Machine Learning: Algorithms and Applications

Advanced Multimedia Research Lab
University of Wollongong

Feature Selection and Extraction

Outline of Topics

Introduction

- Dimensionality reduction can be achieved in two ways:
  - Identify those variables or features that do not contribute to the classification task. In essence select *d* features out of available *p* features. This is also referred to as feature selection in the measurement space or feature selection.
  - Transform from the *p*-dimensional feature space to a lower dimensional space. This is sometimes referred to as feature selection in the transformed domain or feature extraction

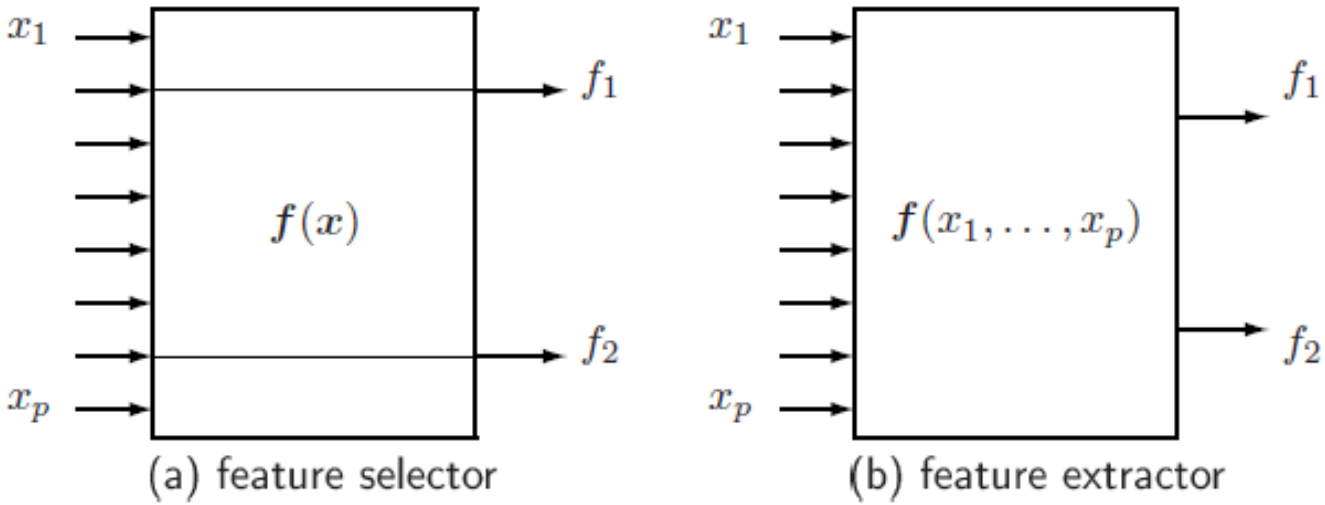


Figure 10.1 Dimensionality reduction by (a) feature selection and (b) feature extraction.

Introduction

- Dimensionality reduction can be achieved in two ways:
  - Identify those variables or features that do not contribute to the classification task. In essence select d features out of available p features. This is also referred to as feature selection in the measurement space or feature selection.
  - Transform from the p-dimensional feature space to a lower dimensional space. This is sometimes referred to as feature selection in the transformed domain or feature extraction
- Feature selection optimises over the set all possible subsets of size d, Ω_d of the p possible features. It seek subset, $\tilde{X}_d$ for which.

$$J(\tilde{X}_d) = \max_{X \in \Omega_d} J(X)$$

- Feature extraction optimises over all possible transformations of the variables and seek the transformation $\tilde{A}$ for which

$$J(\tilde{A}) = \max_{A \in \mathcal{A}} J(A(x))$$

Notation

- Σ is the population covariance matrix. Its maximum likelihood estimate is $\hat{\Sigma}$
- Σ_i is the covariance matrix of class ω_i and its maximum likelihood estimate is $\hat{\Sigma}_i$.
- The maximum likelihood estimates are computed from data as:

$$\hat{\Sigma}_i = \frac{1}{n_i} \sum_{j=1}^{n} z_{ij}(x_j - m_i)(x_j - m_i)^t$$

$$\hat{\Sigma} = \frac{1}{n_i} \sum_{j=1}^{n} (x_j - m_i)(x_j - m_i)^t$$

where

$$z_{ij} = \begin{cases} 1 & \text{if } x_j \in \omega_i \\ 0 & \text{otherwise} \end{cases}$$

and $n_i = \sum_{j=1}^{n} z_{ij}$

Notation

- m_i is the sample mean of class ω_i and it is given by

$$m_i = \frac{1}{n_i} \sum_{j=1}^{n} z_{ij} x_j$$

and m is the sample mean

$$m = \frac{1}{n_i} \sum_{j=1}^{C} \frac{n_i}{n} m_i$$

- The unbiased estimate of the covariance matrix is $\frac{n}{n-1}\hat{\Sigma}$.

Notation

- Denote the within-class scatter matrix (or pooled within-class sample covariance matrix) as

$$S_W = \sum_{i=1}^{C} \frac{n_i}{n} \hat{\Sigma}_i$$

and the unbiased estimate, $S = \frac{n}{n-C} S_W$

- Denote the sample between-class covariance matrix by, S_B,

$$S_B = \sum_{i=1}^{C} \frac{n_i}{n} (m_i - m)(m_i - m)^t$$

- We note that

$$S_W + S_B = \hat{\Sigma}$$

Convince yourself by showing this!

