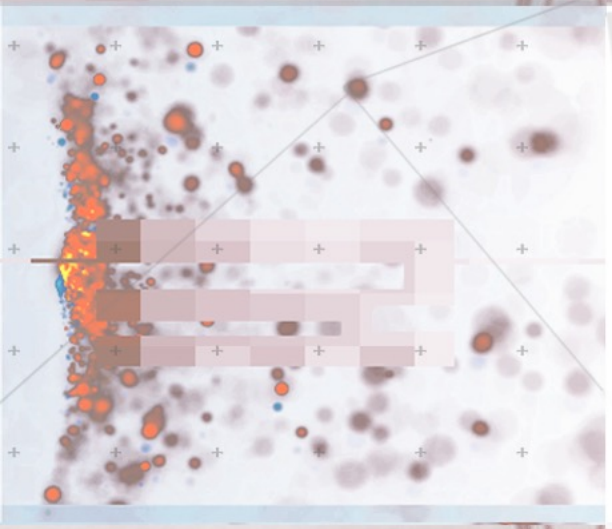
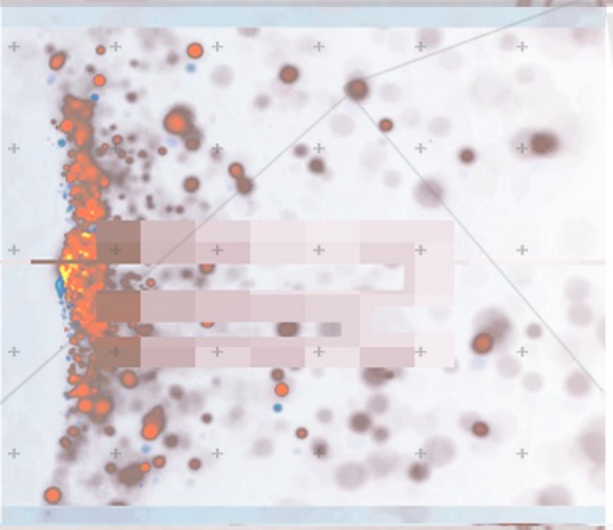


Evaluation of Clustering

Evaluation of Clustering

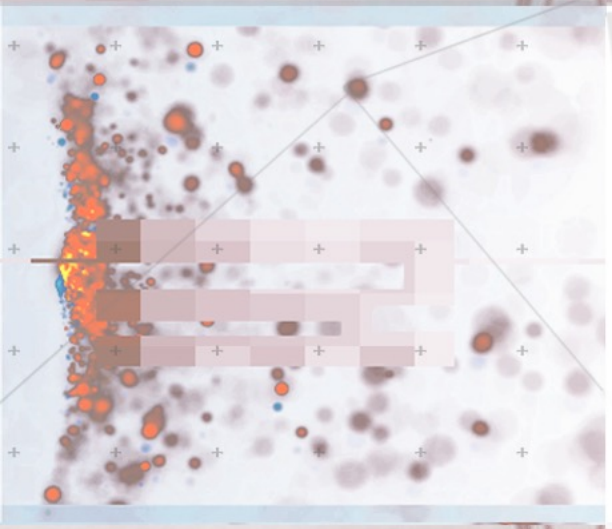
- Evaluation of Clustering: Basic Concepts
- Clustering Tendency
- Determining the Number of Clusters
- Measuring Clustering Quality: Extrinsic Methods
  - Extrinsic vs. intrinsic methods
  - I: Matching-Based Methods
  - II: Information Theory-Based Methods
  - III: Pairwise Comparison-Based Methods
- Measuring Clustering Quality: Intrinsic Methods



Session 1: Evaluation of Clustering: Basic Concepts

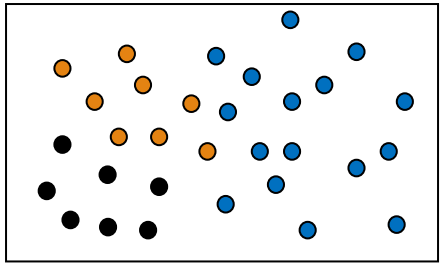
Evaluation of Clustering: Basic Concepts

- Evaluation of clustering
  - Assess the feasibility of clustering analysis on a data set
  - Evaluate the quality of the results generated by a clustering method
- Major issues on clustering assessment and validation
  - **Clustering tendency**
    - Assess the suitability of clustering: whether the data has any inherent grouping structure
  - **Determining the Number of Clusters**
    - Determine for a dataset the right number of clusters that may lead to a good quality clustering
  - **Clustering quality evaluation**
    - Evaluate the quality of the clustering results



