

## Structure in Data

A major objective in data analysis is to identify interesting features or structure in the data.

The graphical methods are very useful in discovering structure.

There are basically two ways of thinking about “structure”.

One has to do with *counts* of observations. In this approach, patterns in the *density* are the features of interest. We may be interested in whether the density is multimodal, whether it is skewed, whether there are holes in the density, and so on.

## Structure in Data

The other approach seeks to identify relationships among the variables.

The two approaches are related in the sense that if there are relationships among the variables, the density of the observations is higher in regions in which the relationships hold.

Relationships among variables are generally not exact, and the relationships are identified by the higher density of observations that exhibit the approximate relationships.

# Structure in Data

An important kind of pattern in data is a relationship to time. Often, even though data are collected at different times, the time itself is not represented by a variable on the dataset.

Interesting structure may also be groups or clusters of data based on some measure of similarity.

When there are separate groups in the data, but the observations do not contain an element or an index variable representing group membership, identifying nearby elements or clusters in the data requires some measure of similarity (or, equivalently, of dissimilarity).

# Types of Data

Although we often assume that the data space is a subspace of  $\mathbb{R}^m$ , a data space may be more general.

Data, for example, may be character strings such as names.

The more general types of data may be mapped from the original data space to a “feature space”, which is a subspace of  $\mathbb{R}^m$ .

The variables may be measured on different scales; they may, of course, represent completely different phenomena, so measurement scales cannot be made the same.

One way of reconciling the measurements, however, is to standardize the data using the transformation

$$X_S = (X - \bar{X}) \text{diag}(1/\sqrt{s_{ii}}),$$

where  $\bar{X}$  is the matrix whose constant columns contain the means of the corresponding columns of  $X$ , and  $\sqrt{s_{ii}}$  is the sample standard deviation of the  $i^{\text{th}}$  column of  $X$ .

# Structure in Data

We may be interested in finding the *nearest neighbors* similarity; or, alternatively, we may be interested in identifying all observations within a given degree of closeness to a given observation. This problem is called a “proximity search”.

The emphasis is on exploratory analysis, and the main goals are to identify clusters in data and to determine lower-dimensional structures in multidimensional data.

Interesting structure may involve clusters of data, or it may be the result of the data lying on or near a space of reduced dimension.

Interesting structure may also be defined generically as properties of the data that differ from expected properties if the data were a random sample from a multivariate normal distribution or from some other standard distribution.

## **“Normal” Data**

The normal (or Gaussian) distribution lies at the heart of many methods of data analysis.

A heuristic definition of structure as a departure from normality can be motivated by the fact that most randomly selected low-dimensional projections of any high-dimensional dataset will appear similar to a random sample from a multivariate normal distribution.

# Cluster Analysis

The usual objective in cluster analysis is to divide the observations into groups that are close to each other or are more homogeneous than the full set of observations.

An observation may consist of categorical variables that may (or may not) specify the class to which the observation belongs. In general, if the  $i^{\text{th}}$  observation can be represented as

$$x_i = (x_i^r, x_i^c),$$

where the subvector  $x_i^c$  represents values of the categorical variables, we may wish to handle the  $x_i^c$  component separately.

# Clustering and Classification

Identifying groups of similar observations in a dataset is an important step in making sense of the data and in understanding the phenomena represented by the data.

Clustering, classification, and discrimination are terms that describe this activity, which lies at the crossroads of a number of traditional disciplines, including statistics, computer science, artificial intelligence, and electrical engineering.

Classification is sometimes called *machine learning*, especially in the more engineering-oriented disciplines.

The first step in forming groups is to develop a definition of the groups. This may be based on similarities of the observations or on closeness of the observations to one another.

# Clustering

Cluster analysis is generally exploratory.

It seeks to determine what groups are present in the data.

If the groups are known from some training set, “discriminant analysis” seeks to understand what makes the groups different and then to provide a method of classifying observations into the appropriate groups.

When discriminant analysis is used to “train” a clustering method, we refer to the procedure as “supervised” classification. Discriminant analysis is mechanically simpler than cluster analysis.

Clustering is “unsupervised” classification.

# Clustering

Because of the large number of possibilities for grouping a set of data into clusters, we generally must make some decisions to simplify the problem.

One way is to decide a priori on the number of clusters.

Another way is to do recursive clustering; that is, once trial clusters are formed, observations are not exchanged from one cluster to another. Two pairs of observations that are in different clusters at one stage of the clustering process would never be split so that at a later stage one member of each pair is in one cluster and the other member of each pair is in a different cluster.

# Recursive Clustering

There are two fundamentally different approaches to recursive clustering.

One way is to start with the full dataset as a single group and, based on some reasonable criterion, partition the dataset into two groups. This is called divisive clustering.

The criterion may be the value of some single variable; for example, any observation with a value of the third variable larger than 5 may be placed into one group and the other observations placed in the other group.

Each group is then partitioned based on some other criterion, and the partitioning is continued recursively. This type of divisive clustering or partitioning results in a classification tree, which is a decision tree each node of which represents a partition of the dataset.

## Recursive Clustering

Another way of doing recursive clustering is to begin with a complete clustering of the observations into singletons.

Initially, each cluster is a single observation, and the first multiple-unit cluster is formed from the two closest observations.

This agglomerative, bottom-up approach is continued so that at each stage the two nearest clusters are combined to form one bigger cluster.

# K-Means Clustering

The objective in K-means clustering is to find a partition of the observations into a preset number of groups,  $k$ , that minimizes the variation within each group. Each variable may have a different variation, of course.

The variation of the  $j^{\text{th}}$  variable in the  $g^{\text{th}}$  group is measured by the within sum-of-squares,

$$s_{j(g)}^2 = \frac{\sum_{i=1}^{n_g} (x_{ij(g)} - \bar{x}_{j(g)})^2}{n_g - 1},$$

where  $n_g$  is the number of observations in the  $g^{\text{th}}$  group, and  $\bar{x}_{j(g)}$  is the mean of the  $j^{\text{th}}$  variable in the  $g^{\text{th}}$  group.

There are  $m$  such quantities, and the objective is to minimize their sum (or a linear combination of them).

# K-Means Clustering

The objective in K-means clustering is to find a partition of the observations into a preset number of groups  $k$  that minimizes, over all groups, the total of the linear combinations of the within sum-of-squares for all variables.

For linear combinations with unit coefficients, this quantity is

$$w = \sum_{g=1}^k \sum_{j=1}^m \sum_{i=1}^{n_g} (x_{ij(g)} - \bar{x}_{j(g)})^2.$$

Determining the partitioning to minimize this quantity is a computationally intensive task.

In practice, we seek a local minimum (that is, a solution such that there is no single switch of an observation from one group to another group that will decrease the objective). Even the procedure used to achieve the local minimum is rather complicated.

# Methods for K-Means Clustering

Hartigan and Wong (1979) give an algorithm (and Fortran code) for performing the clustering. Their algorithm forms a set of initial trial clusters and then transfers observations from one cluster to another while seeking to decrease the sum of squares.

Simulated annealing can also be used to do K-means clustering.

Most of the algorithms for K-means clustering will yield different results if the data are presented in a different order. Those using techniques, such as simulated annealing, that depend on random numbers may yield different results on different runs with the same data in the same order.

In either method for performing K-means clustering, it is necessary to choose initial points and then trial points to move around.

## **K-Means Clustering**

The clustering depends on the variability of the variables.

It may be necessary to scale the variables in order for the clustering to be sensible because the larger a variable's variance, the more impact it will have on the clustering.

# Choosing the Number of Clusters

A major issue is how many clusters should be formed.

This question must generally be addressed in an ad hoc manner.

A number of statistics have been proposed for use in deciding how many clusters to use.

The Calinski-Harabasz index,

$$\frac{b/(k - 1)}{w/(n - k)},$$

where  $b$  is the between-groups sum-of-squares,

$$b = \sum_{g=1}^k \sum_{j=1}^m (\bar{x}_{j(g)} - \bar{x}_j)^2,$$

and  $w$  is the pooled within-groups sum-of-squares, can be used as a stopping rule. The objective is to maximize it.

# Hierarchical Clustering

It is useful to consider a hierarchy of clusterings from a single large cluster to a large number of very small clusters. Hierarchical clustering yields these alternative clusterings.

