CSCI E-82
Advanced Machine Learning, Data Mining & Artificial Intelligence
Lecture 7

Clustering
Classification

Peter V. Henstock
Fall 2018

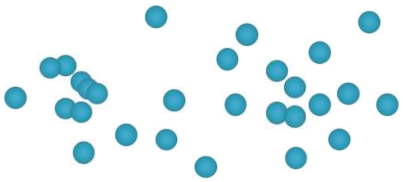
Paper Summary

- Placing an assignment for a paper summary on Canvas soon
- 5 minute recorded presentation
  - Private link on YouTube
- Step 1: Choose a paper and we'll post
  - (possibly related to your final project)
  - Focus will be extensions of research
  - Should have a results section comparing to others in the field
- Step 2: Record video

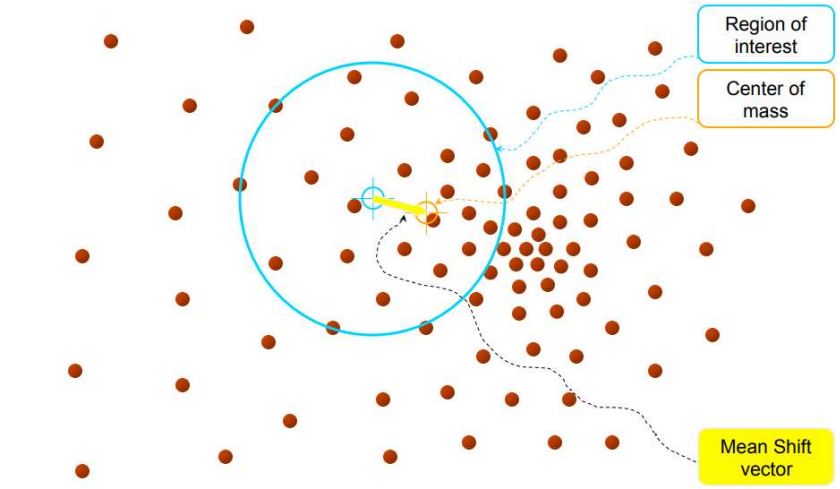
Mean Shift Clustering

Mean Shift Clustering Concept

- Start with random sets of points
- Track each until they “find” a high density region indicative of a cluster
- Segment the space of starting points by the final destinations

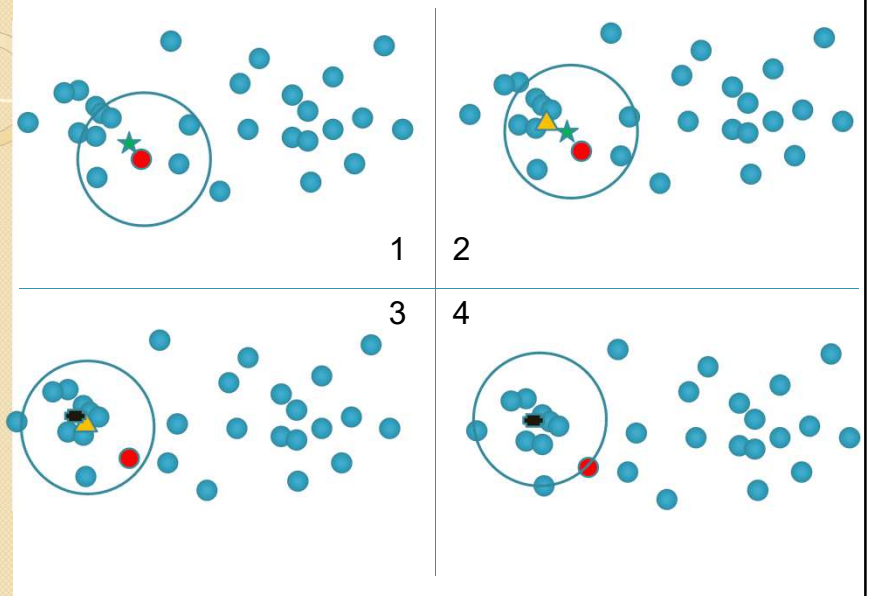


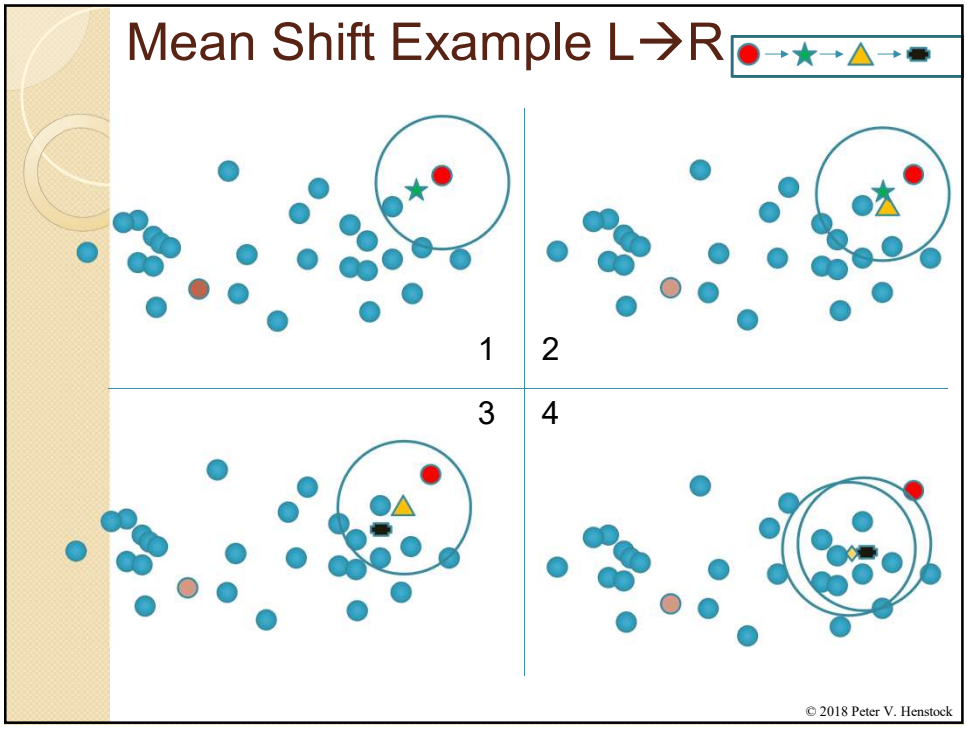
Mean Shift Concept

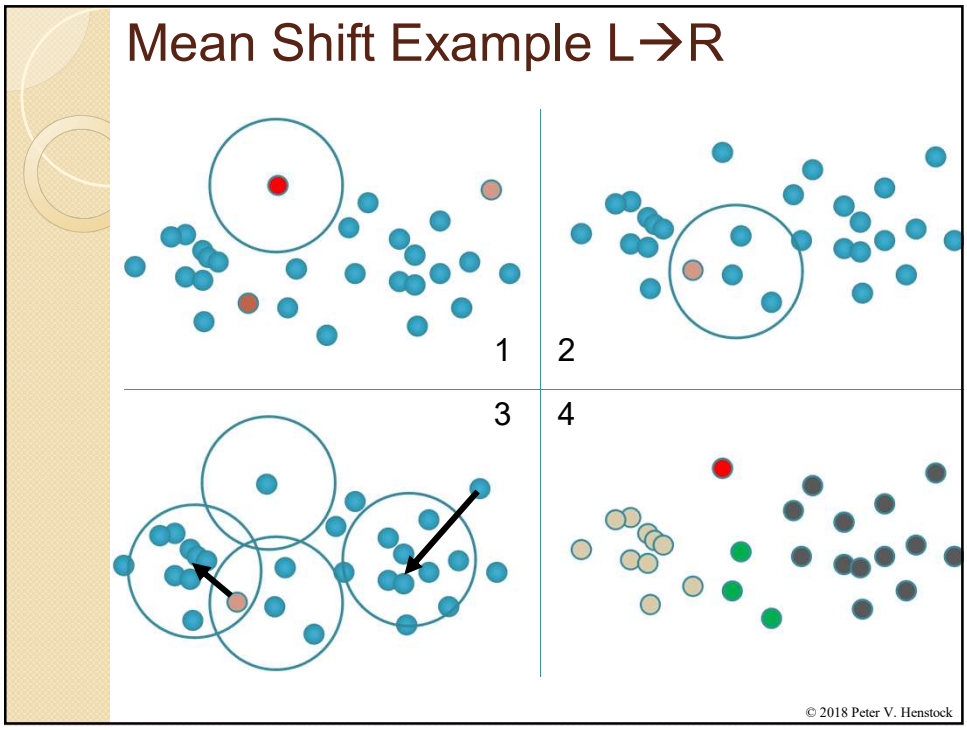


Slide by Y. Ukrainitz & B. Sarel

Mean Shift Example

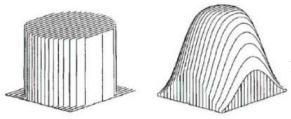



Mean Shift Example L→R
1
2
3
4
© 2018 Peter V. Henstock


Mean Shift Example L→R
1
2
3
4
© 2018 Peter V. Henstock

Math behind Mean Shift

- Kernel function K(x) describes contribution of a given point x to the mean
- K(x) is usually some $f(||x||^2)$
  - f() is nonnegative
  - f() is nonincreasing $f(x) \geq f(y)$ if $x < y$
  - F() is piecewise continuous
  - $\int_{0 \text{ to } \infty} f(x)dx < \infty$
- Flat circle: $K(x) = 1$ if $||x|| \leq$ radius else 0
- Gaussian: $K(x) = \exp(-||x||^2)$