Session 2: Clustering Tendency

Clustering Tendency: Whether the Data Contains Inherent Grouping Structure

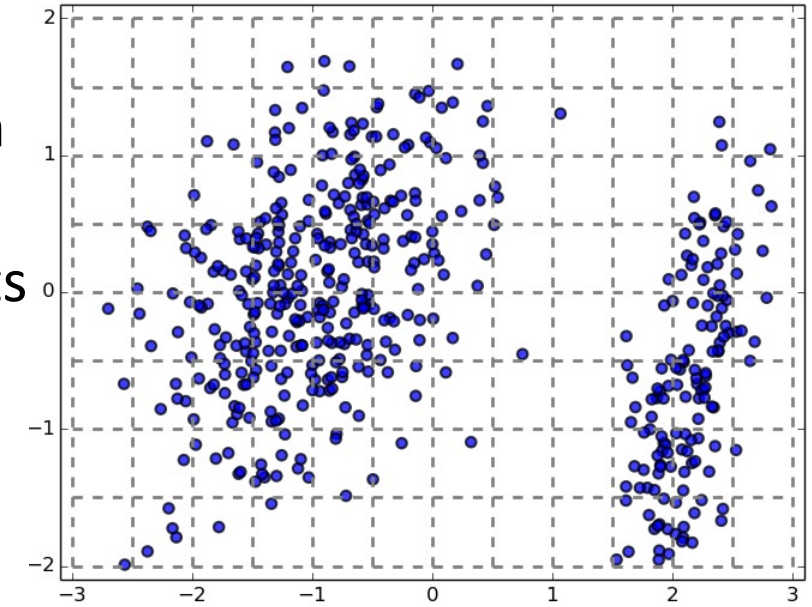


- Assess the **suitability of clustering**
  - Whether the data has any *"inherent grouping structure"* — non-random structure that may lead to meaningful clusters
- Determine ***clustering tendency*** or ***clusterability***
  - A hard task because there are so many different definitions of clusters
    - Different definitions: Partitioning, hierarchical, density-based and graph-based
  - Even fixing a type, still hard to define an appropriate null model for a data set
- There are some **clusterability assessment methods**, such as
  - **Spatial histogram**: Contrast the histogram of the data with that generated from random samples
  - **Distance distribution**: Compare the pairwise point distance from the data with those from the randomly generated samples
  - **Hopkins Statistic**: A sparse sampling test for spatial randomness

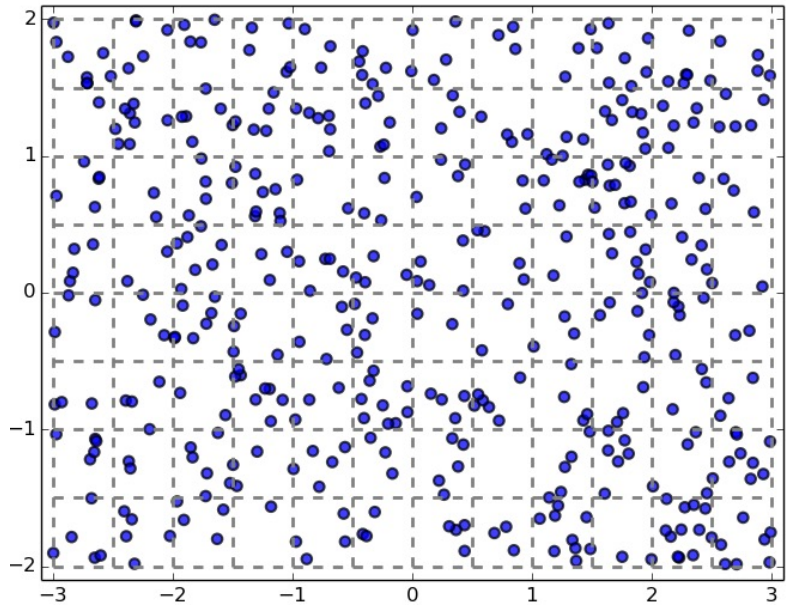
To be covered here

Testing Clustering Tendency: A Spatial Histogram Approach

- **Spatial Histogram Approach:** Contrast the *d*-dimensional histogram of the input dataset ***D*** with the histogram generated from random samples
  - Dataset D is clusterable if the distributions of two histograms are rather different
- Method outline
  - Divide each dimension into equi-width bins, count how many points lie in each cell, and obtain the *empirical joint probability mass function* (EPMF)

