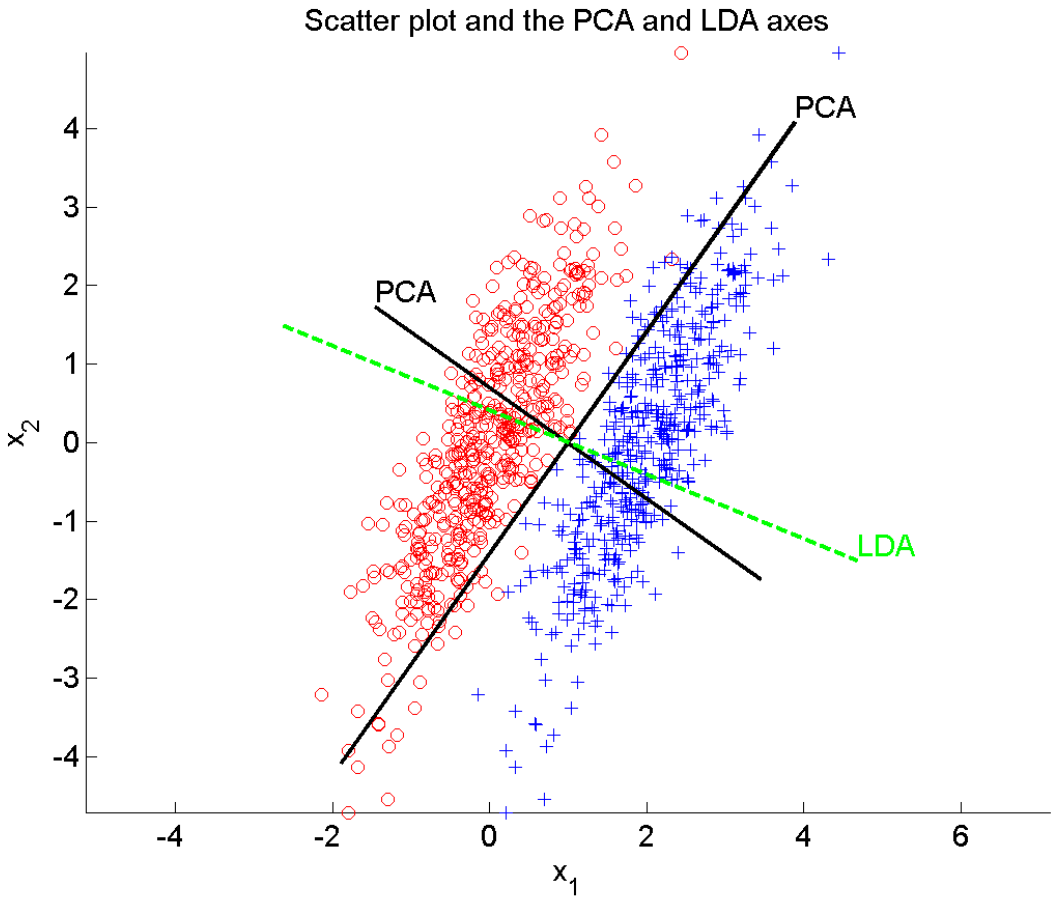
$$J(\mathbf{W}) = \frac{|\mathbf{W}^T\mathbf{S_B}\mathbf{W}|}{|\mathbf{W}^T\mathbf{S_W}\mathbf{W}|}$$

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

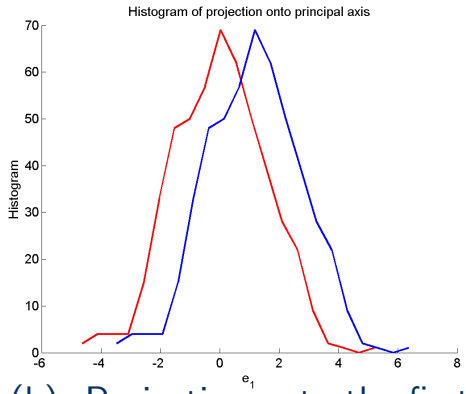
Linear Discriminant Analysis

- It can be shown that $J(\mathbf{W})$ is maximized when the columns of $\mathbf{W}$ are the eigenvectors of $\mathbf{S_W^{-1}S_B}$ having the largest eigenvalues.
- 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 that we discussed earlier can be used to train Bayesian classifiers.

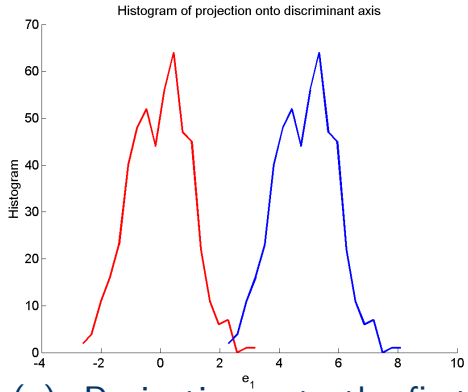
Examples



(a) Scatter plot.