The results of a hierarchical clustering can be depicted as a tree.

Each point along the bottom of the tree may correspond to a single observation.

Nodes higher up in the diagram represent successively larger groups.

The number of clusters depends on the level in the tree, as indicated in the plot.

Many of the algorithms for hierarchical clustering will yield different results if the data are presented in a different order.

# Agglomerative Hierarchical Clustering

In agglomerative hierarchical clustering, we first begin with a large number of clusters, generally as many as the number of observations, so that each cluster consists of a single observation, and then we combine clusters that are nearest to each other.

To define distances between groups, we must first consider from what points within the groups to measure distance.

One way is to measure the distance between a central point in one group, such as the mean or median of the group, and the corresponding central point in the other group. These methods often do not work very well in hierarchical clustering.

In agglomerative hierarchical clustering, the distance between two clusters is usually chosen in one of three ways.

# Agglomerative Hierarchical Clustering

- The minimum distance between a point in the first cluster and a point in the second cluster. Using this criterion results in what is sometimes called “single linkage” clustering.
- The distance between clusters is the average of the distances between the points in one cluster and the points in the other cluster.
- The largest distance between a point in one cluster and a point in the other cluster. Using this criterion results in what is sometimes called “complete linkage” clustering.

# Agglomerative Hierarchical Clustering

In addition to the choice of the two points to define the distance, different distance metrics can be chosen.

Most clustering methods use an  $L_2$  metric.

By changing the distance metric and the clustering method, several different cluster trees can be created from a single dataset. No one method seems to be useful in all situations.

We can see the differences in hierarchical clustering with different distance measures between clusters using the example on page 243. In this example, the clusters at any intermediate stage except the first are different.

# Model-Based Hierarchical Clustering

In the general clustering problem, we may assume that the data come from several distributions, and our problem is to identify the distribution from which each observation arose.

Without further restrictions, this problem is ill-posed; no solution is any better than any other.

We may, however, impose the constraint that the distributions be of a particular type.

We may then formulate the problem as one of fitting the observed data to a mixture of distributions of the given type.

The R/S-Plus function `mclust` performs model-based clustering.

# Divisive Hierarchical Clustering

Most hierarchical clustering schemes are agglomerative; that is, they begin with no clusters and proceed by forming ever-larger clusters.

In divisive hierarchical clustering, we begin with a single large cluster and successively divide the clusters into smaller ones.

At each stage, the cluster with the largest dissimilarity between any two of its observations is selected to be divided.

To divide the selected cluster, the observation with the largest average dissimilarity to the other observations of the selected cluster is used to define a “splinter group”.

Next, observations that are closer to the splinter group than to their previous groups are assigned to the splinter group. This is continued until all observations have been assigned to a single cluster. The result is a hierarchical clustering. The R/S-Plus function `diana` determines clusters by this method.

# Clustering and Classification by Space Tessellations

Groups in data can naturally be formed by partitioning a space in which the data are represented.

If the data are represented in a cartesian coordinate system, for example, the groupings can be identified by polytopes that fill the space.

Groups are separated by simple planar structures.

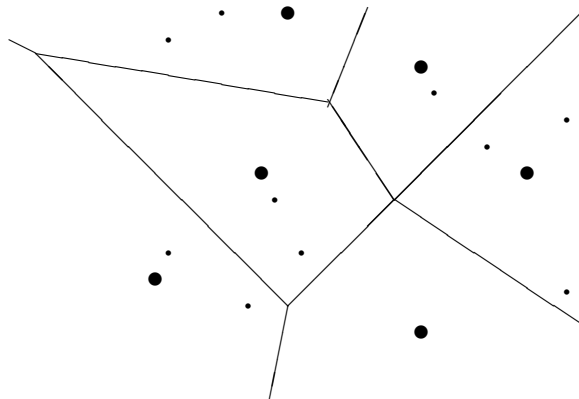
Groups may be formed by the values of only a subset of the variables.

# Space Tessellations

A simple example is data in which one or more variables represent geographic location.

Clusters may be defined based on location, either by methods such as we have discussed above or by regions that tessellate the space.

The tessellations may be preassigned regions, perhaps corresponding to administrative or geographical boundaries.



# Methods for Tessellations

The Bowyer-Watson algorithm exploits this property for computing a Delaunay triangulation (Bowyer, 1981, Watson, 1981). Starting with  $d + 1$  points and a simplex, the algorithm proceeds by recursive insertion of nodes. For each new node:

1. Find any simplices whose circumscribed hyperspheres include the new node.
2. Create a cavity by eliminating these simplices (if there are any).
3. Create the new set of simplices by connecting the new point to the nodes that define this cavity.

## **Meanings of Clusters; Conceptual Clustering**

Identification of clusters in a dataset is usually only a small part of a larger scientific investigation.

Another small step toward the larger objective of understanding the phenomenon represented by the data is to characterize the groups in the data by simple descriptions in terms of ranges of individual variables.

A set of conjunctive rules that results can aid in understanding the phenomenon being studied. The rules that define classes can be formulated in terms of either numerical or categorical variables.

## **Meanings of Clusters; Conceptual Clustering**

If the intent of an analysis is interpretation or understanding of the phenomenon that gave rise to the data, simplicity of the description of clusters has great value, even if it is achieved at some cost in accuracy.

If, on the other hand, the objective is an ad hoc classification of the observations in a given dataset, simplicity is not important, and often an algorithmic “black box” is a more effective classifier.

## **Fuzzy Clustering**

Fuzzy set theory has been applied to clustering, as it has to most problems that have a component of uncertainty.

Instead of observations being grouped into definite clusters, they are given membership probabilities.

# Fuzzy Clustering

The membership probability of the  $i^{\text{th}}$  observation in the  $g^{\text{th}}$  group is  $u_{ig}$ . The memberships satisfy

$$0 \leq u_{ig} \leq 1$$

and

$$\sum_{g=1}^k u_{ig} = 1 \quad \text{for all } i = 1, \dots, n.$$

The quantity to be minimized is

$$\sum_{g=1}^k \sum_{j=1}^m \sum_{i=1}^n u_{ig}^2 (x_{ij} - \bar{x}_{j(g)})^2,$$

where, as before,  $\bar{x}_{j(g)}$  is the mean of the  $j^{\text{th}}$  element of the vectors  $x_i$  that are in the  $g^{\text{th}}$  group.

Because group membership is a weight, however,

$$\bar{x}_{j(g)} = \frac{\sum_{i=1}^n u_{ig}^2 x_{ij}}{\sum_{i=1}^n u_{ig}^2}.$$

# Transformations of Data

In many data-analytic procedures, we perform various linear transformations on the data, with predictable results on the analysis.

For example, we often perform a simple univariate standardization of the data by subtracting the sample mean and dividing by the sample standard deviation.

For the typical data matrix  $X$  whose columns represent variables and whose rows represent multivariate observations, we may standardize each variable by subtracting the column mean from each value in the column and dividing by the standardization of the column.

# Clustering and Transformations of the Data

Transformations on the data may change the relative values of measures of similarity.

This, of course, affects any method of analysis that depends on measures of similarity.

A severe limitation of clustering results from the dependence of the clusters on the scaling of the data.

Doing a simple univariate standardization of the data by subtracting the sample mean and dividing by the sample standard deviation affects the clustering.

# Clustering and Transformations of the Data

Even the units of measurement affects the clustering.

Whether one variable is measured in grams or kilograms affects the relative distance of any one observation to the other observations.

If all variables in the dataset are of the same type (mass, say), it is easy to measure them all in the same units; if some are of one type and some are of another type, decisions on units are not as easy.

These decisions, however, affect the results of clustering.

# Clustering and Transformations of the Data

The dependence of the clustering on transformations of the data results from the effect on the distance measures. This can have a useful place in the data analysis.

Transformations can help in the identification of clusters.

Transformations may also be useful in finding other types of structure in data.

# Clustering of Variables

There is a basic duality between the  $m$  “variables” and the  $n$  “observations” of the dataset  $X$ .

We have been discussing clustering of observational units. Clustering observations is done by measures of distance, possibly scaled by  $S$ .