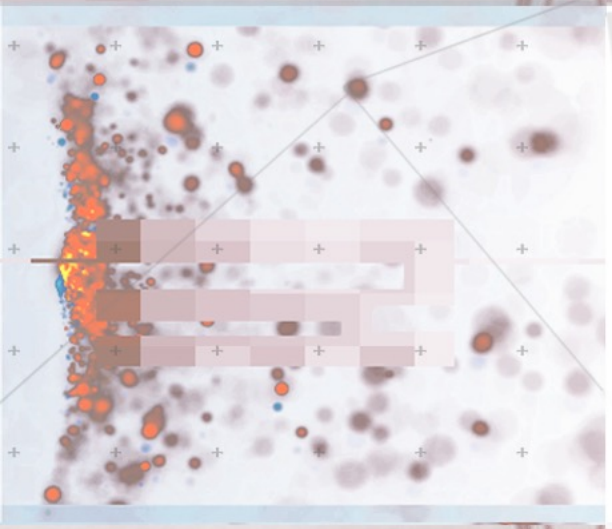


(a) Input dataset



(b) Data generated from random samples

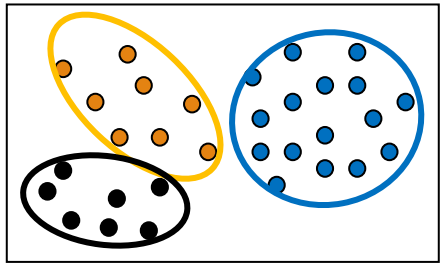
  - Do the same for the randomly sampled data
  - Compute how much they differ using the *Kullback-Leibler* (*KL*) *divergence* value



Session 3: Determining the Number of Clusters

Determining the Number of Clusters

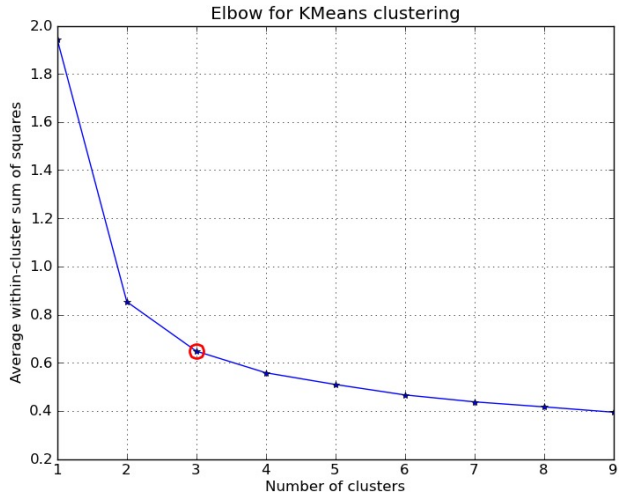
- The appropriate number of clusters controls the proper granularity of cluster analysis
  - Finding a good balance between compressibility and accuracy in cluster analysis



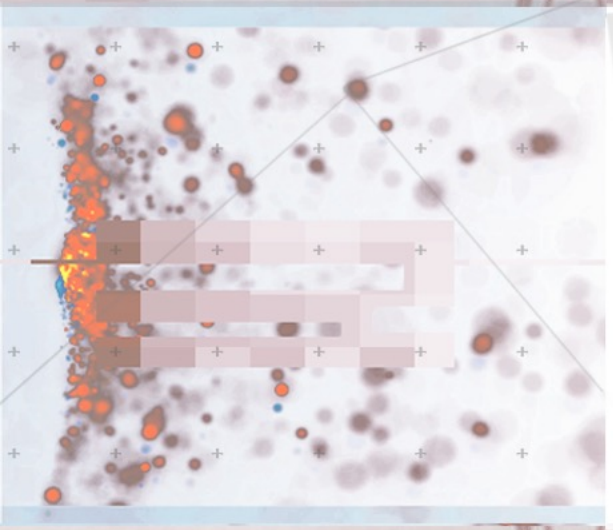
- Two undesirable extremes
  - The whole data set is one cluster: No value of clustering
  - Treating each point as a cluster: No data summarization
- The right number of clusters often depends on the distribution's shape and scale in the data set, as well as the clustering resolution required by the user
- **Methods for determining the number of clusters**
  - **Empirical method**
    - # of clusters: $k \approx \sqrt{n/2}$ for a dataset of n points (e.g., n = 200, k = 10)
      - Each cluster is expected to have about $\sqrt{2n}$ points

Finding K, the Number of Clusters: Additional Methods

- **Elbow method**: Use the turning point in the curve of the sum of within cluster variance with respect to the # of clusters
  - Increasing the # of clusters can help reduce the sum of within-cluster variance of each cluster
  - But splitting a cohesive cluster gives only a small reduction