General ideas: Feature selection

Problem definition

The problem of feature selection is defined as follows: given a set of p features, select a subset of size d that leads to the smallest classification error.

- The solution is simple but could be intractable: evaluate the optimality criterion, J, for all combinations of the d variables selected from p and select the combination for which the criterion is maximum.
- The number of subsets to evaluate is

$$\binom{p}{d} = \frac{p!}{(p-d)!d!}$$

General ideas: Feature selection

- The feature selection method could be categorised in terms of the search method used:

  Optimal methods can lead to globally optimal solution but are computationally expensive - exhaustive search ; accelerated search; Monte Carlo methods.

  Suboptimal methods attempt to achieve a trade off between optimality and computational speed.

- Feature selection methods can also be categorised in terms of the criteria used:

  Filter methods select features independent of the classification algorithm.

  Wrapper methods select features based on the classification algorithm.

General ideas: Feature selection

- Filter methods select features independent of the classification algorithm. Essentially, this group of methods estimate the overlap between (assumed) data distribution and favour those features for which overlap is minimum.

- One of the methods used to measure separability of distribution is the probabilistic distance between two distribution $p(x|\omega_1)$ and $p(x|\omega_2)$. Example is the divergence or Kullback-Leibler distance

$$J_D(\omega_1, \omega_2) = \int [p(x|\omega_1) - p(x|\omega_2)] \log \left\{ \frac{p(x|\omega_1)}{p(x|\omega_1)} \right\} dx$$

- The separability measures could be computed recursively by constructing the feature set at stage k from that obtained in stage $k-1$ through the addition or subtraction of a number of features to the current set.

General ideas: Feature selection

- Wrapper methods select features based on the classification algorithm. The criterion used in these methods is the classification error rate.
- Error estimation is based on a separate test set; or techniques such as bootstrap and jackknife are used when there is insufficient data.
- Any of the methods we discussed under classifier performance evaluation can be adapted for use.

General ideas: Search algorithm

- Basic idea is to build up a set of d features incrementally:
  - bottom-up: starting with empty set, if X_k represents a set of k features, the best set at a given iteration, $\hat{X}$, is the set for which the feature selection criterion is maximum

$$J(\hat{X}) = \max_{X \in \Omega} J(X) \tag{1}$$

    The set Ω of all sets of features is determined from the set at the previous iteration.

  - top-down: starting with the full set of features features are removed and the remaining sets tested against the criterion.

Optimal search: Branch and Bound

- This method assumes the monotonicity property that says, for two subsets of features (or variables), X and Y,

$$X \subset Y \Rightarrow J(X) < J(Y) \tag{2}$$

Branch and bound procedure[AW]

1: Build a tree of the possible selections.
2: Start at the top level and proceed down the rightmost branch evaluating the cost function J at each node.
3: If J is less than the current threshold, abandon further search down the particular branch and backtrack to the previous branching node.
4: Continue search down the rightmost branch
5: If, on the search any branch the bottom level is reached, then if the value of J for this level is larger than current threshold, update the threshold and begin backtracking.

Optimal search: Branch and Bound

We shall describe the method by way of example (Figure 10.6). Let us assume that we wish to find the best three variables out of a set of five. A tree is constructed whose nodes represent all possible subsets of cardinality 3, 4, and 5 of the total set as follows. Level 0 in the tree contains a single node representing the total set. Level 1 contains subsets of the total set with one variable removed and level 2 contains subsets of the total set with two variables removed. The numbers to the right of each node in the tree represent a subset of variables. The number to the left represents the variable that has been removed from the subset of the parent node in order to arrive at a subset for the child node. Level 2 contains all possible subsets of five variables of size three. Note that the tree is not symmetrical. This is because removing variables 4 then 5 from the original set (to give the subset $\{1,2,3\}$) has the same result as removing variable 5 then variable 4. Therefore, in order for the subsets not to be replicated, we have only allowed variables to be removed in increasing order. This removes unnecessary repetitions in the calculation.