(a) Flat kernel (b) Gaussian kernel

https://www.slideshare.net/sandtouch/meanshift-tracking-presentation

Math behind Mean Shift

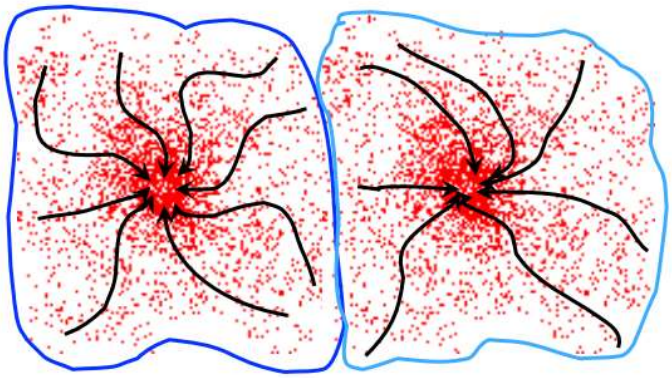
- Kernel function K(x) describes contribution of given point x to the mean
- Sample mean $m(x)=\Sigma_i x_i K(x-x_i) / \Sigma_i K(x-x_i)$
  - What does this equation remind you of?
- Mean shift is m(x) – x
- Mean shift algorithm:
  - move x → m(x)
  - Repeat until m(x) = x
- Trajectory is x, m(x), m(m(x))…

Properties of Mean Shift

- Converges on peak changing speed
  - Similar to a gradient descent where slow down near the peak or valley, only automatically
  - Essentially performing steepest ascent
- Converges but only if you take mathematically small steps that aren't useful
- Uniform kernel faster but may jump if the density changes abruptly
- Normal kernel slower but smooth

Resulting cluster concept

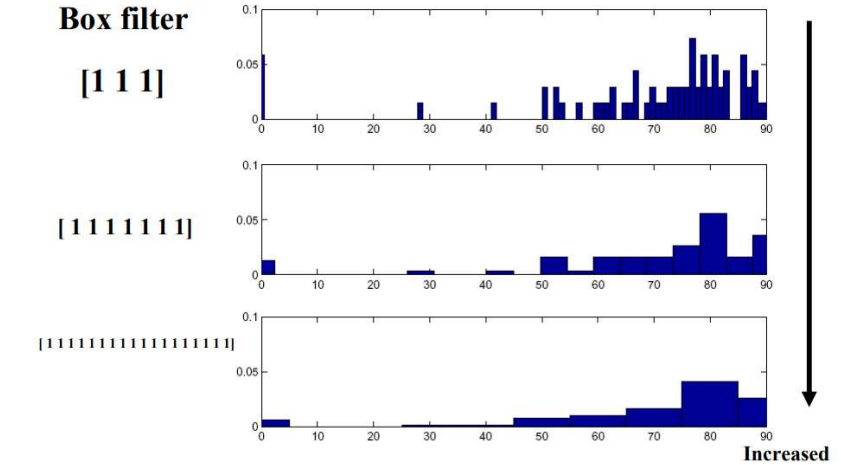
Put starting points all over the image and compute their destinations



Y. Ukrainitz and B. Sarel

How large a circle?

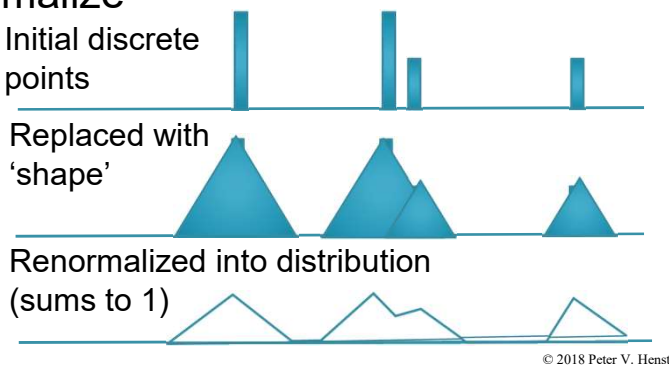
- Same issue as “k”-means
- Related to the topic of Parzen windows



http://www.cse.psu.edu/~rtc12/CSE598G/introMeanShift.pdf

Parzen Window

- Take a distribution and smooth it
- Take a distribution and bin with wide/narrow bins
- Replace each point with a ‘shape’ or kernel
- Renormalize



Parzen Window → Kernel

- Kernel Density Estimation:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

- h is the bandwidth or radius for binning
- d is the dimensionality
- Discretizing data within bins of size h
- Applying a K() function to the buckets

- What if we differentiate f?

Differentiating Kernel Density

- Obtain the change of density
- $x_i = x_{i-1} + \alpha f'(x_i)$

- If we carry out the math on the f'(x) and set =0
- Then obtain mean shift equation

https://saravananthirumuruganathan.wordpress.com/2010/04/01/introduction-to-mean-shift-algorithm/

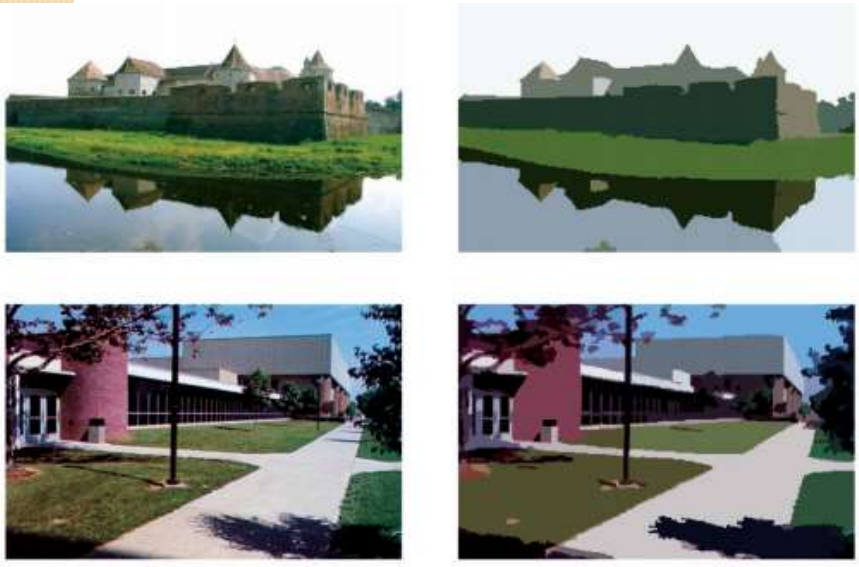
Comparison to K-Means

- K-means assumes know K
  - Mean shift does not assume this
- Speed
  - K-means is O(#clusters #points #iterations)
  - Mean shift is O(#points^2 #iterations)
  - #clusters << #points so Mean shift slower
- Sensitivity
  - K-means is sensitive to initializations
  - Mean shift is sensitive to the radius
- K-means leads to spherical clusters
- Mean shift is robust to outliers

Applying for Image Segmentation

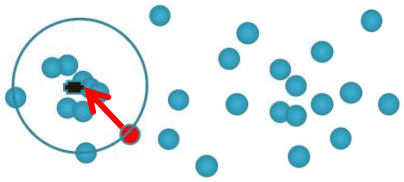
- Adapt the algorithm typically for color
- Normalize color
  - Red = red/sum(red, green, blue)
  - Green = green/ sum(red, green, blue)
  - Blue = blue / sum(red, green, blue)
  - Removes brightness or illumination issues
- Augment mean-shift so that only include pixels of same color – extra parameter
  - Requires a similarity threshold
  - Threshold often called “range resolution parameter”

Used for Image Segmentation



Dorin Comanciciu. Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE PAMI Vol 24. No. 5 May 2002.

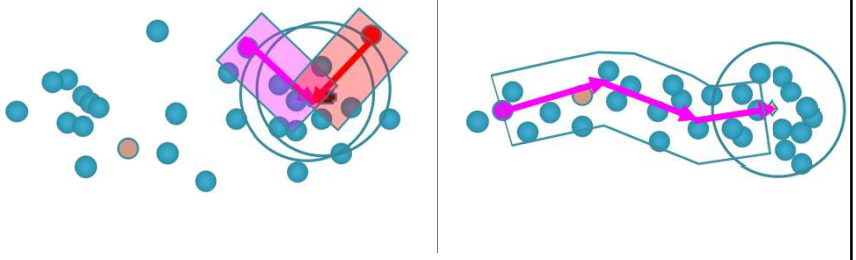
How Fast is the Mean-Shift?



- Each point needs to 'shift' to center
  - Shifting requires multiple iterations
- How would you speed this up?

How Fast is the Mean-Shift?