Consider reversing the roles of variables and observations. Suppose that we wish to cluster the variables (that is, we wish to know which variables have values that are strongly related to each other).

The relative values of the variables provide information on how similar or dissimilar the observations are; conversely, the relative values of the multivariate observations provide information on the similarity of the variables. Clustering of variables is conceptually and mechanically the same as clustering of observations.

## Comparing Clusterings

Various methods of clustering yield different results and the same method yields different results if the data have been transformed.

Which clustering is best cannot in general be determined by analysis of data with no context.

The purpose of the clustering, after all, is to develop a better understanding of a phenomenon of which the data measure various aspects.

Nevertheless, it is instructive to develop numerical measures of the agreement (or, equivalently, disagreement) of different clusterings of the same dataset.

## Comparing Clusterings

A two-way contingency table can be used to represent agreement of two clusterings.

If the classes of one clustering are denoted as  $C_{11}, \dots, C_{1k_1}$  and those of a second clustering as  $C_{21}, \dots, C_{2k_2}$ , a two-way table of the numbers of units falling in the cells is constructed, as shown.

	$C_{11}$	$\dots$	$C_{1k_1}$	
$C_{21}$	$n_{11}$	$\dots$	$n_{1k_1}$	$n_{1\bullet}$
$\vdots$		$\ddots$		$\vdots$
$C_{2k_2}$	$n_{k_21}$	$\dots$	$n_{k_2k_1}$	$n_{k_2\bullet}$
	$n_{\bullet 1}$	$\dots$	$n_{\bullet k_2}$	$n$

The labeling of the clusters is arbitrary. (In a classification problem, the clusters correspond to classes, which are usually known and fixed, given the data.)

## Comparing Clusterings

From the cluster trees shown in Figure 10.11 on page 251, there appear to be two obvious clusters in the first clustering and three clusters in the second clustering. If we identify the clusters from left to right in each tree (so that, for example, the first cluster in the first tree contains the points 2, 5, and 7, and the first cluster in the second tree contains the single point 3), we would have the table below.

	$C_{11}$	$C_{12}$	
$C_{21}$	0	1	1
$C_{22}$	3	0	3
$C_{23}$	0	3	3
	3	4	7

The marginal totals are the counts for the corresponding clusters. The numbers in the cells indicate the extent of agreement of the two clusterings. Perfect agreement would yield, first of all,  $k_1 = k_2$ , and, secondly, a table in which each column and each row contains only one nonzero value.

## Comparing Clusterings

Of the total of  $\binom{n}{2}$  pairs of points, each pair may be:

1. in the same cluster in both clusterings;
2. in different clusters in both clusterings;
3. in the same cluster in one clustering but in different clusters in the other clustering.

Both the first and second events indicate agreement of the clusterings, and the third indicates disagreement.

## Rand's Statistic for Comparing Clusterings

Rand's statistic is a count of the number of pairs of the first and second types divided by the total number of pairs.

This statistic is obviously in the interval  $[0, 1]$ , and a value of 0 indicates total disagreement and a value of 1 complete agreement.

For the two clusterings shown in Figure 10.11 on page 251, with two clusters in the first clustering and three clusters in the second, we see that the count of agreements is 18; hence, Rand's statistic is  $6/7$ .

# Computational Complexity of Clustering

The task of identifying an unknown number of clusters that are distinguished by unknown features is an exceedingly complex problem.

In practical clustering methods, there are generally trade-offs between how clusters are defined and the algorithm used to find the clusters. In the hierarchical clustering algorithms, the algorithm dominates the approach to the problem.

In those hierarchical clustering methods, the definition of clusters at any level is merely what results from a specified algorithm.

After the  $O(mn^2)$  computations to compute the distance matrix, the algorithm requires only  $O(n)$  computations.

# Computational Complexity of K-Means Clustering

Even with the simplifying assumption that the number of clusters is known, the definition of clusters requires a very computationally intensive algorithm. There are  $n^k$  possibilities.

K-means clustering begins with a reasonable definition of clusters, assuming a known number of clusters.

Development of clustering algorithms that are feasible for large datasets is an important current activity.

## **Ordering and Ranking Multivariate Data**

The concept of order or rank within a multivariate dataset can be quite complicated.

# Minimal Spanning Trees

A *spanning tree* for a graph is a tree subgraph that contains all nodes of the given graph.

A spanning tree is not necessarily rooted.

A useful graph of observations is the spanning tree whose edges have the least total distance.

This is called a *minimal spanning tree*, or MST. It is obvious that the number of edges in a minimal spanning tree would be one less than the number of nodes. A minimal spanning tree may not be unique.

It is easy to see that the problem of determining an MST is  $O(n^2)$ . This is prohibitive for large datasets.

The R/S-Plus function `mstree` computes a minimal spanning tree.

# Minimal Spanning Trees

An MST helps us to understand the distribution of the observations and to identify clusters of observations and outlying observations.

The number of edges in the longest path starting at any node is called the *eccentricity* of that node or observation.

The node most distant from a given node is called an *antipode* of the node, and the path between a node with greatest eccentricity and its antipode is called a *diameter* of the tree.

The length of such a path is also called the diameter. (This word also carries both meanings in the familiar context of a circle.)

A node with minimum eccentricity is called a *center* node or a *median*.

# Ranking Data Using Minimal Spanning Trees

Multivariate observations can be ranked or sorted based on a minimal spanning tree.

The procedure is to define a starting node at an endpoint of a tree diameter and then to proceed through the tree in such a way as to visit any shallow subtrees at a given node before proceeding to the deeper subtrees.

For the tree shown in the right-hand plot in Figure 10.12 on page 258, if we choose to begin on the right of the tree, the next seven nodes are in the single path from the first one. Finally, at the eighth node, we choose the shallow subtree for the ninth and tenth nodes. The eleventh node is then the other node connected to the eighth one.

This ordering is shown in the left-hand plot in Figure 10.13.

# Ranking Data Using Convex Container Hulls

Another way to order data is by convex hull peeling.

The idea behind convex hull peeling, which is due to John Tukey, is that the convex hull of a dataset identifies the extreme points, and the most extreme of these is the one whose removal from the dataset would yield a much “smaller” convex hull of the remaining data.

In two dimensions, convex hull peeling takes as the most extreme observation the one on the convex hull with the smallest angle.

Next, the convex hull of all remaining points is determined, and the second most extreme observation is the one on this convex hull with the smallest angle. This process continues until a total ordering of all observations is achieved.

The ordering by convex hull peeling of the dataset from page 257 is shown in Figure 10.14 on page 260.

# Ranking Data Using Convex Hull Peeling

The ordering by convex hull peeling tends to move around the edges of the set of points, often similar to the radial ordering in a minimal spanning tree.

Various programs for computing convex hulls and other problems in computational geometry are available at the site,

[www.geom.umn.edu/software/download/](http://www.geom.umn.edu/software/download/)

Instead of a convex hull, formed by planes, we may consider a smoothed figure such as an ellipsoid with minimum volume.

The minimum-volume ellipsoid that contains a given percentage of the data provides another way of ordering the points in a multivariate dataset.

# Location Depth

A peeled convex hull or an ellipsoid containing a given percentage of data provides an ordering of the data from the outside in.

Another approach to ordering data is to begin in the inside—that is, at the densest part of the data—and proceed outward. For bivariate data, John Tukey introduced the concept of *halfspace location depth* for a given point  $x_c$  relative to the dataset  $X$  whose rows are in  $\mathbb{R}^2$ .

The halfspace location depth,  $d_{\text{hsl}}(x_c, X)$ , is the smallest number of  $x_i$  contained in any closed halfplane whose boundary passes through  $x_c$ .

The halfspace location depth is defined for datasets  $X$  whose rows are in  $\mathbb{R}^m$  by immediate extension of the definition for the bivariate case.

## Ranking Data Using Location Depth

The halfspace location depth provides another way of ordering the data. There are generally many ties in this ordering.

Ordering by location depth emphasizes the interior points, whereas convex hull peeling emphasizes the outer points.

If a single point has a greater location depth than any other point in the dataset, the point is called the *depth median*.

If multiple points have the largest location depth of any in the dataset, the depth median is the centroid of all such points.

Determination of the depth median is computationally intensive.