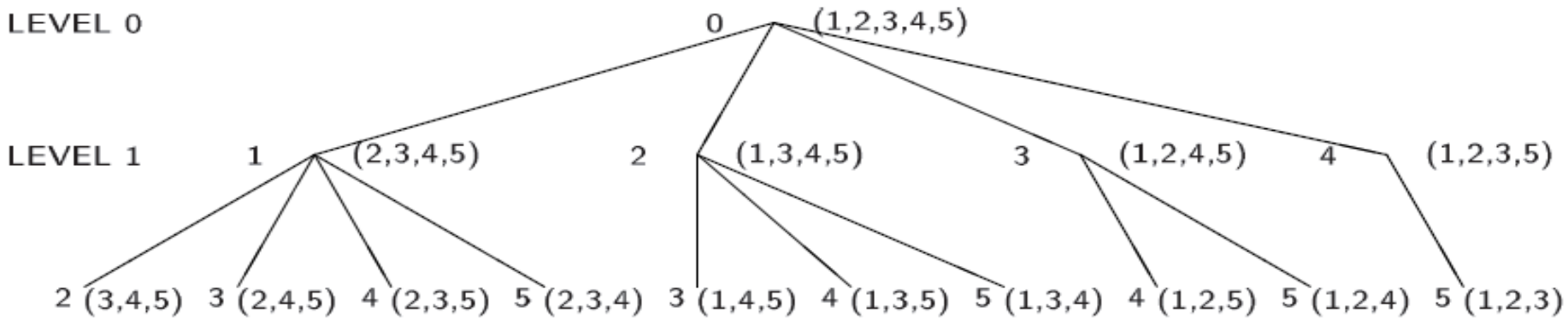


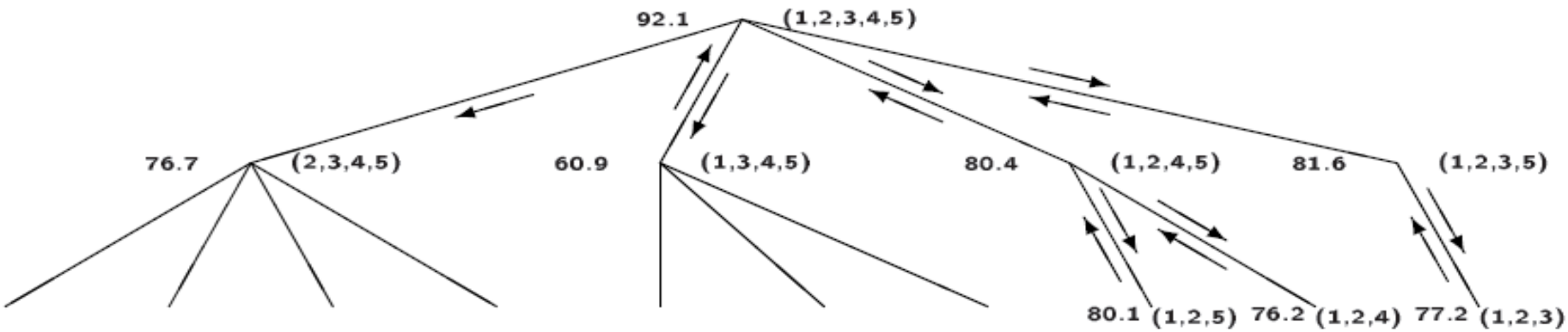
Figure 10.6 Tree figure for branch and bound method.

Optimal search: Branch and Bound

Now we have obtained our tree structure, how are we going to use it? The tree is searched from the least dense part to the part with the most branches (right to left in Figure 10.6).

Figure 10.7 gives a tree structure with values of the criterion J printed at the nodes. Starting at the rightmost set (the set $\{1,2,3\}$ with a value of $J = 77.2$), the search backtracks to the nearest branching node and proceeds down the rightmost branch evaluating $J(\{1, 2, 4, 5\})$. Since $J(\{1, 2, 4, 5\})$ is larger than $J(\{1, 2, 3\})$, the monotonicity property that all subsets of this set will yield a lower value of the criterion function cannot be utilised to discard the branch and so the search then proceeds down the rightmost branch to $J(\{1, 2, 4\})$, which gives a lower value than the current maximum value of J, J^*, and so is discarded. The set $J(\{1, 2, 5\})$ is next evaluated and retained as the current best value (largest on a subset of three variables), $J^* = 80.1$. $J(\{1, 3, 4, 5\})$ is evaluated next, and since this is less than the current best, the search of the section of the tree originating from this node is not performed. This is because we know from the monotonicity property that all subsets of this set will yield a lower value of the criterion function. The algorithm then backtracks to the nearest branching node and proceeds down the next rightmost branch (in this case, the final branch). $J(\{2, 3, 4, 5\})$ is evaluated, and again since this is lower than the current best value on a subset of three variables, the remaining part of the tree is not evaluated.

Thus, although not all subsets of size 3 are evaluated, the algorithm is optimal since we know by the condition (10.9) that those not evaluated will yield a lower value of J.



Suboptimal methods

Best individual *N*: Assign discrimination power estimate to each feature in the original set Ω. The features are ordered as,

$$J(x_1) \geq J(x_2) \geq \cdots \geq J(x_p)$$

and we select as the best set of d features those with best individual scores.

- Computationally simple but not likely to lead to an optimal selection.
- Could yield good result if the features in the set are uncorrelated.
- Does not take advantage of the possible multivariate relationships.

Suboptimal methods

Sequential forward selection (SFS): Set addition and bottom-up search method that adds new features to the set, one at a time. The process is as follows:

SFS procedure

1: Let the set of features at some iteration be X_{d1}. Usually starts with empty set.
2: For each of the features, ζ not yet selected, calculate the criterion function $J_k = J(X_{d1} + \zeta_k)$
3: Add the feature that yields the maximum J to the set X_{d1}.
4: Continue the process until the number of desired features is obtained.

- The main disadvantage is that there is no mechanism of deleting features once they are added to the set.
- The user must select the desired number of features.

Suboptimal methods

Generalised Sequential forward selection (GSFS): The algorithm adds r features at a time to the set rather than one. The process is as follows:

SFS procedure

1: Let the set of features at some iteration be X_{d1}. Usually starts with empty set.
2: From all the remaining features, $p - d_1$, generate all possible subsets Y_r, of size r and calculate the cost function, $J_k = J(X_{d1} + Y_r)$.
3: Add the feature set Y_r, that yields the maximum J to the set X_{d1}.
4: Continue the process until the number of desired features is obtained.

- The disadvantage of not being able to delete features once they are added remains.
- The user must select the desired number of features.
- There is the advantage that the multivariate dependency that may exist is being exploited.

Suboptimal methods

Sequential backward selection (SBS): The algorithm deletes one feature at a time until d features remain. It is top-down search method and proceeds as follows:

SBS procedure

1: Start with the complete set of features.
2: Delete from set the feature ζ_j for which $J(\Omega - \zeta_j)$ is the largest.
3: The new set is $\{\Omega - \zeta_j\}$.
4: Continue the process until the number of desired features is obtained.