- Depends on the number of starting points
- Issue is that there is certain redundancy
- Pass right next to 2-3 other points
- Speed-up is to include all points within distance of the trajectory or within each radius of stop points to reduce search space



Hierarchical K-Means & BIRCH

Hierarchical K-Means

- Hierarchical Clustering Approach for Large Compound Libraries Bocker et al. 2005
- http://pubs.acs.org.ezp-prod1.hul.harvard.edu/doi/pdf/10.1021/ci0500029
- Perform K-means at each level of tree
- Repeat until avg intra-cluster-dist < threshold
  - Perform K-Means clustering on members
- Produce a K-means dendrogram

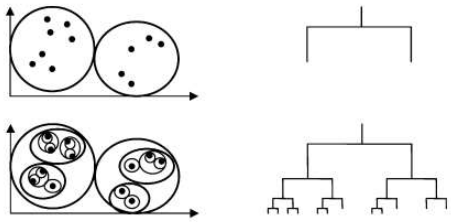
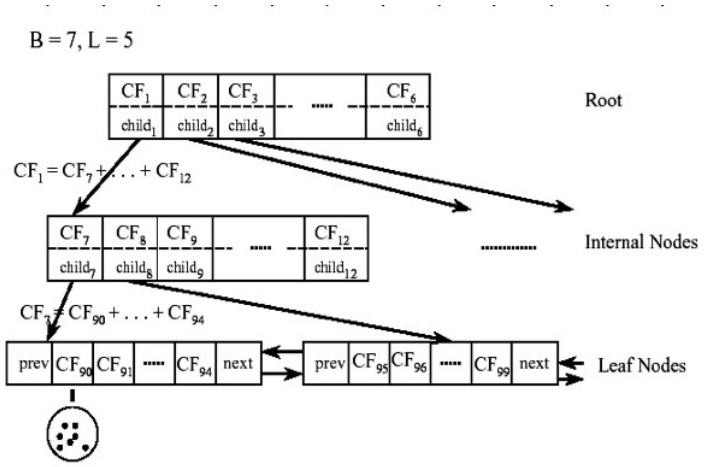


Figure 1. Example of hierarchical clustering ($k = 2$). Data objects (left) are represented by a hierarchical tree structure (right).

Weakness of Hierarchical

- Doesn't scale well
  - Often start with a N^2 distance matrix where N is the number of objects
  - Could use RNN that scales better
- Greedy approach: can't undo steps

BIRCH Feature Tree



BIRCH 1996 (Zhang, et al.)

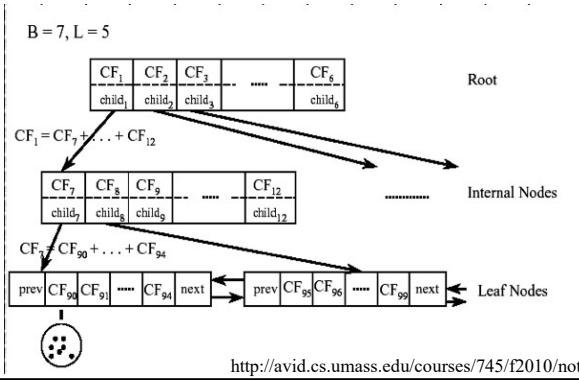
- Balanced Iterative Reducing and Clustering Using Hierarchies
- Step 1: Micro-clustering (low level)
  - Build Clustering Feature tree
  - Small enough so fits in memory as governed by a threshold (remove outliers, etc)
- Step 2: Macro clustering (high level)
  - Cluster leaf nodes into tree using arbitrary method
- Roughly linear so scales well

Clustering Feature Tree

- Has three parts:
  - N = #data points in cluster
  - LS = linear sum of points for each feature
  - SS = sum squared of points for each feature
- Three parts are 3 moments:
- Statistical interpretation:
  - Centroid = LS/N
  - Radius = sqrt($\Sigma (x_i-\mu)^2/n$) = avg dist to centroid
  - Diameter = avg pairwise dist in cluster

Tree Structures

- Lots of different types of structures that have been used for database research as well as clustering
- B-tree, B+tree, KD-tree, etc
- CF-tree has set of CFs at each level



http://avid.cs.umass.edu/courses/745/f2010/notes/DataMining.htm

BIRCH algorithm

- Input 2 parameters
  - Maximum # children of node
  - Maximum diameter of subcluster
- Start off with CF node
- Add points to closest leaf in tree
  - Remember: know diameter of the cluster
  - If cluster diameter > max_diameter
    - Split it
    - Move up to parents to balance if necessary
  - Check #children in node & split if necessary

Issues with BIRCH

- Tree construction depends on order of presentation of the data
  - Strange input patterns to distort the tree
- Diameter criteria restricts shape of nodes
  - Tend to be small and spherical
- Different cluster sizes may not be represented well
  - Due to size parameter of nodes

Extreme Clustering (PERCH)

"Extreme": Hierarchical for massive N & K

- PERCH
  - "Hierarchical Algorithm for Extreme Clustering" Kobren et al. 2017

- Features:
  - Collapsible nodes: not all in memory
  - Balanced tree
  - Bounding boxes for NN approximations
  - Dendrogram purity vs. speed
  - Guarantees

PERCH Concept

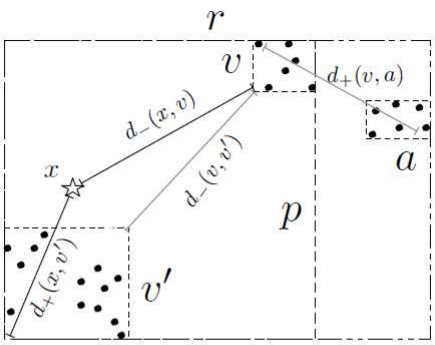
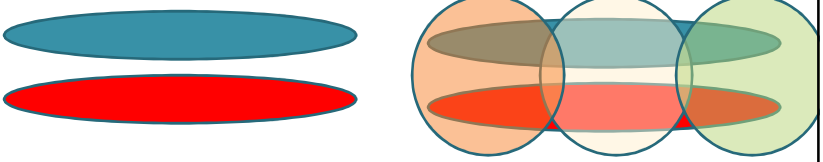
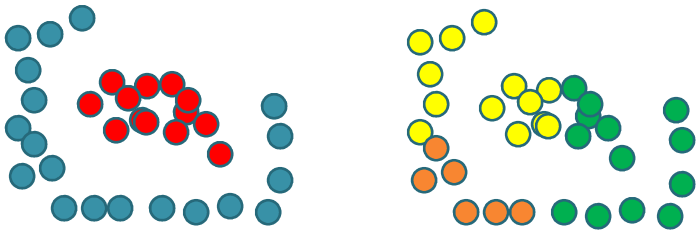


Figure 2: A subtree r with two children p and a; p is the parent of v and v' (all nodes indicated by boxes with dashed outlines). A point x is inserted and descends to v' because $d_+(x,v') < d_-(x,v)$ (black lines). The node v is masked because $d_+(v,a) < d_-(v,v')$ (gray lines).

Connectivity

Problem with Distances Alone

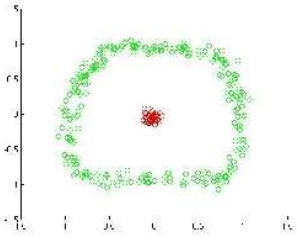




Connectivity

Connectivity of mutual k-nearest neighbor graph in clustering and outlier detection. Chavez et al. 1997

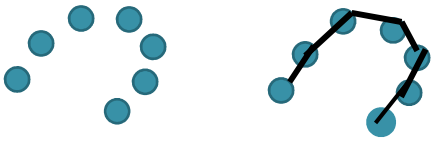
Kernel K-Means Clustering



- https://sites.google.com/site/dataclusteringalgorithms/kernel-k-means-clustering-algorithm
- Generic distances won't work on this
- Not linearly separable classes
- Kernel projects data into higher dimension and then applies K-Means

K-NN Graph

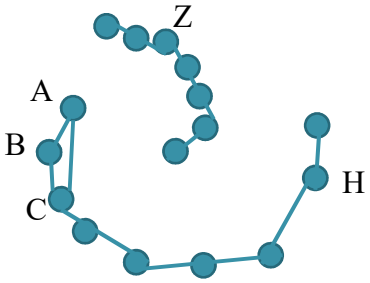
- Introduction of graph-based metric rather than a distance based metric
- Why?