## **Other Ways of Measuring Location Depth**

There are other ways of defining data depth.

One approach is to define a measure of distance of depth based on maximal one-dimensional projections.

This measure can be used to order the data, and it is also useful as an inverse weight for robust estimators of location and scale. It is computationally intensive, and most methods in use depend on sampling of the data.

# Ordering by Clustering

Clustering also provides a way of ordering or, especially, of partially ordering data.

The ordering that arises from clustering, whether divisive or agglomerative, however, depends on local properties, and a global ordering is difficult to identify.

The ordering by a hierarchical clustering of the dataset from page 257 is shown in Figure 10.16 on page 262.

The ordering, or partial ordering, would be from left to right (or from right to left) along the leaves of the tree. Comparison of the cluster tree with the scatter plot in on page 258 shows how nearby points are grouped first.

In this dataset, the points closest together are on the periphery of the data cloud. For a cloud of points that is concentrated around a centroid, the central points would be grouped first.

# Clustering by Ordering

The process of ordering by clustering can be turned around.

The ordering of the data can be used to cluster the data.

For example, the minimal spanning tree shown in Figure 10.12 can be used to form the clusters indicated in Figure 17.

The clusters are defined by a tessellation formed by boundaries perpendicular to the longer edges in the MST.

As it turns out, the four clusters shown in the MST correspond to the four clusters formed by the hierarchical clustering shown in Figure 10.16.

It is not always the case that a clustering can be formed by simple cuts of the branches of a minimal spanning tree to correspond to the clustering formed by a particular hierarchical algorithm.

## Ordering and Ranking of Transformed Data

Minimal spanning trees depend on relative distances between points, so, as we would expect, the minimal spanning tree and any ordering based on it may be different if the data are transformed. Likewise, of course, orderings based on clustering may be changed by transformations of the data.

An important property of convex hulls and the depth of data is that they are not affected by affine transformations.

# Linear Principal Components

The information in observations that consists of  $m$  components may be adequately represented by transformed observations consisting of a smaller set of  $k$  components.

This type of data reduction in the dimension of the data may allow a reduction in the storage requirements, but, more importantly, it may help in understanding the data.

Dimension reduction is useful in identifying structure in the data and also in discovering properties that the data do not measure directly.

We may wish to extract features in the data that have been obscured by measurement error or other sources of noise.

# Linear Principal Components

A basic tenet of data analysis is that variation provides information, and an important approach in statistics is the analysis of variation.

When many variables are present, it may be difficult to identify individual effects, so it might be useful to reduce the number of variables.

Another basic tenet is that covariance among a set of variables reduces the amount of information that the variables contain.

We therefore seek to combine variables in such a way that their covariance is reduced or, more generally, that they are independent. For normal variables, of course, zero covariance is equivalent to independence.

## Dimension Reduction

The basic problem therefore is to transform the observed  $m$ -vectors  $x$  into  $k$ -vectors  $\tilde{x}$  that, as a set, exhibit almost as much variation as the original set and are mutually independent or “almost” so.

Because of differences in the meanings of the variables, it is best first to standardize the data as

$$X_S = (X - \bar{X}) \text{diag}(1/\sqrt{s_{ii}}),$$

where  $\bar{X}$  is the matrix whose constant columns contain the means of the corresponding columns of  $X$ , and  $\sqrt{s_{ii}}$  is the sample standard deviation of the  $i^{\text{th}}$  column of  $X$ .

# Linear Principal Components

There are various ways of combining a set of variables into a smaller set (that is, of transforming  $m$ -vectors into  $k$ -vectors).

One of the simplest methods is to use linear transformations. If the linear transformations result in new variables that are orthogonal (that is, if they have zero sample correlation), and if the data are multivariate normal, then the new variables are independent.

The linear combinations are called “principal components”, “empirical orthogonal functions” or “EOF” (especially in meteorology and atmospheric research), “latent semantic indices” (especially in information retrieval), “factors” (especially in the social sciences), Karhunen-Loève transforms (especially in signal analysis), or “independent components” (also in signal analysis).

# Data Reduction

There are some differences among and within these methods.

The differences have to do with assumptions about probability distributions and with the nature of the transformations.

In factor analysis, a rather strong stochastic model guides the analysis.

In independent components analysis, rather than seeking only orthogonality, which yields zero correlations, and hence independence for normal data, transformations may be performed to yield zero cross moments of higher order. (Correlations are cross moments of second order.)

Independent component analysis, therefore, may involve nonlinear transformations. Any of the methods mentioned above may also utilize nonlinear combinations of the observed variables.

# The Probability Model Underlying Principal Components Analysis

Linear *principal components analysis* (PCA) is a technique for data reduction by constructing linear combinations of the original variables that account for as much of the total variation in those variables as possible.

Linear principal components is a method of “decorrelating” the elements of a vector random variable.

The method depends on the variances of the individual elements, so it is generally best first to perform transformations as necessary so that all elements have the same variance.

Also it is convenient to subtract the mean of each element of the random variable. The transformed vector is thus standardized so that the mean of each element is 0 and the variance of each element is 1.

# The Probability Model Underlying Principal Components Analysis

Consider an  $m$ -vector random variable  $Y$  with variance-covariance matrix  $\Sigma$ , which has 1's along the diagonal.

We seek a transformation of  $Y$  that produces a random vector whose elements are uncorrelated; that is, we seek a matrix  $W$  with  $m$  columns such that  $V(WY)$  is diagonal. (Here,  $V(\cdot)$  is the variance.) Now,

$$V(WY) = W\Sigma W^T,$$

so the matrix  $W$  must be chosen so that  $W\Sigma W^T$  is diagonal.

# The Probability Model Underlying Principal Components Analysis

The obvious solution is to decompose  $\Sigma$ :

$$\Sigma = W^T \Lambda W.$$

The spectral decomposition of the variance-covariance matrix is

$$\Sigma = \sum_{k=1}^m \lambda_k w_k w_k^T,$$

with the eigenvalues  $\lambda_k$  indexed so that  $0 \leq \lambda_m \leq \dots \leq \lambda_1$  and with the  $w_k$  orthonormal; that is,

$$I = \sum_k w_k w_k^T.$$

Now, consider the random variables

$$\tilde{Y}_{(k)} = w_k^T Y,$$

which we define as the *principal components* of  $Y$ .

# The Probability Model Underlying Principal Components Analysis

The first principal component,  $\tilde{Y}_{(1)}$ , is the projection of  $Y$  in the direction in which the variance is maximized, the second principal component,  $\tilde{Y}_{(2)}$ , is the projection of  $Y$  in an orthogonal direction with the largest variance, and so on.

It is clear that the variance of  $\tilde{Y}_{(k)}$  is  $\lambda_k$  and that the  $\tilde{Y}_{(k)}$  are uncorrelated; that is, the variance-covariance matrix of the random vector  $(\tilde{Y}_{(1)}, \dots, \tilde{Y}_{(m)})$  is  $\text{diag}(\lambda_1, \dots, \lambda_m)$ .

Heuristically, the  $k^{\text{th}}$  principal component accounts for the proportion

$$\frac{\lambda_k}{\sum \lambda_j}$$

of the “total variation” in the original random vector  $Y$ .

## PCA

The linear combinations  $\tilde{Y}_{(k)}$  that correspond to the largest eigenvalues are most interesting.

If we consider only the ones that account for a major portion of the total variation, we have reduced the dimension of the original random variable without sacrificing very much of the potential explanatory value of the probability model.

# PCA

Thus, using only the  $p$  largest eigenvalues, instead of the  $m$ -vector  $Y$ , we form the transformation matrix  $W$  as

$$W = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_p^T \end{bmatrix}.$$

This produces the  $p$ -vector  $WY = \tilde{Y} = (\tilde{Y}_{(1)}, \dots, \tilde{Y}_{(p)})$ .

The matrix

$$\Sigma_p = \sum_{k=1}^p \lambda_k w_k w_k^T$$

is the variance-covariance matrix of  $\tilde{Y}$ .

# PCA

The matrix  $\Sigma_p$  is an approximation to the matrix  $\Sigma$ .