- The disadvantage over the SFS method is that there are more computations.
- The user must select the desired number of features.

Generalised sequential backward selection (GSBS): The algorithm deletes a set of features at a time until d features remain.

Suboptimal methods

Plus l - take away r selection: This method incorporates backtracking in the feature selection. It could be bottom-up or top-down. If $l > r$ it is bottom-up and l features are added to the current set using SFS and the worst r features are removed using SBS.

Generalised plus l - take away r selection: The generalised form uses GSFS and GSBS procedures to add and remove sets of features.

- l and r may be decomposed so that $l_i, i = 1, \ldots, n_l$ and $r_j, j = 1, \ldots, n_r$; n_l and n_r are the number of components.

$$0 \leq l_i \leq l \quad \text{and} \quad 0 \leq r_j \leq r$$

$$\sum_{i=1}^{n_l} l_i = l \quad \text{and} \quad \sum_{j=1}^{n_r} r_j = r$$

- Features are added (GSFS(l_i)) in n_l steps by adding l_i features at each increment. Deletion achieved by applying GSBS(r_j), $j = 1, , \ldots, n_r$ successively.

Suboptimal methods

Floating search methods: If we allow the values of l and r to vary from stage to stage we obtain the "floating" method. Suppose we have at stage k, a set of subsets $X_1, \ldots, X_k$ of sizes 1 to k respectively. The corresponding values of the criterion are J_1 to J_k; $J_i = J(X_i)$

Floating search method

1: Select feature x_j from $\Omega - X_k$ that increases the value of J the greatest and add it to the current set $X_{k+1} = X_k + x_j$
2: Find the feature, x_r, in the current set, X_{k+1} that reduces the value of J the least.
3: If this feature is the same as x_j, set $J_{k+1} = J(X_{k+1})$; increment k an go to step 1; otherwise remove it from the set and form $X'_k = X_{k+1} - x_r$
4: Continue removing features from the set X'_k to form reduced sets X'_{k-1}, while $J(X'_{k-1}) > J_{k-1}$; $k = k - 1$; $ork = 2$ then continue with step 1.

General considerations: Feature extraction

- Feature extraction entails the transformation of a given data (using all the variables) to a data set with a reduced number of variables.
- Transformation could be linear or non-linear.
- The methods to be described are also referred to as feature selection in the transformed space.
- They could be supervised or unsupervised.
- They could be based on optimisation of class separability measure.

Principal component analysis

表 1

学生编号	语文	数学	物理	化学
1	90	140	99	100
2	90	97	88	92
3	90	110	79	83
. . .	. . .	. . .	. . .	. . .

Principal component analysis

表 2

学生编号	数学	物理	化学	语文	历史	英语
1	65	61	72	84	81	79
2	77	77	76	64	70	55
3	67	63	49	65	67	57
4	80	69	75	74	74	63
5	74	70	80	84	82	74
6	78	84	75	62	72	64
7	66	71	67	52	65	57
8	77	71	57	72	86	71
9	83	100	79	41	67	50
. . .	. . .	. . .	. . .	. . .	. . .	. . .

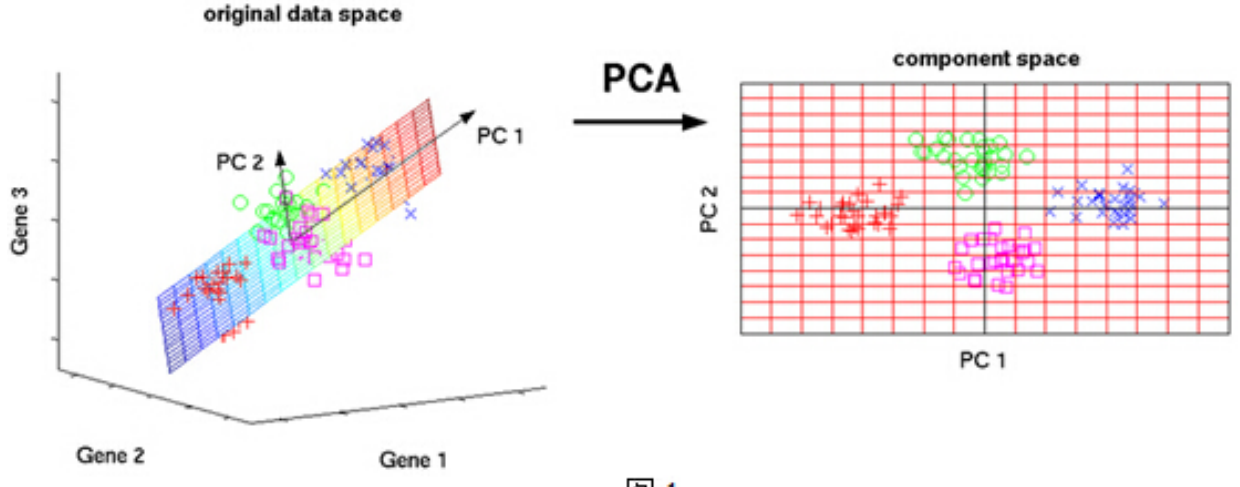


图 1

Principal component analysis

- The essence of principal component analysis (PCA) is to derive new variables (in order of importance) that are linear combinations of the original variables and are uncorrelated.
- The linear combination of the original variables is such that we achieve maximal variance.
- The new variables do not necessarily have physical interpretation.
- PCA is variable-directed and does not make any assumption about the existence or otherwise of groupings or cluster within the data. It is an unsupervised feature extraction method.