- K-NN Graph:
  - Two points u and v are connected if u is among the top-k closest neighbors of v
  - K=2

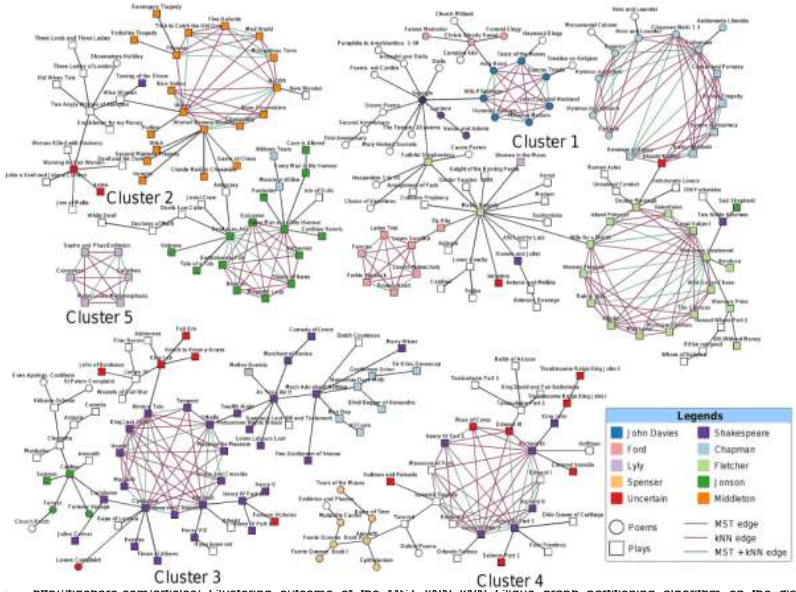
- Research on fast approaches
  - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.150.9547&rep=rep1&type=pdf

Motivating Example



- Dist (A,C) = 1
- Dist(A,B) = 1
- Dist(B,C) = 1
- Dist(C,H) = 5
- Dist(C,Z) = ∞

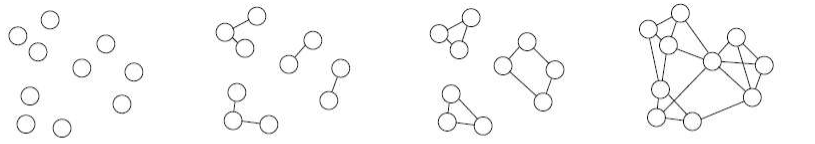
K-NN and MST for words



- http://figshare.com/articles/_Clustering_outcome_of_the_MST_kNN_kNN_Clique_graph_partitioning_algorithm_on_the_distance_matrix_produced_by_using_pair_wise_Jensen_Shannon_divergence_of_the_works_token_frequencies_/1218611

Examples of k-NN Graphs

Different values of k



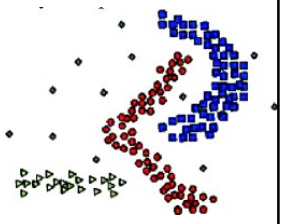
(a) Original Data in 2D (b) 1-nearest neighbor graph (c) 2-nearest neighbor graph (d) 3-nearest neighbor graph

Figure 7: *k*-nearest graphs from an original data in 2D.

Modify distance criteria

- K-means and hierarchical use distance
- Distance becomes less useful as number of dimensions increases
- Distance fails for irregular shapes

- Idea is to use topology not distance
- Apply #hops instead of Euclidian distance
- Use similar algorithms



CHAMLEON

CHAMLEON 1999 Karypis et al.

- Graph partitions into many small clusters using a moderate value of k-NN
- Divide partitions into smaller clusters
- Merges small clusters in bottom-up manner
  - Relative interconnectivity
  - Relative closeness

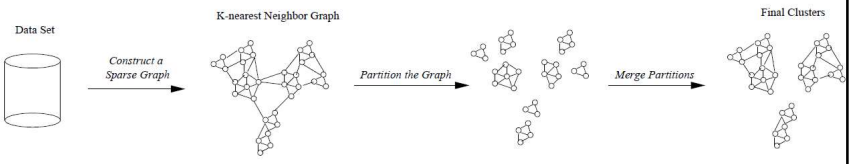


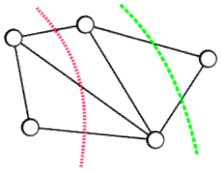
Figure 6: Overall framework CHAMELEON.

Two criteria

- Relative Interconnectivity (normalized)

$$RI(C_i, C_j) = \frac{|EC_{\{C_i,C_j\}}|}{\frac{|EC_{C_i}|+|EC_{C_j}|}{2}}$$

  - EC = edge-cut of cluster containing both C_i and C_j such that cluster is broken into C_i & C_j



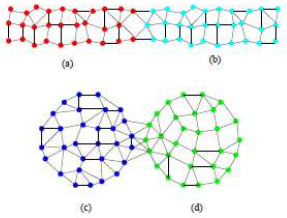
- Relative closeness
  - SEC = average weight of edges belonging to the min-cut bisector

$$RC(C_i, C_j) = \frac{\overline{S}_{EC_{\{C_i,C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\overline{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\overline{S}_{EC_{C_j}}}$$

Peter V. Henstock

Visual Idea

- Relative Interconnectivity
  - Prefers merging a&b over c&d (left)



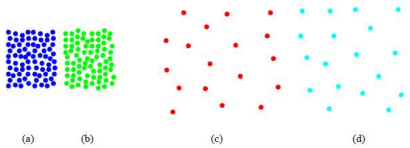
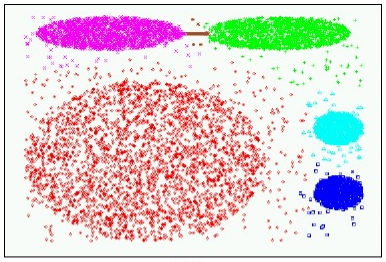


Figure 2: Example of clusters for merging choices.

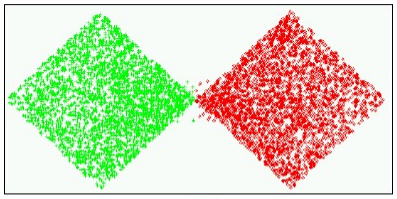
- Relative Closeness
  - Prefers merging c&d over a&b (right)

- Only merge based on thresholding these two criteria

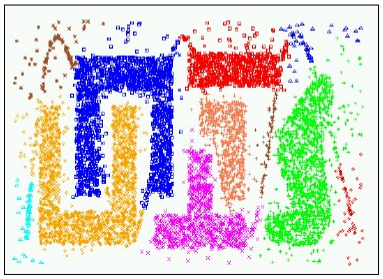
How does CHAMELEON work?



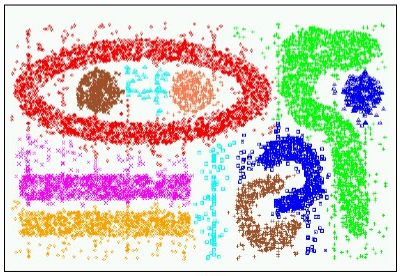
DS1



DS2



DS3



DS4

Subspace Clustering

Subspace

- Requirements of Subspace
  - If a, b in subspace W then a+b in W
  - If a in subspace W then ka in W for constant k
  - Note that 0 must be in subspace

Subspace Clustering

- Feature sets are often high dimensional
  - Underlying data may be lower dimensional
- Clustering of video of moving cars
  - Might need multiple subspaces for cars
  - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.225.2898&rep=rep1&type=pdf

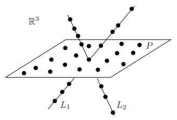


Fig. 1: A set of sample points in $\mathbb{R}^3$ drawn from a union of three subspaces: two lines and a plane.

Goals of Subspace Clustering

- Want to find:
  - #subspaces
  - Dimensions
  - Subspace bases
  - Members
- Why not just do PCA?
  - PCA assumes single subspace
  - Dimensions are selected
  - Bases are eigenvectors
  - Members are x - mean

CLIQUE (Aggrawal, et al. 1998)

- Grid-based: summarizes data by cell grid

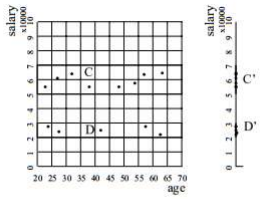


Figure 2: Identification of clusters in subspaces (projections) of the original data space.

- Density-based subspace method
  - Connects neighboring grid cells in subspace if their density > threshold

CLIQUE Algorithm

- Find 1D dense regions
- Extend to 2D using APRIORI algorithm
  - Finds minimum set of descriptors
  - Repeat for higher dimensionality

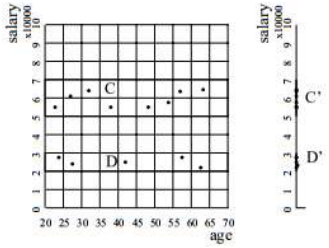


Figure 2: Identification of clusters in subspaces (projections) of the original data space.

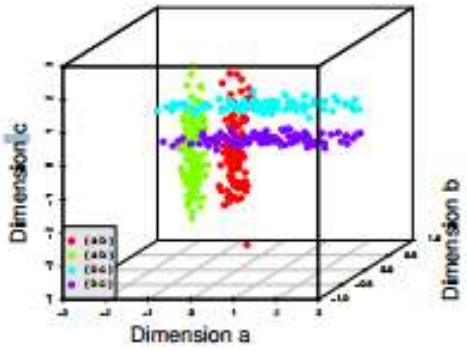
CLIQUE

- Good
  - Finds subspaces at high dimensionality
  - Reasonably fast: O(N) and dimensions
- Bad & Ugly
  - Quantization is always an issue in cells
  - Different results based on how divide
  - If large range in a dimension
    - Lots of bins that are empty?
    - Very coarse bins that are less useful
    - Differentially sized bins are more difficult

Subspace Clustering

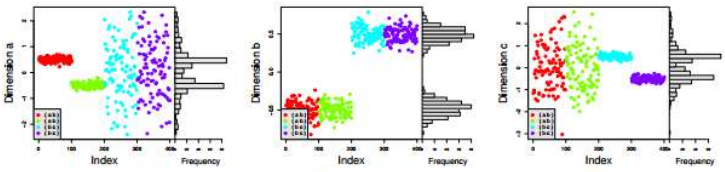
- Distance-based clustering
  - Using all the dimensions
  - Great features but diluted with noise
- What if we use “useful” dimensions only
  - Much larger search space
  - Potentially better insights

Motivation



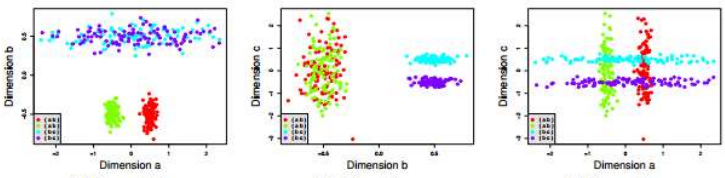
- Parsons 2004: Subspace Clustering
- Why we need to consider this area

Projections by axis



(a) Dimension a (b) Dimension b (c) Dimension c

Figure 3: Sample data plotted in one dimension, with histogram. While some clustering can be seen, points from multiple clusters are grouped together in each of the three dimensions.

