Interlude-Box Plot

* Tend to be bewildering to many
  + Hard to interpret
* May give wrong representation of data
  + Assume normal distributed data

Density Plot

* Multiple classes

Lollipop Charts

* Makes it easier to see and compare positions than scatter plots

High-D Objects

* Features vectors are typically high dimensional
  + Many elements in a vector
* Is tricky, difficult to understand

Curse of Dimensionality

* Points are all at about the same distance from one another
  + Concentration of distances
  + As n increases, it is impossible to distinguish two points by (Euclidian) distance
    - Unless they are in the same cluster

Sparseness Demonstration

* Space gets extremely sparse
  + With every extra dimension points get pulled apart further
  + Distances becomes meaningless
* Chart, radar chart

  Description automatically generated

Space and Memory Management

* Indexing (and storage) also gets very expensive
  + Exponential growth in the number of dimensions
  + 4D: 65k cells 5D: 1M cells 6D: 16M cells 7D: 268M cells

Scatter plot

* Projection of the data items into a bivariate basis of axes
* More than two variables
  + Scatterplot matrix
    - Difficult to see multivariate relationship

Biplots

* 2D projection
* A basis of two orthogonal axis vectors defined in N-D space
  + A = {a1, a2, …, an}
  + B = {b1,b2, … ,bn}
* A projection {xa, xb} of x into the 2D basis spanned by {a, b} is
* Plots data and dimension axes into single visualization
  + Uses first two PCA vectors as basis to project into
* Projection may not be fully accurate, projection ambiguity
  + Always checkout the PCA scree plot to gauge accuracy
  + May suggest false relationship due to dimension reduction

Projection Ambiguity

* Causes inaccuracies
  + Close neighbors in the projections may not be close neighbors in the original higher dimensional space

Meet the subspace voyager

* Decomposes high-D data spaces into lower D subspaces by
  + Clustering
  + Classification
  + Reducing clusters to intrinsic dimensionality via local PCA
* Allow users to interactively explore these lower-D subspaces
  + Explore them as a chain of 3D subspaces
  + Transition seamlessly to adjacent 3D subspaces on demand

Multidimensional Scaling (MDS)

* MDS preserves similarity relationships, prevents ambiguity
  + Scattered points in high-dimensions
  + Adjacency matrices
* Maps the distances between observations from N-D into low-D
  + Attempts to ensure the differences between pairs of points in this reduced space match as closely as possible
* The input to MDS is a distance (dissimilarity) matrix
  + Use the dissimilarity matrix because you want similar points mapped closely
  + Dissimilar point pairs will have greater values and map further apart
* Summarization
  + Task: find the configuration of image points whose pairwise distances are most similar to the original inter-point distance
  + Formally
    - Define:
    - Claim
  + In general: an exact solution is not possible
  + Inter point distance -> invariance features
* Algorithm
  + Iterative procedure to find a good configuration of image points
    - Initialization
      * Begin with some random initial configuration
    - Alter the image points and try find a configuration of points that minimizes the following sum-of-squares error function
* Uses
  + Distance Metric
    - Best for data
      * Euclidian distance
      * Cosine distance
    - Best for attribute
      * 1-|correlation| distance
      * Use abs if do not care about positive or negative correlations
      * Leave off abs if want positively correlated attribute points closer
  + Diagram

    Description automatically generated

Distance Matrix

* MDS turns a distance matrix into a network or point cloud
  + Correlation, cosine, Euclidian, and so on
* Suppose you know a matrix of distance among cities
* Result of MDS compare with real map, the graph are skewed

Force-Directed Algorithm

* Spring like system
  + insert springs within each node
  + the length of the spring encodes the desired node distance
  + start at an initial configuration
  + iteratively move nodes until an energy minimum is reached

Parallel Coordinates

* Grouping the data into sub population
* Patterns -> correlation
  + Diagram

    Description automatically generated
* How to order axes
  + Create a correlation matrix
  + Run a mass spring model
  + Run traveling salesman on the correlation nodes
  + Use it to order the axes
  + Vertices are attributes, edges are correlations
    - Vertex: size determined by
    - Edge length is measure of (1-|correlation|)
    - Edge: color/intensity -> sign/strength of correlation
* Correlation strength can often be improved by constraining a variable’s value range

Start Coordinate

* Coordinate system based on axes positioned in a “star”, or circular pattern
  + No prior PCA and subsequent projection
  + A point P is plotted as a vector sum of all axis coordinates
* Operations defined on Star Coords
  + Scaling changes contribution to resulting visualization
  + Axis rotation can visualize correlations
  + Used to reduce projection ambiguity
* RadViz
  + Diagram

    Description automatically generated

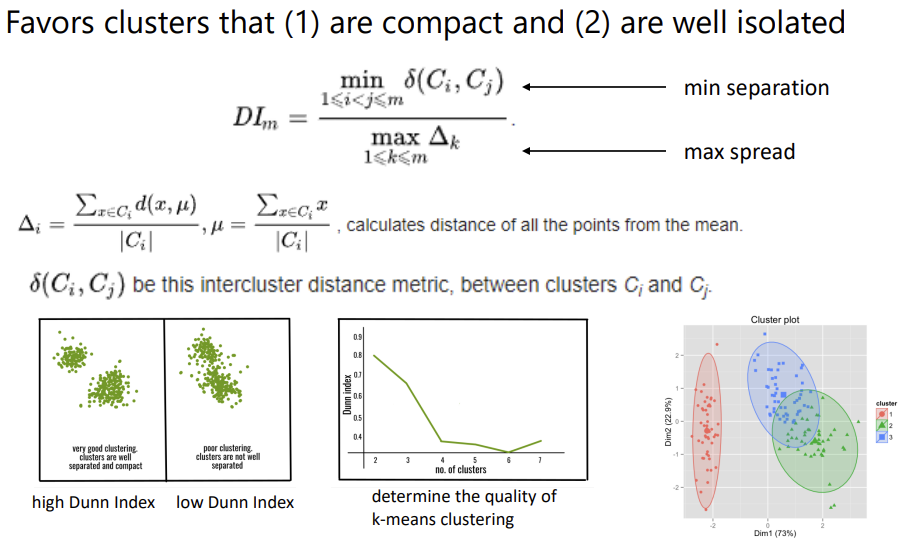
Radar Chart

* Equivalent to a parallel coordinates plot,, with the axes arranged radially
  + Each star represents a single observation
  + Can show outliers an commonalities nicely
* Disadvantage
  + Hard to make trade-off decisions
  + Distorts data to some extents when lines are filled in

Automated Scatterplot Selection

* Distance Consistency
  + Measures how “pure” a cluster is
  + Pick the views with highest normalized DSC

DUNN Index

* Favors cluster that are compact and are well isolated
* 

SC agnostics

* Describe scatterplot features by graph theoretic measures
  + Mostly build on minimum spanning tree
  + Can be used to summarize large sets of scatterplots