Principal component analysis

- Geometrically, the principal component analysis produces a single line of best fit when given the problem of fitting a line to a bi-variate data X and Y; the solution minimises the sum of squares of the perpendicular distance from the sample points to the line.
- The variable defined by the line of best fit is the first principal component. The second principal component is the variable defined by the line that is orthogonal to the first.

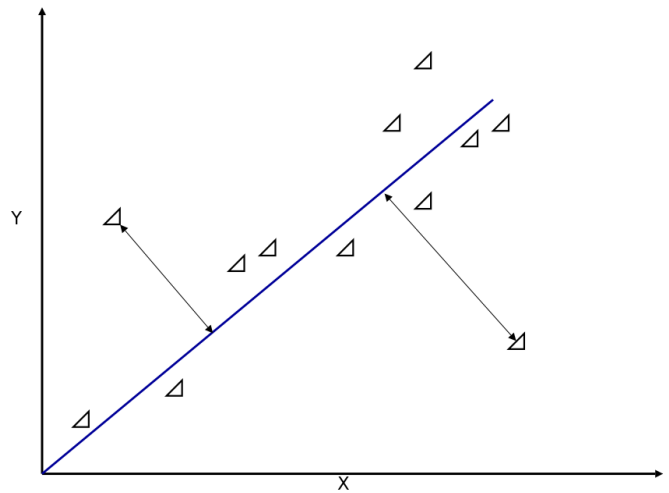


Figure: Principal component line of best fit

Principal component analysis

- If the data were projected onto the first principal axis (vector representing first PC), the variation in the direction of the first principal component is proportional to the sum of squares of the distances from the second principal axis.
- Similarly the variation along the second principal axis is proportional to the sum of squares of the distances from the first principal axis.
- Total sum of squares is constant and minimising the sum of squares from a given line is equivalent to maximising the sum of squares from its perpendicular.
- The principal axes are the principal components while the variances are the principal values.

Principal component analysis

- Let $x_1, \ldots, x_p$ be the set of original variables (or features). Further, let $\xi_i, i = 1, \ldots, p$ be linear combinations of the variables

$$\xi_i = \sum_{j=1}^{p} a_{ij} x_j$$

or

$$\xi = \mathbf{A}^t \mathbf{x}$$

where ξ and $\mathbf{x}$ are vectors of random variables and $\mathbf{A}$ is the matrix of coefficients.

- We seek the orthogonal transformation $\mathbf{A}$ yielding new variables ξ_j that have stationary values of their variance. In other word maximise variance.

Principal component analysis

- We start with the first variable, ξ_1, and write

$$\xi_1 = \sum_{j=1}^{p} a_{1j} x_j$$

- Choose the coefficients $\mathbf{a}_1 = \begin{bmatrix} a_{11} & a_{12}, \ldots, a_{1p} \end{bmatrix}^t$ to maximize the variance of ξ_1 subject to the constraint $\mathbf{a}_1^t \mathbf{a}_1 = |\mathbf{a}_1| = 1$
- Compute the variance of ξ_1 as,

$$\begin{aligned} var(\xi_1) &= E[\xi_1^2] - E[\xi_1]^2 \\ &= E[a_1^t x x^t a_1] - E[a_1^t x] E[x^t a_1] \\ &= a_1^t (E[x x^t] - E[x] E[x^t]) a_1 \\ &= a_1^t \mathbf{S} a_1 \end{aligned}$$

- $\mathbf{S}$ is the covariance of the matrix of $\mathbf{x}$ and $E[\cdot]$ denotes expectation. Note that we have used $\mathbf{S}$ rather than Σ to denote variance. This avoids confusion with summation sign.

Principal component analysis

- We use the method of Lagrange multiplier to find the stationary value of the variance $a_1^t \mathbf{S} a_1$ subject to $a_1^t a = 1$

$$f(a_1) = a_1^t \mathbf{S} a_1 - \nu a_1^t a_1$$

  ν is the Lagrange multiplier.

- Differentiate with respect to the components of a_1 and equate to zero,

$$\mathbf{S} a_1 - \nu a_1 = 0$$

- Non-trivial solution requires a_1 to be an eigenvector of $\mathbf{S}$ and ν an eigenvalue.
- Variance of ξ_1 is $a_1^t \mathbf{S} a_1$

$$a_1^t \mathbf{S} a_1 = \nu a_1^t a_1 = \nu$$

  and to maximize this variance choose ν to be the largest eigenvalue and ξ_1 the corresponding eigenvector.

Principal component analysis

- Other principal components can be found similarly with appropriate constraints.
- The second principal component is the eigenvector corresponding to the second largest eigenvalue and it is orthogonal to the first principal component.
- The principal components are:

$$\xi = \mathbf{A}^t \mathbf{x}$$

$\mathbf{A} = \begin{bmatrix} a_1, & \ldots, a_p \end{bmatrix}$ is the matrix whose columns are the eigenvectors of $\mathbf{S}$

- Sum of the variances of the principal components is

$$\sum_{i=1}^{p} var(\xi_i) = \sum_{i=1}^{p} \lambda_i$$

sum of the eigenvalues of the data covariance matrix $\mathbf{S}$ and this is the total variance of the original variables.

Principal component analysis

- The first k principal components account for

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

of the total variance.