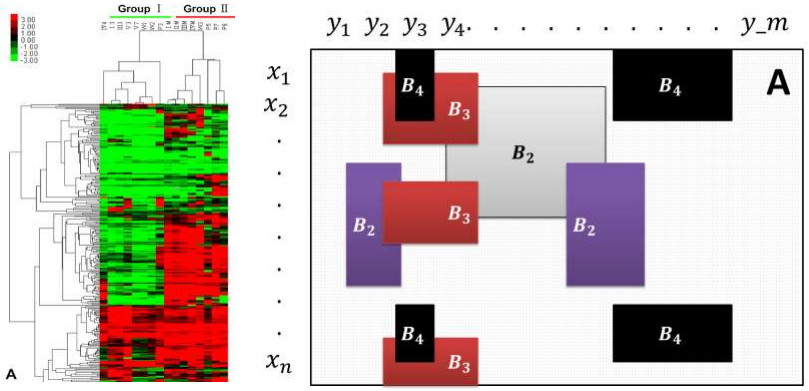


(a) Dims a & b (b) Dims b & c (c) Dims a & c

Figure 4: Sample data plotted in each set of two dimensions. In both (a) and (b) we can see that two clusters are properly separated, but the remaining two are mixed together. In (c) the four clusters are more visible, but still overlap each other are are impossible to completely separate.

Biclustering

- Left = global clustering with all rows and all clustering
- Right = biclustering finding subdimensional groups of rows and columns simultaneously

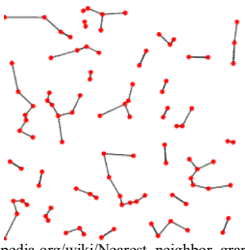


- http://www.biomedcentral.com/1471-2164/11/173/figure/F3
- http://www.abonyilab.com/biclustering

Spectral Clustering

Spectral Clustering

- Works for any shaped clusters
- Typically need to run it once
- Based on networks

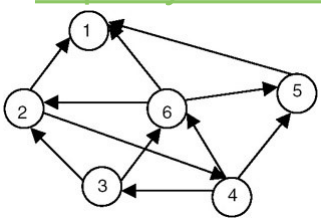


https://en.wikipedia.org/wiki/Nearest_neighbor_graph

- How do you turn data into a network?
- Could create a KNN Graph
  - Each vertex goes to K nearest neighbors
  - Fast algorithms exist
    - "Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures, Dong et al. 2011

Graph to Matrix

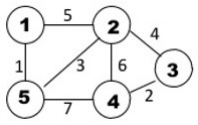
- Graphs can be represented as matrix
- Matrix[row_i,col_j] = edge weight i to j
  - Symmetric for undirected graphs
- Example from http://flylib.com/books/en/2.264.1.152/1/



	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	0	1	0	0
3	0	1	0	0	0	1
4	0	0	1	0	1	1
5	1	0	0	0	0	0
6	1	1	0	0	1	0

Adjacency, Degree & Laplacian

- Adjacency matrix (symmetric) contains undirected distance between all nodes



0	5	0	0	1
5	0	4	6	3
0	4	0	2	0
0	6	2	0	7
1	3	0	7	0

https://www.researchgate.net/post/What_is_the_effect_of_manipulating_the_Adjacency_Matrix_of_a_Network

- Degree = diagonal matrix: sum per row
  - Diagonal{6, 18, 6, 15, 11}
- Laplacian = D – A: What do you notice?

Laplacian				
6	-5	0	0	-1
-5	18	-4	-6	-3
0	-4	6	-2	0
0	-6	-2	15	-7
-1	-3	0	-7	11

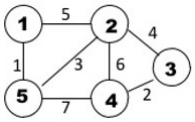
Degree				
6	0	0	0	0
0	18	0	0	0
0	0	6	0	0
0	0	0	15	0
0	0	0	0	11

Adjacency				
0	5	0	0	1
5	0	4	6	3
0	4	0	2	0
0	6	2	0	7
1	3	0	7	0

Eigenvalue & Eigenvector

- Common matrix process on A is to compute eigenvalue (λ) and eigenvector (ν) based on the equation: $A\nu = \lambda\nu$
- Sort the eigenvalues increasing order
- Count(λ=0) = #connected components
- If graph is connected and $\lambda_2 > 0$
  - Then λ_2 "algebraic connectivity
  - Higher λ_2 → more connected
  - Partition graph by $\nu_2 > 0$ vs. $\nu_2 < 0$

Spectral Clustering



Laplacian

6	-5	0	0	-1
-5	18	-4	-6	-3
0	-4	6	-2	0
0	-6	-2	15	-7
-1	-3	0	-7	11

Degree

6	0	0	0	0
0	18	0	0	0
0	0	6	0	0
0	0	0	15	0
0	0	0	0	11

Adjacency

0	5	0	0	1
5	0	4	6	3
0	4	0	2	0
0	6	2	0	7
1	3	0	7	0

v =

-0.4472	0.7507	0.4125	0.1407	-0.2157
-0.4472	0.0312	0.0163	-0.4890	0.7481
-0.4472	-0.6477	0.5618	0.2345	-0.0993
-0.4472	-0.1264	-0.3877	-0.5260	-0.5975
-0.4472	-0.0078	-0.6030	0.6398	0.1643

d =

-0.0000	0	0	0	0
0	5.8025	0	0	0
0	0	7.2641	0	0
0	0	0	18.8280	0
0	0	0	0	24.1054

1 component

$\lambda_2 = 5.8025 > 0$

$\nu_2 =$ [0.75, 0.03 -0.65 -0.13, -0.01]^T

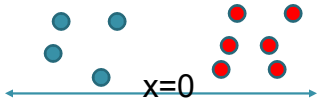
Best cut is 1-2 vs. 3-4-5

Why the magical λ_2?

- For a symmetric matrix M
  - $\lambda_2 = \min_x \frac{x^T M x}{x^T x}$ is solution to this
- Numerator for Laplacian L
  - $x^T L x = \sum_{i,j=1}^{n} L_{ij} x_i x_j = \sum_{i,j=1}^{n} (D_{ij} - A_{ij}) x_i x_j$
  - $= \sum_{i=1}^{n} D_{ij} x_i x_j - \sum_{(i,j) \in edges} 2 x_i x_j$
  - $= \sum_{(i,j) \in edges} (x_i^2 + x_j^2 - 2 x_i x_j)$
  - $= \sum_{(i,j) \in edges} (x_i - x_j)^2$
- We are minimizing distances (weights) in using clustering scheme based on λ_2
- Rayleigh Theorem: λ_2 is solution

Why the zero threshold?

- Eigenvector of unit length
  - So $\Sigma x_i^2 = 1$
- First eigenvector is 1s (not explained)
- Eigenvectors are orthogonal
  - So $\Sigma x_i \bullet 1 = 0$ or $\Sigma x_i = 0$
- If we are splitting nodes into 2 groups, then we want some in each group
- Since sum is 0, we use 0 as threshold



What about more partitions?

Recursive splitting approach

- Perform Laplacian cut recursively
- Not particularly efficient or stable
- Once a standard approach for VLSI
- "New spectral methods for ratio cut partitioning and clustering" Hagen & Kahng 1992

- Multiple Eigenvectors

What about more partitions?

- Multiple eigenvalue method
  - Normalized Cuts & Image Segmentation”
  - Shi & Malik 2000 IEEE PAMI

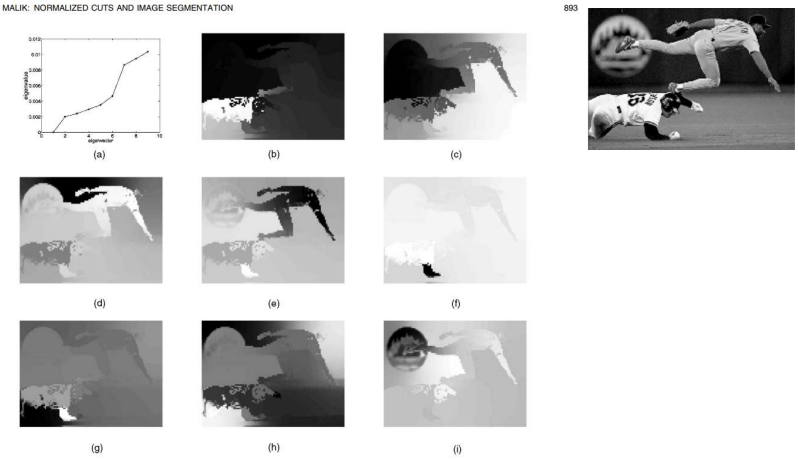


Fig. 3. Subplot (a) plots the smallest eigenvectors of the generalized eigenvalue system (11). Subplots (b)-(i) show the eigenvectors corresponding the second smallest to the ninth smallest eigenvalues of the system. The eigenvectors are reshaped to be the size of the image.