In fact, it is the matrix of rank  $p$  closest to  $\Sigma$  as measured by the Frobenius norm,

$$\|\Sigma - \Sigma_p\|_F.$$

The properties of the principal components are even more useful if the underlying random variable  $Y$  has a multivariate normal distribution.

In that case, the principal components vector  $\tilde{Y}$  also has a multivariate normal distribution, and the elements of  $\tilde{Y}$  are independent.

# Principal Components Analysis of Data

In the basic multivariate data structure of  $X$ , we often consider the rows to be realizations of some multivariate random variable, such as  $Y$  in the discussion above.

Because of differences in the meanings of the variables in the data matrix  $X$ , it is best first to standardize the data:

$$X_S = (X - \bar{X}) \text{diag}(1/\sqrt{s_{ii}}).$$

In the following, we will assume that this has been done. We will assume that  $X$  has been standardized and not continue to use the notation  $X_S$ .

# Principal Components Analysis of Data

Using the sample variance-covariance matrix  $S$  as an estimate of the population variance-covariance matrix  $\Sigma$ , we can perform a principal components analysis of the data that follows the same techniques as above for a random variable.

Hence, we first determine the spectral decomposition of  $S$ :

$$S = \sum_j \hat{\lambda}_j \hat{w}_j \hat{w}_j^T.$$

The principal components of the vector of observed variables  $x$  are

$$\tilde{x}_{(j)} = \hat{w}_j^T x.$$

# Principal Components Analysis of Data

Corresponding to the generic data vector  $x$  is the generic vector of principal components,

$$\tilde{x} = (\tilde{x}_{(1)}, \dots, \tilde{x}_{(m)}).$$

For each observation  $x_i$ , we can compute a value of the principal components vector,  $\tilde{x}_i$ .

From the spectral decomposition that yields the principal components, it is easy to see that the sample variance-covariance matrix of the principal components is diagonal.

## **Principal Components Analysis of Data**

The first principal component is the hyperplane that minimizes the orthogonal distances to the hyperplane.

The principal components are transformations of the original system of coordinate axes. It is difficult to relate any of the new axes to the old axes, however.

# Dimension Reduction by Principal Components Analysis

We can reduce the dimension of the data by considering the transformed variables  $\tilde{x}_{(i)}$ , each of which is a vector formed using the eigenvectors corresponding only to the  $p$  largest eigenvalues.

As before, we form the  $p \times m$  transformation matrix  $\widehat{W}$ ,

$$\widehat{W} = \begin{bmatrix} \widehat{w}_1^T \\ \widehat{w}_2^T \\ \vdots \\ \widehat{w}_p^T \end{bmatrix}.$$

For the  $i^{\text{th}}$  observation  $x_i$ , this produces the  $p$ -vector  $\tilde{x}_i = (\tilde{x}_{i(1)}, \dots, \tilde{x}_{i(p)})$

# Dimension Reduction by Principal Components Analysis

The question is how to choose  $p$ ; that is, how much we can reduce the dimensionality of the original dataset.

A simple approach that is often employed is to choose  $p$  as the number of the ranked eigenvalues just prior to a large gap in the list.

A plot of the ordered values or of the values scaled by their total may be useful in identifying the point at which there is a large dropoff in effect. Such is called a scree plot.

The scree plot can be either a line plot as in the figure or a bar chart in which the heights of the bars represent the relative values of the eigenvalues. The key feature in a scree plot is an “elbow”, if one exists.

# Dimension Reduction by Principal Components Analysis

The effect of each of the original variables (the elements of  $x$ ) on each principal component is measured by the correlation between the variable and the principal component. This is called the “component loading” of the variable on the principal component. The component loading of the  $j^{\text{th}}$  variable on the  $k^{\text{th}}$  principal component is the correlation

$$\frac{w_{kj}\sqrt{\hat{\lambda}_k}}{\sqrt{s_{jj}}}.$$

(Note that  $w_{kj}$  is the  $j^{\text{th}}$  element of the  $k^{\text{th}}$  eigenvector.)

# Principal Components and Transformations of the Data

Variation provides information.

Variables with large sample variances will tend to predominate in the first principal component.

Consider the extreme case in which the variables are uncorrelated (that is, in which  $S$  is diagonal). The principal components are determined exactly by the variances, from largest to smallest.

This is a natural and desirable consequence of the fact that variation provides information.

In principal components analysis, the relative variation from one variable to another is the important factor in determining the rankings of the components.

# Principal Components and Transformations of the Data

It may not be meaningful to measure the relative variation from one variable to another. The variance of a variable depends on the units of measurement.

Suppose that one variable represents linear measurements in meters. If, for some reason, the unit of measurement is changed to centimeters, the effect of that variable in determining the principal components will increase one hundredfold.

The component loadings can help in understanding the effects of data reduction through principal components analysis.

Notice that the component loadings are scaled by the square root of the variance.

# Principal Components and Transformations of the Data

Another approach to scaling problems resulting from the choice of unit of measurement is to use the correlation matrix,  $R$ , rather than the variance-covariance matrix. The correlations result from scaling the covariances by the square roots of the variances.

The principal components resulting from the use of  $R$  are not the same as those resulting from the use of  $S$ .

Change of units of measurement is just one kind of simple scaling transformation. Transformations of any kind are likely to change the results of a multivariate analysis.

# Principal Components of Observations

The basic symmetry between the “variables” and the “observations” of the dataset  $X$ , we can likewise reverse their roles in principal components analysis.

Suppose, for example, that the observational units are individual persons and the variables are responses to psychological tests.

Principal components analysis as we have described it would identify linear combinations of the scores on the tests. These principal components determine relationships among the test scores.

If we replace the data matrix  $X$  by its transpose and proceed with a principal components analysis as described above, we identify important linear combinations of the observations that, in turn, identify relationships among the observational units.

In the social sciences, a principal components analysis of variables is called an “R-Type” analysis and the analysis identifying relationships among the observational units is called “Q-Type”.

## Principal Components of Observations

In the usual situation, as we have described, the number of observations,  $n$ , is greater than the number of variables,  $m$ .

If  $X$  has rank  $m$ , then the variance-covariance matrix and the correlation matrix are of full rank. In a reversal of the roles of observations and variables, the corresponding matrix would not be of full rank.

The analysis could proceed mechanically as we have described, but the available information for identifying meaningful linear combinations of the observations would be rather limited.

# Computation of Principal Components Directly from the Data Matrix

The sample variance-covariance matrix  $S$  is

$$S = \frac{1}{n-1}(X - \bar{X})^T(X - \bar{X}).$$

Formation of the  $S$  or the sample correlation matrix  $R$  emphasizes the role that the sample covariances or correlations play in principal component analysis.

However, there is no reason to form a matrix such as  $(X - \bar{X})^T(X - \bar{X})$ , and indeed we may introduce significant rounding errors by doing so.

# Computation of Principal Components Directly from the Data Matrix

The singular value decomposition (SVD) of the  $n \times m$  matrix  $X - \bar{X}$  yields the square roots of the eigenvalues of  $(X - \bar{X})^T (X - \bar{X})$  and the same eigenvectors. (The eigenvalues of  $(X - \bar{X})^T (X - \bar{X})$  are  $(n - 1)$  times the eigenvalues of  $S$ .)

We will assume that there are more observations than variables (that is, that  $n > m$ ). In the SVD of the centered data matrix, we write

$$X - \bar{X} = UAV^T,$$

where  $U$  is an  $n \times m$  matrix with orthogonal columns,  $V$  is an  $m \times m$  orthogonal matrix, and  $A$  is an  $m \times m$  diagonal matrix with nonnegative entries, called the singular values of  $X - \bar{X}$ .

# Computation of Principal Components Directly from the Data Matrix

The spectral decomposition in terms of the singular values and outer products of the columns of the factor matrices is

$$X - \bar{X} = \sum_{i=1}^m \sigma_i u_i v_i^T.$$

The vectors  $u_i$ , called the “left eigenvectors” or “left singular vectors” of  $X - \bar{X}$ , are the same as the eigenvectors of  $S$ . The vectors  $v_i$ , the “right eigenvectors”, are the eigenvectors that would be used in a Q-type principal components analysis. The reduced-rank matrix that approximates  $X - \bar{X}$  is

$$\tilde{X}_p = \sum_{i=1}^p \sigma_i u_i v_i^T$$

for some  $p < \min(n, m)$ .