- Reducing the dimension of the original variable (or feature vector) is usually done subject to the retained components accounting for some fraction, d, of the total variance. So choose k such that,

$$\sum_{i=1}^{k} \lambda_i \geq d \sum_{i=1}^{p} \lambda_i \geq \sum_{i=1}^{k-1} \lambda_i$$

- Transformed data of reduced dimension is

$$\xi_k = \mathbf{A}_k^t \mathbf{x}$$

where $\mathbf{A}_k = \begin{bmatrix} a_1, & \ldots, a_k \end{bmatrix}$ is an $p \times k$ matrix and $\xi_k = \begin{bmatrix} \xi_1, & \ldots, \xi_k \end{bmatrix}^t$

Principal component analysis

An alternative approach is to examine the eigenvalue spectrum and see if there is a point where the values fall sharply before levelling off at small values (the 'scree' test). We retain those principal components corresponding to the eigenvalues before the cut-off point or 'elbow' (Figure 10.9).

However, on occasion the eigenvalues drift downwards with no obvious cutting point and the first few eigenvalues account for only a small proportion of the variance.

It is very difficult to determine the 'right' number of components and most tests are for limited special cases and assume multivariate normality. Jackson (1991) describes a range of procedures and reports the results of several comparative studies.

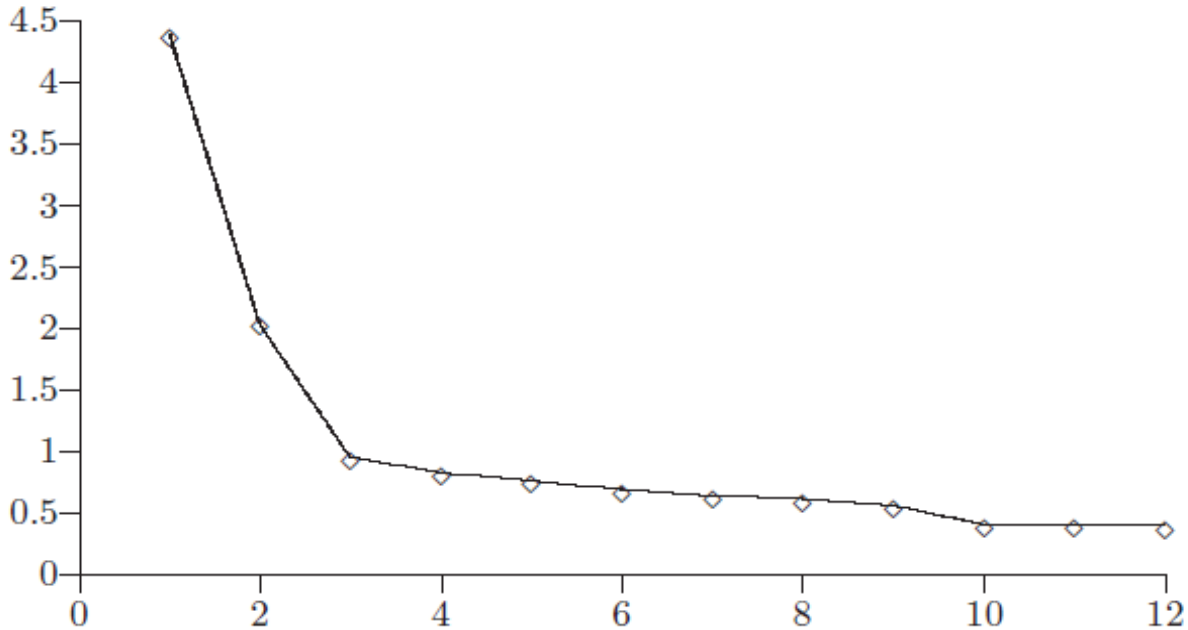


Figure 10.9 Eigenvalue spectrum: a plot of the ordered eigenvalues against eigenvalue number, with an 'elbow' at the third eigenvalue.

Principal component analysis

- For ξ to have zero mean, we must define it using

$$\xi = \mathbf{A}^t(x - \mu)$$

  μ is the sample mean in practice.

- What do the points in the reduced dimension correspond to in the original space?

$$\mathbf{x} = \mathbf{A}\xi + \mu$$

- The "recovered" x after approximation by retaining r principal components is

$$\mathbf{x}_r = \mathbf{A}_r\xi_r + \mu$$

Principal component analysis

$$\boldsymbol{x}_r = \boldsymbol{A}_r \boldsymbol{A}_r^T (\boldsymbol{x} - \boldsymbol{\mu}) + \boldsymbol{\mu}$$

The transformation $\boldsymbol{A}_r \boldsymbol{A}_r^T$ is of rank r and maps the original data distribution to a distribution that lies in an r-dimensional subspace (or on a *manifold* of dimension r) in $\mathbb{R}^p$. The vector $\boldsymbol{x}_r$ is the position the point $\boldsymbol{x}$ maps down to, given in the original coordinate system (the projection of $\boldsymbol{x}$ on to the space defined by the first r principal components). This is illustrated in Figure 10.10 for a projection of a two-dimensional point onto its first principal component.