What about more partitions?

- Need to normalize Laplacian
  - $L_{norm} = D^{-1/2}LD^{-1/2}$
  - Compute eigenvalues/vectors on Lnorm
- This part is different:
  - $U = [\nu_1, \nu_2, \ldots \nu_n]$ eigenvectors
  - Take first k for k-clusters
  - Normalize each row to 1.0: $Y_{ij} = X_{ij}/(\Sigma X_{ij}^2)^{1/2}$
  - Perform k-means clustering on rows
- Essentially a dimensionality reduction
- Ng-Jordan-Weiss Method 2001

Results from Paper

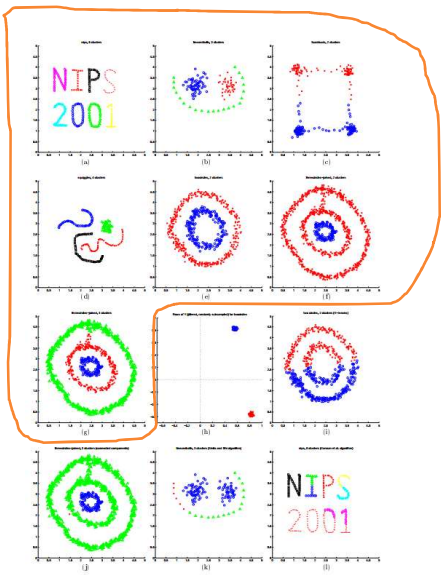


Figure 1: Clustering examples, with clusters indicated by different symbols (and colors, where available). (a-g) Results from our algorithm, where the only parameter varied across runs was k. (h) Rows of Y (jittered, subsampled) for `twocircles` dataset. (i) K-means. (j) A "connected components" algorithm. (k) Meila and Shi algorithm. (l) Kannan et al. Spectral Algorithm I. (See text.)

Practical Clustering

Official Steps of Clustering

On clustering validation techniques
Halkidi et al. J. Intelligent Info Sys.
Oct 2001

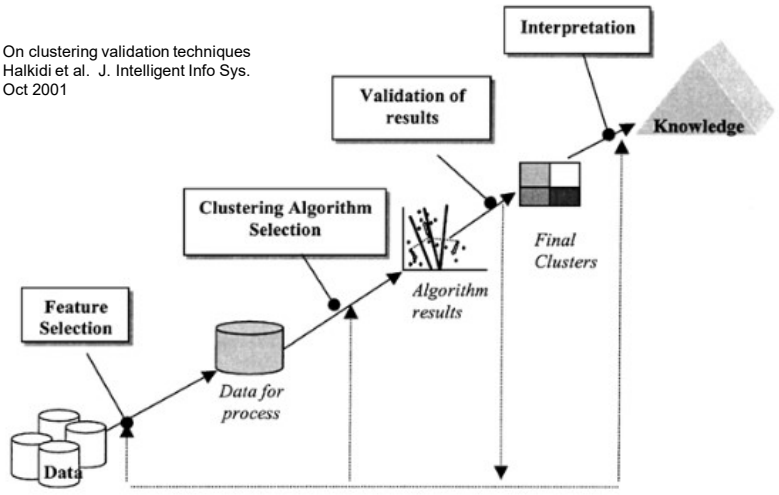


Figure 1. Steps of clustering process.

Motivation

6	94.3	61.2	67.4	82.3	54.0	59.4	72.2	73.9	64.1	26.1	19.1	91
4	92.6	98.4	88.4	92.1	70.1	40.9	18.1	79.6	46.3	5.56	-0.17	30
7	-1.85	24.2	17.2	15.0	29.2	15.0	23.7	-4.29	7.61	17.0	9.61	19
3	64.0	41.5	19.7	-7.57	49.3	8.38	21.7	52.9	43.1	0.77	9.88	41
8	76.0	12.1	23.0	21.7	72.0	25.5	64.1	87.8	2.98	12.6	6.36	18
2	98.0	49.8	53.3	84.9	58.7	96.5	53.9	75.7	95.2	89.0	81.5	85
7	97.9	41.1	37.1	39.9	45.5	69.8	61.8	94.8	48.6	32.3	40.9	72
1	95.5	19.9	28.6	81.5	74.3	88.6	46.9	59.5	31.9	20.7	26.3	58
4	14.1	24.7	20.5	16.2	19.9	0.93	27.3	5.67	19.5	25.2	5.0	19
2	11.0	7.0	15.1	55.3	29.3	21.8	10.4	6.02	18.9	16.4	1.87	25
5	11.1	14.3	-19.6	14.3	30.3	17.1	14.7	-8.57	1.37	10.7	9.86	21
	55.6	38.3	20.6	16.4	38.1	9.18	42.7	35.6	22.6	25.1	14.1	22
7	78.9	29.4	24.4	15.8	55.4	7.38	37.4	85.1	10.6	31.5	10.4	26
4	45.0	44.4	28.6	18.5	35.7	31.7	32.8	46.1	20.9	21.7	19.3	22
1	63.0	15.8	11.0	3.6	24.1	16.0	18.7	28.1	11.5	10.2	9.75	12
8	42.2	28.7	15.7	32.7	40.6	10.3	25.5	6.54	18.9	37.0	13.0	19
9	-7.46	11.0	7.79	8.82	6.21	15.2	18.4	-0.5	8.89	6.42	1.95	11
1	35.9	59.7	32.7	24.1	97.6	20.1	50.1	91.2	56.0	28.3	9.2	54

Practical guide to clustering

- Standard tool for grouping data
- First step in many analyses
- Hypothesis generation step

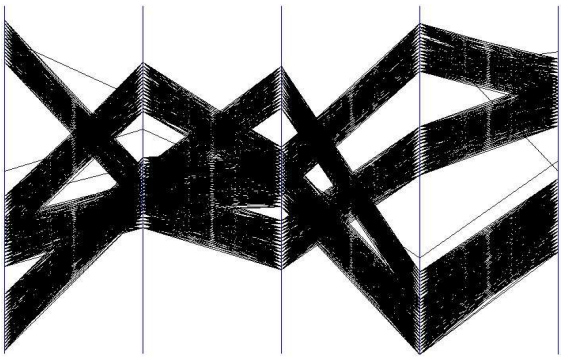
- Lots of techniques
- Which one do you use?

Goals of Clustering

- Identify a correct breakdown of the data
  - Validation from knowledge
  - Known data should be together
- Allow “business” folk to split/merge clusters to provide guidance
- Visualize
  - Dendrogram
  - Heat Map

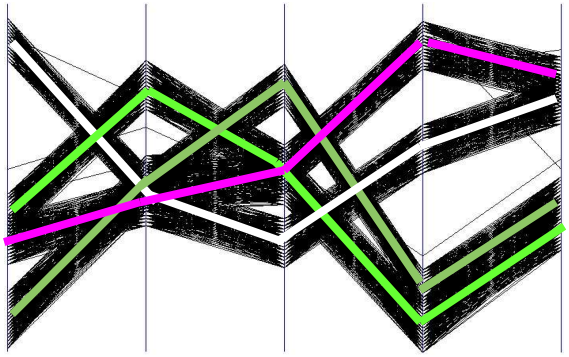
Visualization works

- How many clusters are there?
- Any outliers?
- Which series has highest target activity & lowest toxicity
- Which series are correlated to one with highest activity?



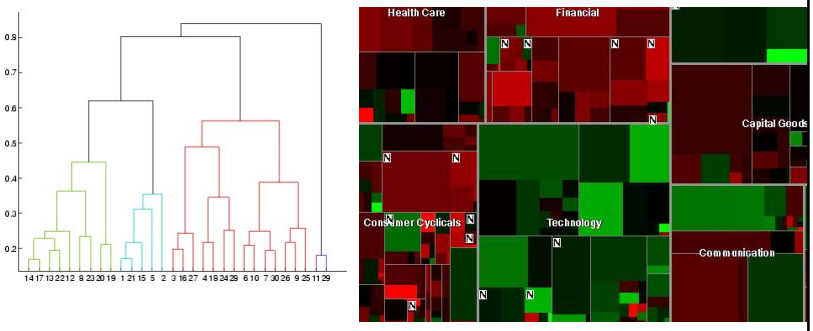
Why Parallel Coordinates?

- Any outliers?
- Which series has highest target activity & lowest toxicity
- Which series are correlated to one with highest activity?