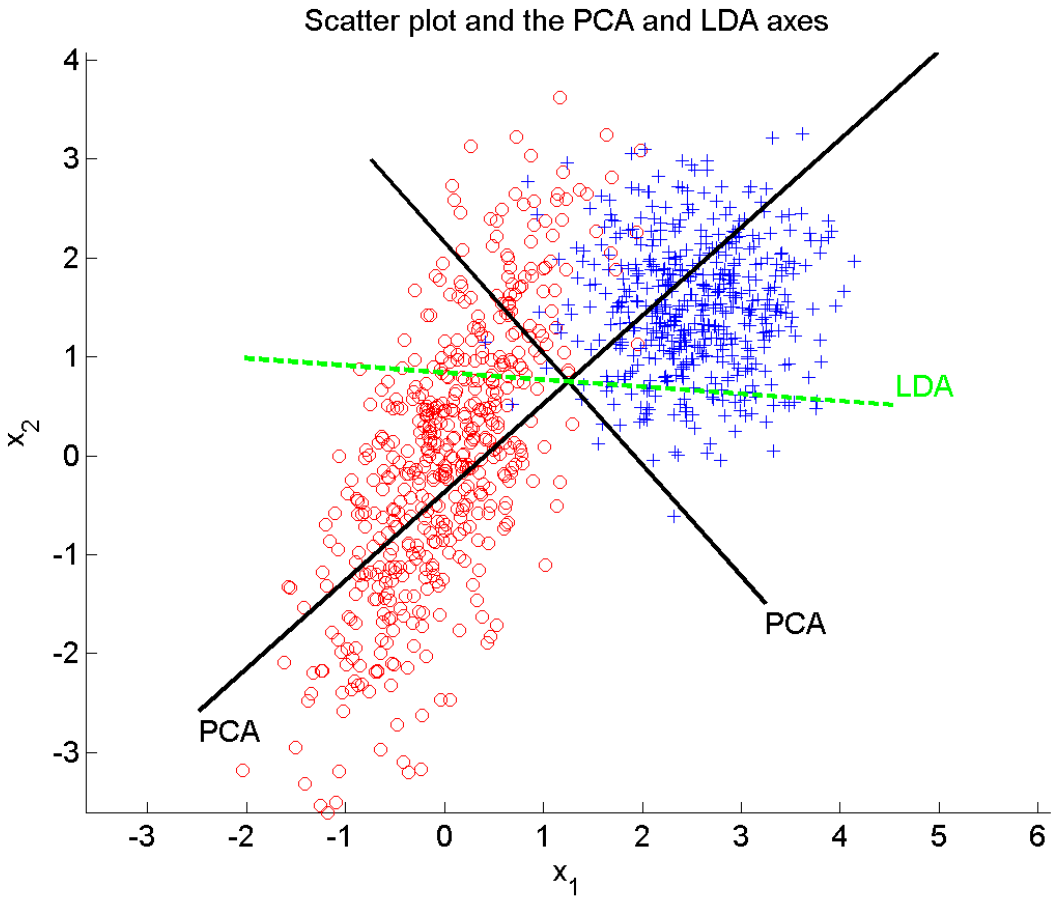
(b) Projection onto the first PCA axis.



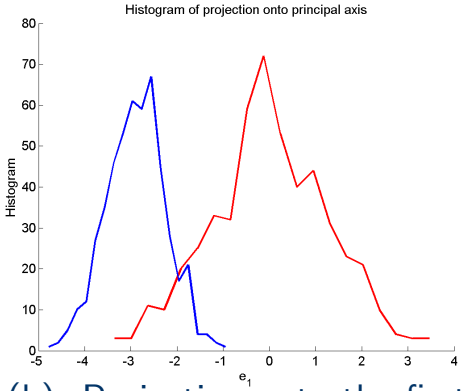
(c) Projection onto the first LDA axis.

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.

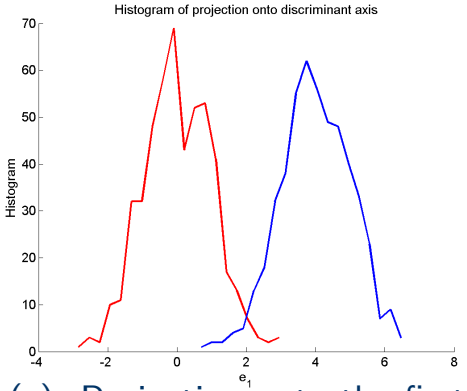
Examples



(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.

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 typically a function of the classification error.
- Note that, the use of the 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.

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.