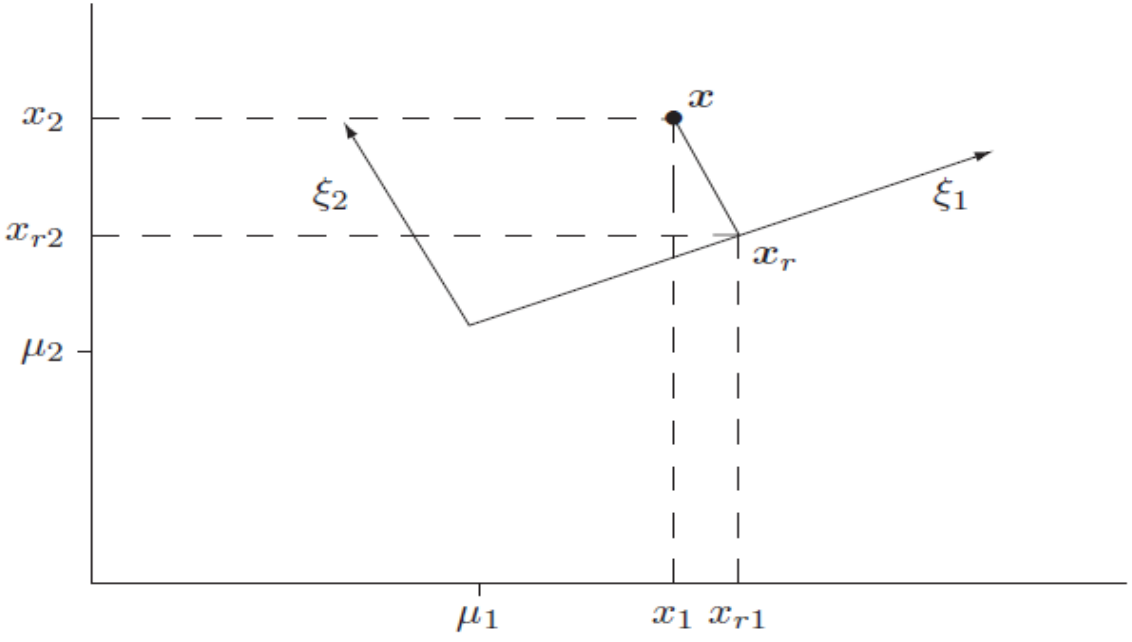


Figure 10.10 Reconstruction from projections: $\boldsymbol{x}$ is approximated by $\boldsymbol{x}_r$ using the first principal component.

Principal component analysis

Principal components analysis is often the first stage in a data analysis and is used to reduce the dimensionality of the data while retaining as much as possible of the variation present in the original dataset.

Principal components analysis takes no account of groups within the data (i.e. it is unsupervised). Although separate groups may emerge as a result of projecting data to a reduced dimension, this is not always the case and dimension reduction may obscure the existence of separate groups.

Figure 10.11 illustrates a dataset in two dimensions with three separate groups and the principal component directions. Projection onto the first eigenvector will remove group separation while projection onto the second retains group separation. Therefore, although dimension reduction may be necessary, the space spanned by the vectors associated with the first few principal components will not necessarily be the best for discrimination.

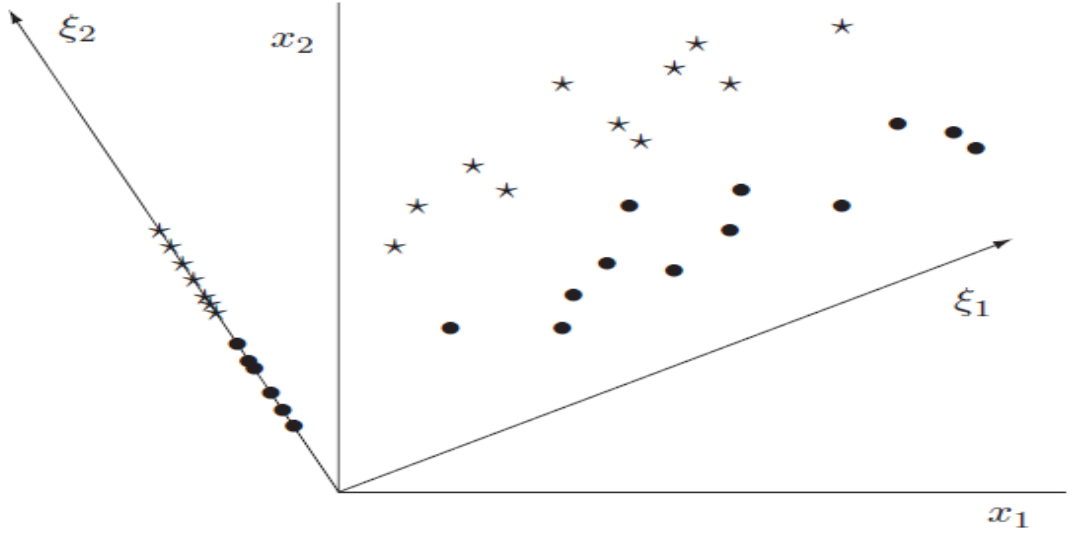


Figure 10.11 Two-group data and the principal axes.

Principal component analysis

- How to use principal component analysis to select features? The straight answer is that it depends on the task at hand. Likely that components good for regression (good fit to data) may not be good for discrimination.
- Most common method is to retain eigenvalues that account for 90% of total variance.
- Another method is to retain the kth eigenvector if its eigenvalue λ_k exceeds $\sum_{i=1}^{p}(1/i)$
- After all this consideration there is no guarantee that the transformed variables will be better for discrimination than a subset selection obtained from the "feature selection" methods we have considered.