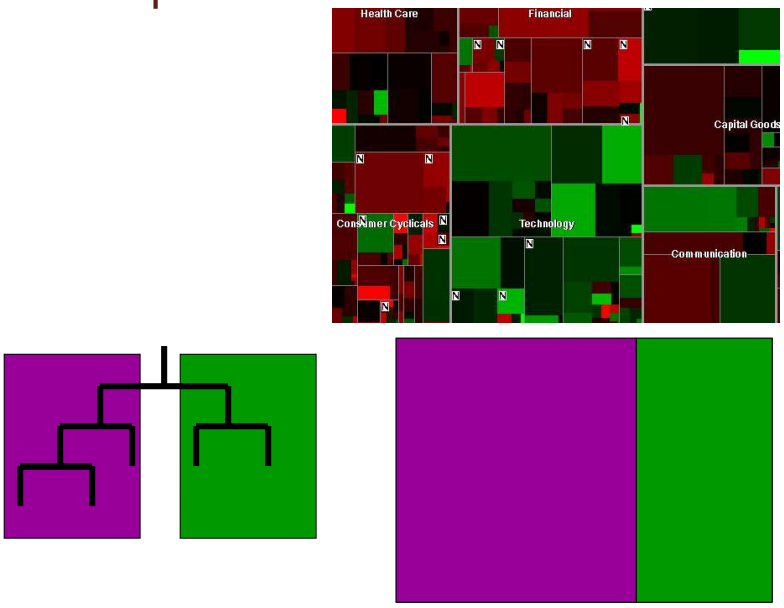


Hierarchical Clustering

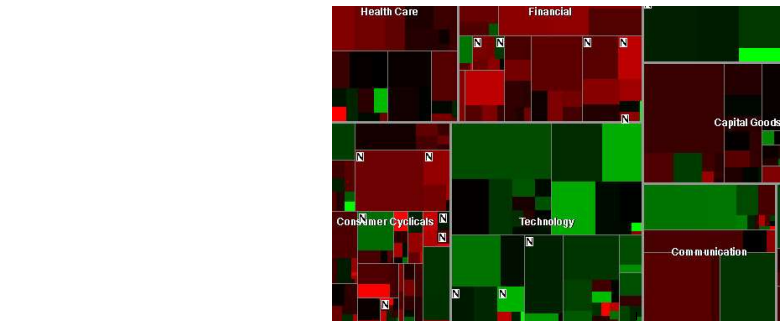
- Use hierarchical if the data is small enough
- Cluster first, decide where to cut later
- Interactive process

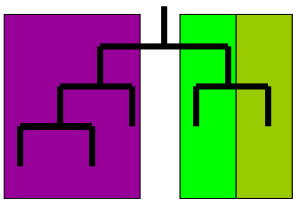


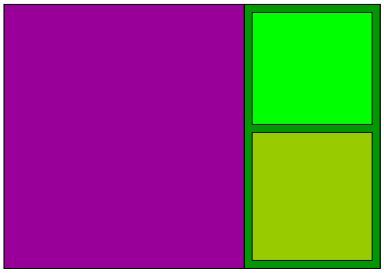
Tree Map Cluster View



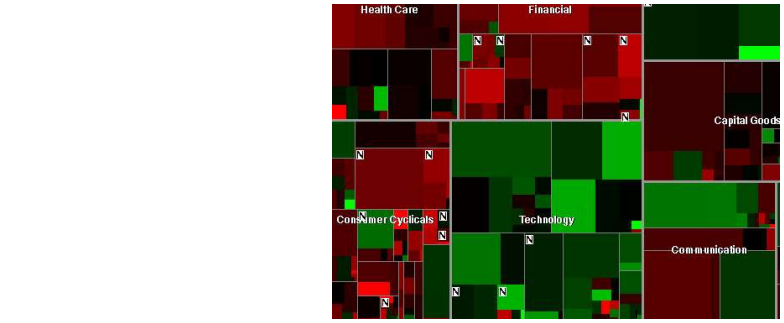
Tree Map Cluster View

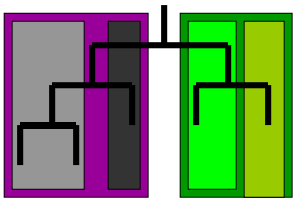


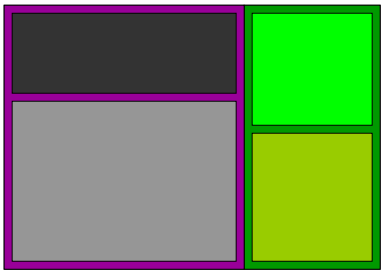




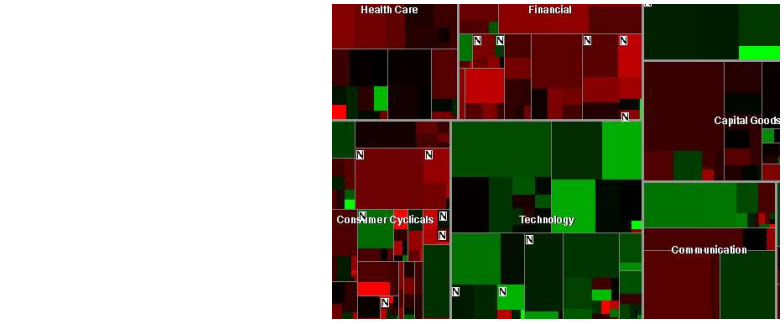
Tree Map Cluster View

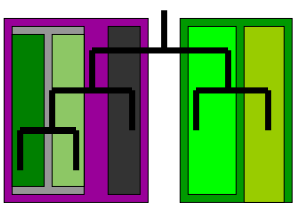


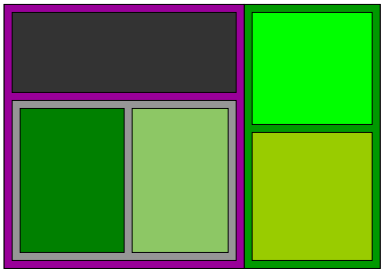




Tree Map Cluster View

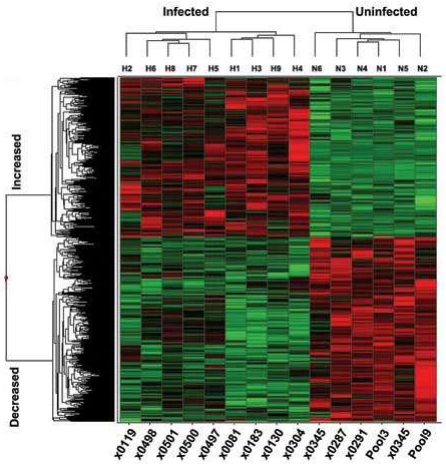






Recommended Strategies

- Use industry practices where possible
- Does green mean up or down?

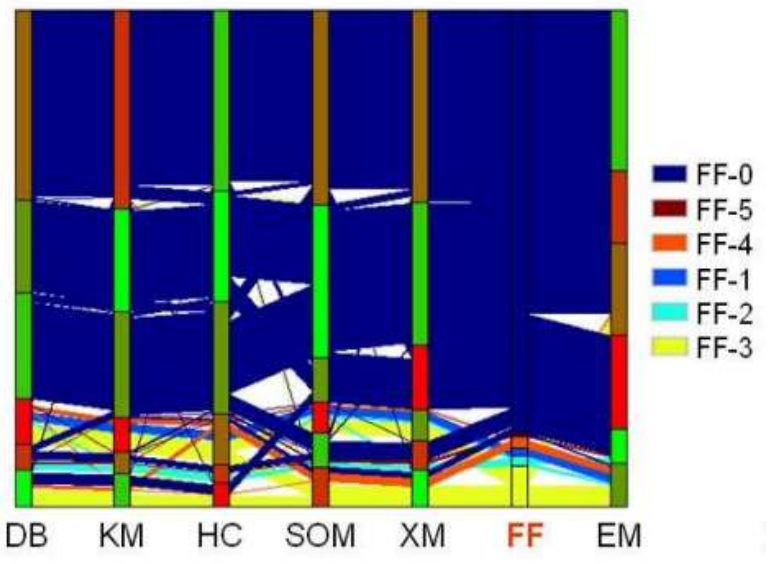


- http://compbio.pbworks.com/w/page/16252903/Microarray%20Clustering%20Methods%20and%20Gene%20Ontology

For Repeatable Results

- Each method has its own nuances
- Run multiple clustering methods
- Run multiple parameter settings
- Look for patterns in the data as to how things cluster across the different groups
- Tools are not great for comparing results

Parallel Sets to Visualize

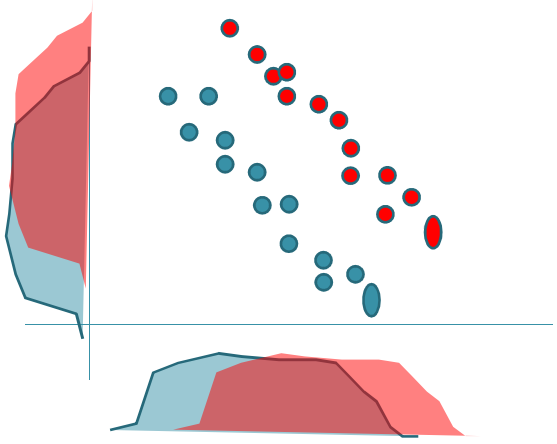


- Visually Comparing Multiple Partitions of Data with Applications to Clustering Zhou,..Grinstein