## Computational Issues

For the eigenanalysis computations in PCA, if the sample variance-covariance matrix  $S$  is available, it is probably best to proceed with the decomposition of it.

If  $S$  is not available, there is generally no reason to compute it just to perform PCA. The computations to form  $S$  are  $O(m^3)$ .

Not only do these computations add significantly to the overall computational burden, but  $S$  is more poorly conditioned than  $X$  (or  $X - \bar{X}$ ). The SVD decomposition is therefore the better procedure.

# PCA for Clustering

An objective of principal components analysis is to identify linear combinations of the original variables that are useful in accounting for the variation in those original variables. This is effectively a clustering of the variables.

For many purposes, these derived features carry a large amount of the information that is available in the original larger set of variables.

For example, the information carried by the smaller set of features identified in PCA may be useless in clustering the observations. See Figures 10.21 and 10.22.

Principal components analysis emphasizes the direction of maximum variation. If the main source of variation in the dataset is the variation between clusters, then PCA will identify the clusters.

# Robustness of Principal Components

Outlying observations may have a major effect on the principal components analysis. The first few principal components are very sensitive to these outlying observations. If the outliers were not present, or if they were perturbed slightly, a different set of the first few principal components would likely result.

There are generally two ways of dealing with outlying observations.

One way is to identify the outliers and remove them temporarily. Another way is to use methods that are not much affected by outliers.

The principal components resulting from using the robust sample variance-covariance,  $S_R$ , on page 121 are less affected by outliers than those resulting from the usual sample variance-covariance,  $S$ .

# Robustness of Principal Components

If outliers can be identified and removed temporarily, a standard analysis can be performed.

This identification, removal, and analysis procedure can be applied in stages.

The major problem, of course, is that as extreme observations are removed, the variability in the dataset is reduced, so other, valid observations are more likely to appear as outliers.

In general, a data analyst must assume that every observation carries useful information, and no observation must be discarded until its information is incorporated into the analysis.

For purposes of PCA, outliers can be identified in a preliminary step using a clustering procedure or even by using Q-type principal components analysis.

# Factor Analysis

Factor analysis is mechanically similar to principal components analysis. The main differences involve the probability model.

In factor analysis, we begin with a model that relates a centered  $m$ -vector random variable  $Y$  (observable) to an underlying, unobservable  $k$ -vector random variable, whose elements are called “factors” .

The factors have a mean of 0.

In this model, the observable vector  $Y$  consists of linear combinations of the factors plus an independent random vector of “unique errors”, which is modeled by a random variable with a mean of 0.

# Factor Analysis

The unique errors are independent of the factors.

If we let  $F$  represent the vector of factors and  $E$  represent the errors, we have

$$Y - \mu = \Gamma F + E,$$

where  $\mu$  is the mean of  $Y$  and  $\Gamma$  is an  $m \times k$  fixed (but unknown) matrix, called the “factor loadings” matrix.

Generally, the number of factors is less than the number of the observable variables.

In some applications, such as in psychology, the factors may be related to some innate characteristics that are manifested in observable behavior.

## Factor Analysis

We denote the variance-covariance matrix of  $Y$  by  $\Sigma$ , that of  $F$  by  $\Sigma_F$ , and that of  $E$  by  $\Psi$ , which is diagonal by the assumptions in the model. We therefore have the relationship

$$\Sigma = \Gamma \Sigma_F \Gamma^\top + \Psi.$$

Now, if we let  $\tilde{\Gamma} = \Gamma \Sigma_F^{-\frac{1}{2}}$  and  $\tilde{F} = (\Sigma_F^{-\frac{1}{2}})^{-1} F$ , we have

$$\Sigma = \tilde{\Gamma} \tilde{\Gamma}^\top + \Psi.$$

## Factor Analysis

An equivalent model is one in which we assume that the underlying factors have the identity as their variance-covariance matrix, and so we have

$$\Sigma = \Gamma\Gamma^{\top} + \Psi.$$

The diagonal elements of  $\Psi$  are called the *specific variances* of the factors and the diagonal elements of  $\Gamma\Gamma^{\top}$  are called the *commonalities* of the factors.

The transformations above that indicate that  $\Gamma\Gamma^{\top}$  can be used instead of  $\Gamma\Sigma_F\Gamma^{\top}$  raise the issue of more general transformations of the factors, leading to an indeterminacy in the analysis.

## Factor Analysis

If we decompose  $\Sigma - \Psi$  as we did in PCA, (with  $\Delta$  replacing  $\Lambda$ ) we have

$$\Sigma - \Psi = W^T \Lambda W.$$

The factor-loading matrix therefore is

$$\Gamma = W^T \Lambda^{\frac{1}{2}}.$$

# Factor Analysis of Data

In practical applications of factor analysis, we must begin with a chosen value of  $k$ , the number of factors.

This is similar to choosing the number of principal components in PCA, and there are some ways of adaptively choosing  $k$ , but the computational approaches that we discuss below assume a fixed value for  $k$ .

As usual, we consider the rows of the data matrix  $X$  to be realizations of a multivariate random variable.

In factor analysis, the random variable has the same relationships to other random variables as  $Y$  above; hence, the observation  $x$  (a row of  $X$ ) is related to the realization of two other random variables,  $f$  and  $e$ , by

$$x - \bar{x} = \Gamma f + e.$$

# Factor Analysis of Data

The objective in factor analysis is to estimate the parameters in the model —that is, the factor loadings,  $\Gamma$ , and the variances,  $\Sigma$  and  $\Psi$ .

There are several methods for estimating these parameters.

In one method, the estimation criterion is *least squares* of the sum of the differences in the diagonal elements of  $\Sigma$  and  $S$ , that is, minimize the function  $g$ :

$$g(\Gamma, \Psi) = \text{trace}((S - \Sigma)^2).$$

This criterion leads to the *principal factors method*.

# Principal Factors Method

The minimization proceeds by first choosing a value  $\hat{\Psi}^{(0)}$  and then performing a decomposition similar to that in principal components, except that instead of decomposing the sample variance-covariance matrix  $S$ , an eigenanalysis of  $S - \hat{\Psi}^{(0)}$  is performed, yielding the value for  $\Gamma$ :

$$\hat{\Gamma}^{(0)} = (\hat{W}^{(0)})^T (\hat{\Lambda}^{(0)})^{\frac{1}{2}}.$$

Next, the minimization problem is solved for  $\Psi$  with the fixed value of  $\hat{\Gamma}^{(0)}$ .

The steps are then repeated. Convergence criteria are usually chosen based on norms of the change in the estimates from one iteration to the next.

The factors derived using the principal factors method (that is, the linear combinations of the original variables) are the same as would be obtained in ordinary PCA if the variance of the noise (the unique errors) were removed from the variance-covariance of the observations prior to performing the PCA.

# Factor Analysis of Data

Another common method for estimating  $\Gamma$ ,  $\Sigma$ , and  $\Psi$  uses the *likelihood criterion* that results from the asymptotic distributions. Using the negative of the log of the likelihood, we have the minimization problem,

$$\min l(\Gamma, \Psi) = \min(\log|\Sigma^{-1}S| - \text{trace}(\Sigma^{-1}S)).$$

This criterion results in the method of maximum likelihood.

Solution of the minimization problem is also done in iterations over two stages, as we did in the least squares method above.

An advantage of the maximum likelihood method is that it is independent of the scales of measurement.

## Factor Analysis of Data

Other common methods for factor analysis include generalized least squares, image analysis (of two different types), and alpha factor analysis.

The methods for factor analysis begin with the computation of the sample variance-covariance matrix  $S$  or the sample correlation matrix  $R$ .

As in the case of PCA, the results are different, just as the results are generally different following any transformation of the data.

# Rotations in Factor Analysis

Note that the model does not define the factors uniquely; any rotation of the factors would yield the same model.

In principal components analysis, a similar indeterminacy could also occur if we allow an arbitrary basis for the PCA subspace defined by the chosen  $k$  principal components.

The factors are often rotated to get a basis with some interesting properties.

A common criterion is parsimony of representation, which roughly means that the matrix has few significantly nonzero entries. This principle has given rise to various rotations, such as the varimax, quartimax, and oblimin rotations.