Principal component analysis

	x	y
	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
Data =	3.1	3.0
	2.3	2.7
	2	1.6
	1	1.1
	1.5	1.6
	1.1	0.9

Principal component analysis

	x	y
	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
Data =	3.1	3.0
	2.3	2.7
	2	1.6
	1	1.1
	1.5	1.6
	1.1	0.9

	x	y
	0.69	0.49
	-1.31	-1.21
	0.39	0.99
	0.09	0.29
DataAdjust =	1.29	1.09
	0.49	0.79
	0.19	-0.31
	-0.81	-0.81
	-0.31	-0.31
	-0.71	-1.01

Principal component analysis

	x	y
	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
Data =	3.1	3.0
	2.3	2.7
	2	1.6
	1	1.1
	1.5	1.6
	1.1	0.9

	x	y
	0.69	0.49
	-1.31	-1.21
	0.39	0.99
	0.09	0.29
DataAdjust =	1.29	1.09
	0.49	0.79
	0.19	-0.31
	-0.81	-0.81
	-0.31	-0.31
	-0.71	-1.01

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

Principal component analysis

	x	y
	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
Data =	3.1	3.0
	2.3	2.7
	2	1.6
	1	1.1
	1.5	1.6
	1.1	0.9

	x	y
	0.69	0.49
	-1.31	-1.21
	0.39	0.99
	0.09	0.29
DataAdjust =	1.29	1.09
	0.49	0.79
	0.19	-0.31
	-0.81	-0.81
	-0.31	-0.31
	-0.71	-1.01

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix} \qquad cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

Principal component analysis

	x	y
	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
Data =	3.1	3.0
	2.3	2.7
	2	1.6
	1	1.1
	1.5	1.6
	1.1	0.9

	x	y
	0.69	0.49
	-1.31	-1.21
	0.39	0.99
	0.09	0.29
DataAdjust =	1.29	1.09
	0.49	0.79
	0.19	-0.31
	-0.81	-0.81
	-0.31	-0.31
	-0.71	-1.01

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix} \qquad cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

Principal component analysis

Transformed Data (Single rigenvector)
x
-0.827970186
1.77758033
-0.992197494
-0.274210416
-1.67580142
-0.912949103
0.991094375
1.14457216
0.438046137
1.22382056

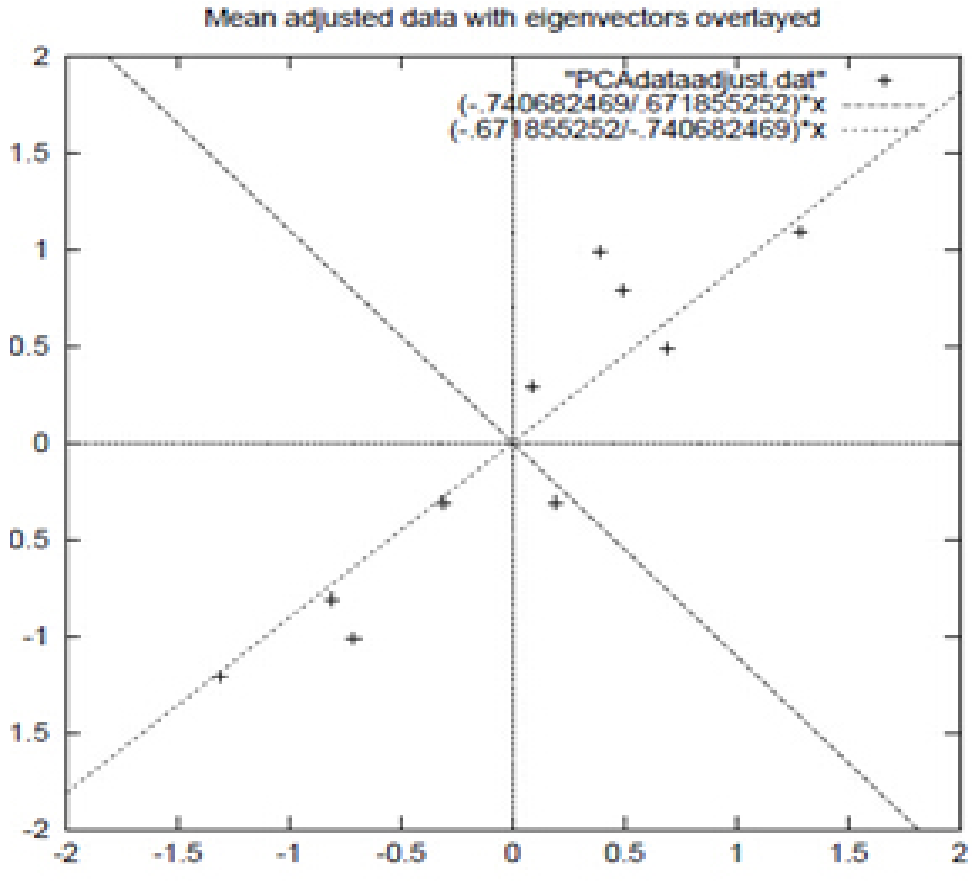


图 2

References

1. A. Webb , “Statistical Pattern Recognition,” Second Edition, John Wiley and Sons, 2002.
2. Anil K. Jain, Robert P.W. Duin and Jianchang Mao “Statistical Pattern Recognition: A Review,” IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, pp. 4 - 37, January 2000.
3. Richard O. Duda, Peter E. Hart and David G. Stork, “Pattern Classification,” Second Edition, John Wiley and Sons, 2001.
4. K. V. Mardia, J. T. Kent and J. M. Bibby, “Multivariate Analysis,” Academic Press, 2003.