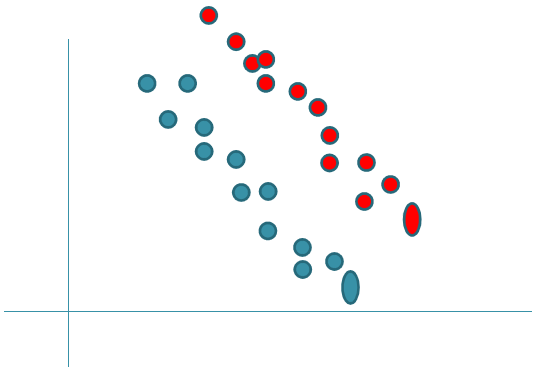
Linear Discrimination

Fisher Linear Discrimination

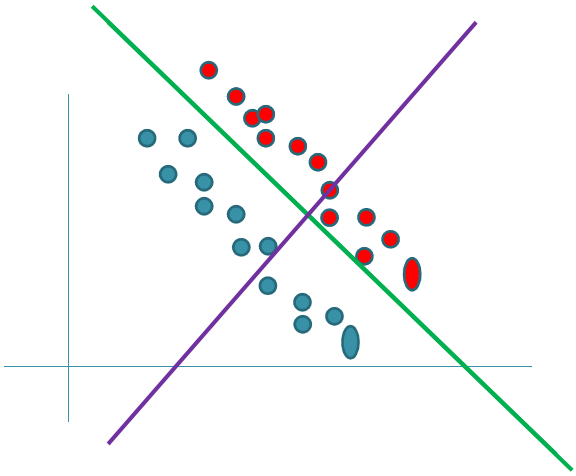
- Extension of PCA approach to facilitate classification



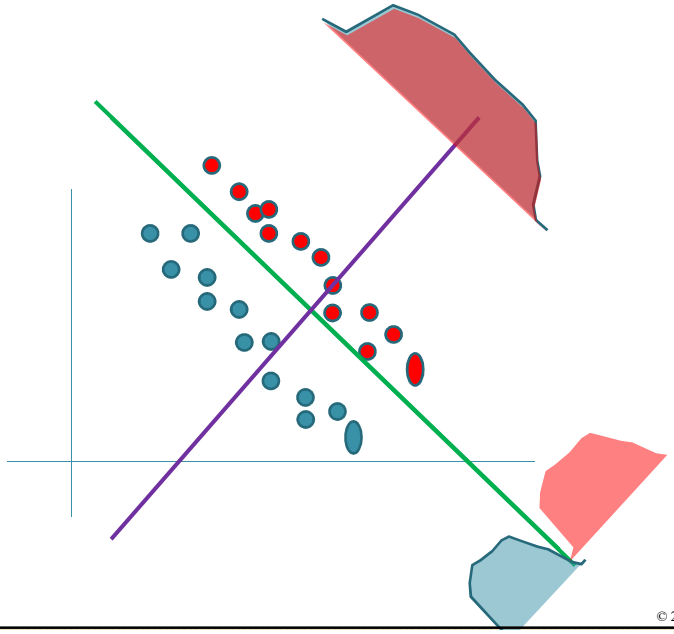
How could we project to separate?



How could we project to separate?



How could we project to separate?



Fischer's Linear Discrimination

- $w = S_w^{-1}(m_1 - m_2)$
- w = transformed vector
- m_1 = mean of one class
- m_2 = mean of other class
- $S_w = S_1 + S_2$
- $Si = \Sigma(x-m_i)^T (x-m_i)$ = "scatter matrix"
- If the data are normal with equal covariance matrices
- $w = \Sigma^{-1}(\mu_1 - \mu_2)$

Assessment of Classifiers

Cross-validation: Assessing predictions

- Divide the ***known*** data (projects) into 4 groups

Set1	Train	Train	Train	Test
Set2	Train	Train	Test	Train
Set3	Train	Test	Train	Train
Set4	Test	Train	Train	Train

- Train on 75% of the data; test on 25%
- Build 1 tree per set on combined 3 “Train” groups
- Assess “Test” results on corresponding tree in set
- Since it’s known data, assess the predicted outcomes from Test against known values

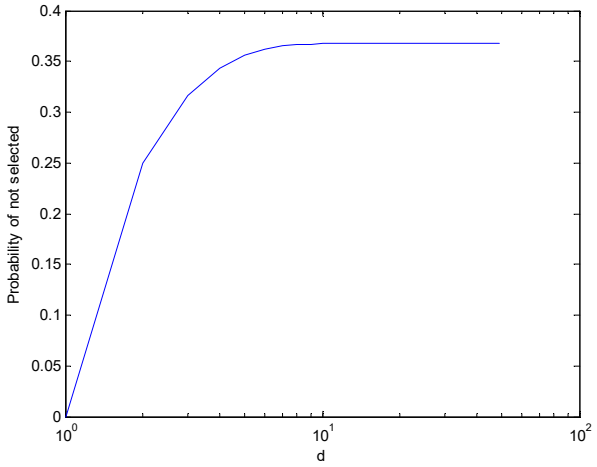
Bootstrap Method

- Cross-validation
  - Divide data into multiple training & test sets
  - Cross-validation tests every point once
  - Sample "test sets" without replacement
  - All non "test sets" are the training
- Bootstrap = alternative to cross-validation
  - Selects test sets *with* replacement
  - Fraction of data set for test set:
    - 1-0.632 = 0.368 or 36.8% of full data set

Why 0.632 – 0.368 split?

- Data set has d rows
- In a single sample
  - Row has 1/d chance of being selected
  - Row has 1–1/d chance of not being selected
- Sample it d times with replacement
- Over d samples →
  - P(not chosen) = $(1-1/d)^d$
- As d→large # →
  - P(not chosen) → 0.368 = 1/e

Why is it 0.632 : 0.368?



- As d increases, it plateaus at 0.368
- 1-0.368 = 0.632

Bootstrap Method

- Sample with replacement
- 0.632 bootstrap is a standard

- Repeat the sampling process k times
- Model accuracy Acc(M)
  - $Acc(M) = \frac{1}{k}\sum_{i=0}^{k} 0.632\ Acc(Mi)_{testset}$ +
    $+ \frac{1}{k}\sum_{i=0}^{k} 0.368\ Acc(Mi)_{trainset}$

How to evaluate?

- Bootstrap is optimistic in general
- Better to use a 10-fold cross-validation provided you have enough data
- However, bootstrap is widely used for small sets of samples as the alternative is not very good

High Level Evaluation Approach

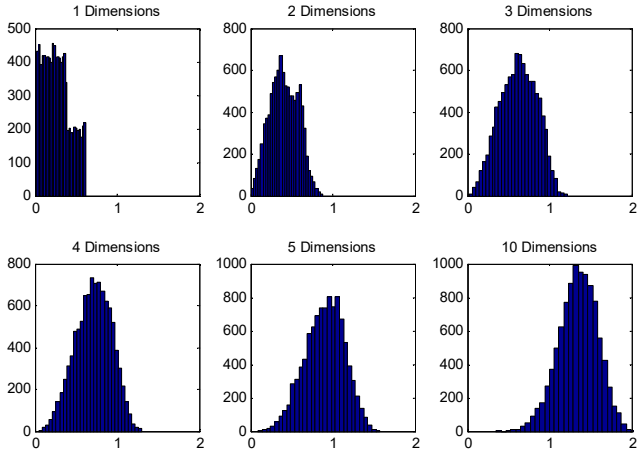
- Holdout: train on 2/3, test on 1/3
  - Bootstrap = repeated sampling within holdout then average results
- Cross-Validation (Jackknife)
  - K-fold
  - Leave-one-out
  - Stratified cross validation
    - Recognize people from front, side, back
    - Most pictures are from front so biased

K-NN

- Good
- Bad
- Ugly

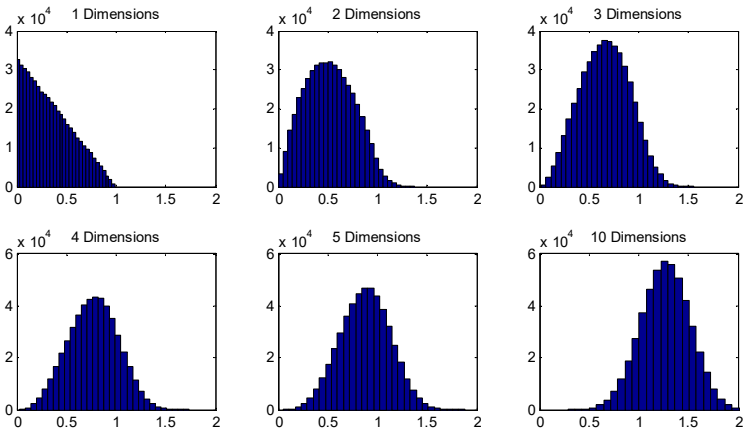
Curse of Dimensionality

- Uniform sampling in $[0,1]^{dim}$
- Distribution of Euclidian nearest neighbor



Curse of Dimensionality

- Uniform sampling in $[0,1]^{dim}$
- Distribution of all pair Euclidian distances



K-NN

- Good
- Bad
- Ugly

kNN

- Good
  - Fast to train
  - Easy to explain
- Bad
  - Slow to predict: lot of baseline computation
  - Trial and error to determine a good K
- Ugly
  - Not clear what the topology really is
  - All-features distance typically

Decision Trees

- Good
- Bad
- Ugly

Good and Bad of Decision Trees

- Good:
  - No assumptions about distributions of data
  - Fairly robust compared to statistical approaches
- Bad:
  - How many rules needed?
  - Hard limiter at each node
- Ugly:
  - 9 out of 10 dentists recommend flossing
  - Lupus is a collection of 17 or 24 symptoms
    - No good way to represent this in decision tree