# Latent Semantic Indexing

An interesting application of the methods of principal components, called *latent semantic indexing*, is used in matching keyword searches with documents.

The method begins with the construction of a term-document matrix,  $X$ , whose rows correspond to keywords, whose columns correspond to documents (web pages, for example), and whose entries are the frequencies of occurrences of the keywords in the documents.

A singular value decomposition is performed on  $X$  (or on  $X - \bar{X}$ ), and then a reduced-rank matrix  $\widetilde{X}_p$  is defined.

A list of keywords is matched to documents by representing the keyword list as a vector,  $q$ , of 0's and 1's corresponding to the rows of  $X$ .

The vector  $\widetilde{X}_p^T q$  is a list of scores for the documents. Documents with larger scores are those deemed relevant for the search.

# Latent Semantic Indexing

A semantic structure for the set of documents can also be identified by  $\widetilde{X}_p$ . Semantically nearby documents are mapped onto the same singular vectors.

A variation of latent semantic indexing is called probabilistic latent semantic indexing, or nonnegative-part factorization.

This approach assumes a set of hidden variables whose values in the matrix  $H$  correspond to the columns of  $X$  by a *nonnegative matrix factorization*,

$$X = WH,$$

where  $W$  is a matrix with nonnegative elements.

The relationship of the model in probabilistic latent semantic indexing to the standard latent semantic indexing model is similar to the differences in factor analysis and principal components analysis.

# Linear Independent Components Analysis

Independent components analysis (ICA) is similar to principal components analysis and factor analysis. Both PCA and ICA have nonlinear extensions.

In linear PCA and ICA, the objective is to find a linear transformation  $W$  of a random vector  $Y$  so that the elements of  $WY$  have small correlations.

In linear PCA, the objective then is to find  $W$  so that  $V(WY)$  is diagonal, and, as we have seen, this is simple to do.

If the random vector  $Y$  is normal, then 0 correlations imply independence.

The objective in linear ICA is slightly different; instead of just the elements of  $WY$ , attention may be focused on chosen transformations of this vector, and instead of small correlations, independence is the goal.

## Linear Independent Components Analysis

The transformations of  $WY$  are often higher-order sample moments. The projections that yield diagonal variance-covariance matrices are not necessarily orthogonal.

In the literature on ICA, which is generally in the field of signal processing, either a “noise-free ICA model”, similar to the simple PCA model, or a “noisy ICA model”, similar to the factor analysis model, is used. Most of the research has been on the noise-free ICA model.

# Projection Pursuit

The objective in projection pursuit is to find “interesting” projections of multivariate data.

Interesting structure in multivariate data may be identified by analyzing projections of the data onto lower-dimensional subspaces. The projections can be used for optimal visualization of the clustering structure of the data or for density estimation or even regression analysis. (The approach is related to the visual approach of the grand tour.)

Reduction of dimension is also an important objective, especially if the use of the projections is in visualization of the data.

Projection pursuit requires a measure of the “interestingness” of a projection.

# Interestingness of Data

A randomly selected projection of a high-dimensional dataset onto a low-dimensional space will tend to appear similar to a sample from a multivariate normal distribution with that lower dimension.

This fact, which may be thought of as a central limit theorem for projections, implies that a multivariate normal dataset is the least “interesting”.

A specific projection of the given dataset, however, may reveal interesting features of the dataset.

In projection pursuit, therefore, the objective is to find departures from normality in linear projections of the data.

Departures from normality may include such things as skewness and “holes” in the data, or multimodality.

# The Probability Model Underlying Projection Pursuit

Consider an  $m$ -vector random variable  $Y$ . In general, we are interested in a  $k$ -dimensional projection of  $Y$ , say  $A^T Y$ , such that the random variable  $A^T Y$  is very different from a  $k$ -variate normal distribution.

Because all one-dimensional marginals of a multivariate normal are normal, and cross products of normals are multivariate normal, we will concentrate on one-dimensional projections of  $Z$ .

We want to find  $Z = a^T Y$  such that the random variable  $Y$  is “most different” from a normal random variable. (Two-dimensional projections are of particular interest, especially in graphics.)

# The Probability Model Underlying Projection Pursuit

The structure of interest (that is, a departure from normality) can be considered separately from the location, variances, and covariances of the vector  $Y$ ; therefore, we will assume that  $E(Y) = 0$  and  $V(Y) = I$ .

Prior to applying projection pursuit to data, we center and sphere the data so that the sample characteristics are consistent with these assumptions.

To quantify the objectives in projection pursuit, we need a measure, or index, of the departure from normality.

# Projection Indexes for the Probability Model

One way to quantify departure from normality is to consider the probability density function of the projected variable and compare it to the probability density function  $\phi$  of a standard normal random variable.

If  $p$  is the density of  $Z$ , we want it to be very different from  $\phi$ . This is an opposite problem from the function approximation problem, but the approaches are related.

Whereas in function approximation, the Chebyshev norm is generally of most interest in seeking a function that is “different”, an  $L_2$  norm,

$$\int_{-\infty}^{\infty} (p(z) - \phi(z))^2 dz,$$

makes more sense as a measure of the difference.

## Projection Indexes for the Probability Model

The objective in projection pursuit is to find an  $a$  that maximizes this norm or index.

A standard approach is to use orthogonal polynomials to approximate the index, and to name the index after the type of orthogonal polynomial used.

The index considered before is called the *Hermite index* because Hermite polynomials are appropriate for approximation over the unbounded domain.

# Projection Indexes for the Probability Model

Another way of defining an index for a given  $a$ , is first to map  $Z = a^T Y$  into  $[-1, 1]$  by the transformation

$$R = 2\Phi(Z) - 1,$$

where  $\Phi$  is the CDF of a standard normal distribution.

If  $p_Z$  is the probability density of  $Z$ , then the probability density of  $R$  is

$$p_R(r) = \frac{\frac{1}{2}p_Z\left(\Phi^{-1}\left(\frac{r+1}{2}\right)\right)}{\phi\left(\Phi^{-1}\left(\frac{r+1}{2}\right)\right)}.$$

If  $Z$  has a normal distribution with a mean of 0 and variance of 1,  $R$  has a uniform distribution over  $(-1, 1)$  and so has a constant density of  $\frac{1}{2}$ .

Hence, the problem is to find  $a$  such that the density,  $p_R$ , of  $R$  is very different from  $\frac{1}{2}$ .

## Projection Indexes for the Probability Model

The relevant  $L_2$  norm is

$$L(a) = \int_{-1}^1 \left( p_R(r) - \frac{1}{2} \right)^2 dr,$$

which simplifies to

$$L(a) = \int_{-1}^1 p_R^2(r) dr - \frac{1}{2}.$$

This norm, which is a scalar function of  $a$  and a functional of  $p_R$ , is sometimes called the *Legendre index* because Legendre polynomials are natural approximating series of orthogonal polynomials for functions over finite domains

# Projection Indexes for the Probability Model

Various other measures of departure from normality are possible, for example, an index based on ratios of moments of a standard normal distribution, or indexes based on entropy (called Shannon entropy or differential entropy):

$$- \int_{\mathbb{R}^m} p(z) \log p(z) dz.$$

The entropy is maximized among the class of all random variables when the density  $p$  is the standard multivariate normal density (mean of zero and variance-covariance matrix equal to the identity).

For any other distribution, the entropy is strictly smaller.

Other kinds of measures of departure from normality can be contemplated.

Almost any goodness-of-fit criterion could serve as the basis for a projection index.

# Projection Pursuit in Data

We now consider one-dimensional projection pursuit in a given set of data  $X$  (the familiar  $n \times m$  matrix in our data analysis paradigm). For each projection  $a$ , we *estimate* the projection index associated with  $a$  under the assumption that the rows in  $X$  are independent realizations of a random variable  $Y$ . The vector  $Xa$  contains independent realizations of the scalar random variable  $Z = a^T Y = Y^T a$ .

The question is how similar the distribution of  $Z$  is to a normal distribution. The problem with measures of departure from normality is the difficulty in estimating the terms.

To estimate the projection index, we must *approximate* an integral.

The indexes lend themselves to approximation by standard series of orthogonal polynomials.

# Projection Pursuit in Data

For  $L(a)$ , expanding one factor of  $p_R^2$  in the equation in Legendre polynomials and leaving the other unexpanded, we have

$$L(a) = \int_{-1}^1 \left( \sum_{k=0}^{\infty} c_k P_k(r) \right) p_R(r) dr - \frac{1}{2},$$

where  $P_k$  is the  $k^{\text{th}}$  Legendre polynomial. We have the Legendre coefficients for the expansion

$$c_k = \frac{2k + 1}{2} \int_{-1}^1 P_k(r) p_R(r) dr.$$

Substituting this into the expression above, because of the orthogonality of the  $P_k$ , we have

$$L(a) = \sum_{k=0}^{\infty} \frac{2k + 1}{2} (\mathbb{E}(P_k(R)))^2 - \frac{1}{2},$$

where the expectation  $\mathbb{E}$  is taken with respect to the distribution of the random variable  $R$ .

## Projection Pursuit in Data

Each term in the equation is an expectation and therefore can be estimated easily from a random sample.

The sample mean is generally a good estimate of an expectation; hence, for the  $k^{\text{th}}$  term, from the original observations  $x_i$ , the projection  $a$ , and the normal CDF transformation, we have

$$\begin{aligned}\hat{E}(P_k(R)) &= \frac{1}{n} \sum_{i=1}^n P_k(r_i) \\ &= \frac{1}{n} \sum_{i=1}^n P_k(2\Phi(a^\top x_i) - 1).\end{aligned}$$

A simple estimate of the squared expectation is just the square of this quantity.

## Projection Pursuit in Data

Obviously, in practice, we must use a finite approximation to the infinite expansion of  $p_R$ . After terminating the expansion at  $j$ , we have the truncated Legendre projection index,  $L_j(a)$ ,

$$L_j(a) = \sum_{k=0}^j \frac{2k+1}{2} (\mathbb{E}(P_k(R)))^2 - \frac{1}{2}.$$

The approximation can be estimated easily from the sample:

$$\hat{L}_j(a) = \frac{1}{2n^2} \sum_{k=0}^j (2k+1) \left( \sum_{i=1}^n P_k(2\Phi(a^\top x_i) - 1) \right)^2 - \frac{1}{2}.$$

This expression is easily evaluated.

# Projection Pursuit in Data

The problem now is to determine

$$\max_a \hat{L}_j(a).$$

Scaling of  $a$  is not relevant, so we may restrict  $a$  so that the sum of its elements is some given value, such as 1. In general, this is not an easy optimization problem.

There are local minima.

Use of an optimization method such as Newton's method may require multiple starting points. An optimization method such as simulated annealing may work better.

Although the more terms retained in the orthogonal series expansions, the better is the approximation, it is not necessarily the case that the better-approximated index is more useful.

# Exploratory Projection Pursuit

The most important use of projection pursuit is for initial exploratory analysis of multivariate datasets.

Different indexes may be useful in identifying different kinds of structure.

The Legendre index is very sensitive to outliers in the data. If identification of the outliers is of specific interest, this may make the index useful.

On the other hand, if the analysis should be robust to outliers, the Legendre index would not be a good one.

The Laguerre-Fourier index, which is based on an expansion in Laguerre polynomials, is particularly useful in identifying clusters in the data.

## Computational Issues

Projection pursuit involves not only the computation of an index but the optimization of the index as a function of the linear combination vector. This approach is therefore computationally intensive.

The optimization problem is characterized by many local maxima. Rather than being interested in a global maximum, in data analysis with projection pursuit, we are generally interested in inspecting several projections, each of which exhibits an interesting structure—that is, some locally maximal departure from normality as measured by a projection index. This also adds to the computational intensity.

# Higher Dimensions

Many properties of one- and two-dimensional objects (lines and planes) carry over into higher-dimensional space just as we would expect.

Although most of our intuition is derived from our existence in a three-dimensional world, we generally have no problem dealing with one- or two-dimensional objects.

There are many situations, however, in which our intuition derived from the familiar representations in one-, two-, and three-dimensional space leads us completely astray.

This is particularly true of objects whose dimensionality is greater than three, such as volumes in higher-dimensional space.

## Higher Dimensions

The problems of understanding data in higher dimension is not just with our intuition, however; it is indeed the case that some properties do not generalize to higher dimensions.

The *shape* of a dataset is the total information content that is invariant under translations, rotations, and scale transformations.

Quantifying the shape of data is an interesting problem.

# Data Sparsity in Higher Dimensions

We measure space both linearly and volumetrically. The basic cause of the breakdown of intuition in higher dimensions is that the relationship of linear measures to volumetric measures is exponential in dimensionality.

The cubing we are familiar with in three-dimensional space cannot be used to describe the relative sizes of volumes (that is, the distribution of space).

Volumes relative to the linear dimensions grow very rapidly.

There are two consequences of this. One is that the volumes of objects with interior holes, such as thin boxes or thin shells, are much larger than our intuition predicts. Another is that the density of a fixed number of points becomes extremely small.

## **Data Sparsity in Higher Dimensions**

The density of a probability distribution decreases as the distribution is extended to higher dimensions by an outer product of the range.

This happens fastest going from one dimension to two dimensions but continues at a decreasing rate for higher dimensions.

The effect of this is that the probability content of regions at a fixed distance to the center of the distribution increases; that is, outliers or isolated data points become more common.

# Data Sparsity in Higher Dimensions

This is easy to see in comparing a univariate normal distribution with a bivariate normal distribution. If  $X = (X_1, X_2)$  has a bivariate normal distribution with mean 0 and variance-covariance matrix  $\text{diag}(1, 1)$ ,

$$\Pr(|X_1| > 2) = 0.0455,$$

whereas

$$\Pr(\|X\| > 2) = 0.135.$$

The probability that the bivariate random variable is greater than two standard deviations from the center is much greater than the probability that the univariate random variable is greater than two standard deviations from the center.

The consequence of these density patterns is that an observation in higher dimensions is more likely to appear to be an outlier than one in lower dimensions.

# Volumes of Hyperspheres and Hypercubes

It is interesting to compare the volumes of regular geometrical objects and observe how the relationships of volumes to linear measures change as the number of dimensions changes. Consider, for example, that the volume of a sphere of radius  $a$  in  $d$  dimensions is

$$\frac{a^d \pi^{d/2}}{\Gamma(1 + d/2)}.$$

The volume of a superscribed cube is  $(2a)^d$ . Now, compare the volumes. Consider the ratio

$$\frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)}.$$

For  $d = 3$ , this is 0.524; for  $d = 7$ , however, it is 0.037. As the number of dimensions increases, more and more of the volume of the cube is in the corners.

# Volumes of Geometric Objects

For two objects of different sizes but the same shape, with the smaller one centered inside the larger one, we have a similar phenomenon of the content of the interior object relative to the larger object.

The volume of a thin shell as the ratio of the volume of the outer figure (sphere, cube, whatever) is

$$\frac{V_d(r) - V_d(r - \epsilon)}{V_d(r)} = 1 - \left(1 - \frac{\epsilon}{r}\right)^d.$$

As the number of dimensions increases, more and more of the volume of the larger object is in the outer thin shell.

This is the same phenomenon that we observed above for probability distributions.

# The Curse of Dimensionality

The computational and conceptual problems associated with higher dimensions have often been referred to as “the curse of dimensionality”. How many dimensions cause problems depends on the nature of the application.

In higher dimensions, not only do data appear as outliers, but they also tend to lie on lower dimensional manifolds. This is the problem sometimes called “multicollinearity”.

The reason that data in higher dimensions are multicollinear, or more generally, concurve, is that the number of lower dimensional manifolds increases very rapidly in the dimensionality: the rate is  $2^d$ .

# The Curse of Dimensionality

Whenever it is possible to collect data in a well-designed experiment or observational study, some of the problems of high dimensions can be ameliorated. In computer experiments, for example, Latin hypercube designs can be useful for exploring very high dimensional spaces.

Data in higher dimensions carry more information in the same number of observations than data in lower dimensions. Some people have referred to the increase in information as the “blessing of dimensionality” .

The support vector machine approach in fact attempts to detect structure in data by mapping the data to higher dimensions.