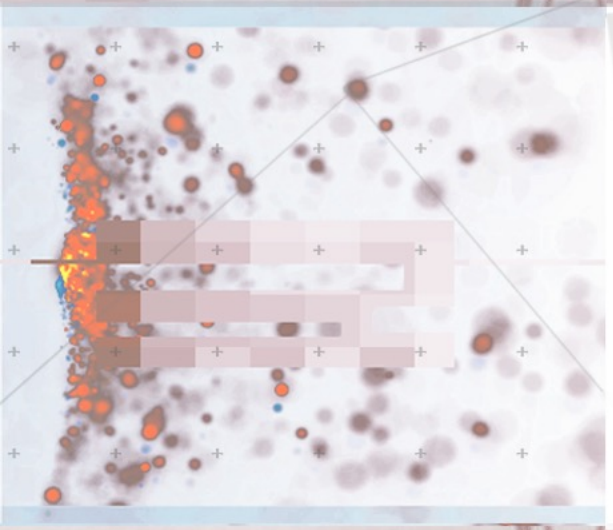
- **Cross validation method**
  - Divide a given data set into *m* parts, and use *m* – 1 parts to obtain a clustering model
  - Use the remaining part to test the quality of the clustering
    - For example, for each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and their closest centroids to measure how well the model fits the test set
  - For any $k > 0$, repeat it *m* times, compare the overall quality measure w.r.t. different *k's*, and find # of clusters that fits the data the best



Session 4: Clustering Evaluation: Measuring Clustering Quality

Measuring Clustering Quality

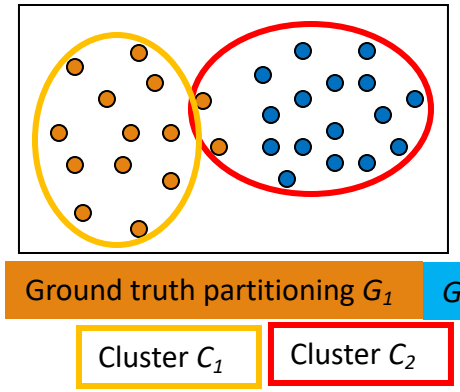
- **Clustering Evaluation**: Evaluating how good the clustering results are
  - No commonly recognized best suitable measure in practice
- **Extrinsic vs. intrinsic methods**: depending on whether *ground truth* is used
    - Ground truth: the ideal clustering built by using human experts
  - **Extrinsic**: Supervised, employ criteria not inherent to the dataset
    - Compare a clustering against prior or expert-specified knowledge (i.e., the ground truth) using certain clustering quality measure
  - **Intrinsic**: Unsupervised, criteria derived from data itself
    - Evaluate the goodness of a clustering by considering how well the clusters are separated and how compact the clusters are (e.g., silhouette coefficient)



Session 5: Clustering Evaluation: Extrinsic Methods

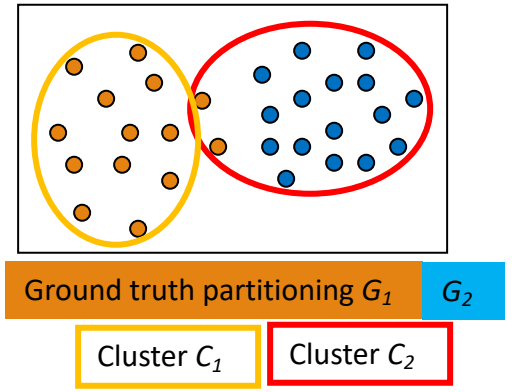
General Criteria for Measuring Clustering Quality with Extrinsic Methods

- Given the **ground truth** C_g, $Q(C, C_g)$ is the **quality measure** for a clustering C
- $Q(C, C_g)$ is good if it satisfies the following **four** essential criteria
  - **Cluster homogeneity**
    - The purer, the better
  - **Cluster completeness**
    - Assign objects belonging to the same category in the ground truth to the same cluster
  - **Rag bag better than alien**
    - Putting a heterogeneous object into a pure cluster should be penalized more than putting it into a *rag bag* (i.e., "miscellaneous" or "other" category)
  - **Small cluster preservation**
    - Splitting a small category into pieces is more harmful than splitting a large category into pieces

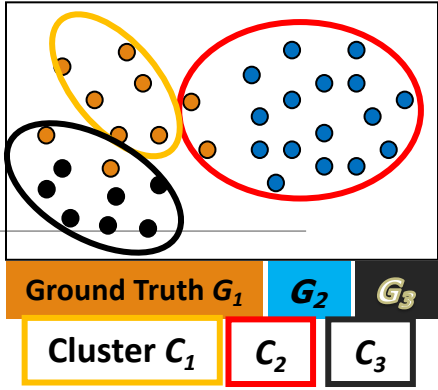


Commonly Used Extrinsic Methods

- **Matching-based methods**
  - Examine how well the clustering results match the ground truth in partitioning the objects in the data set
- **Information theory-based methods**
  - Compare the distribution of the clustering results and that of the ground truth
  - Information theory (e.g., entropy) used to quantify the comparison
  - Ex. Conditional entropy, normalized mutual information (NMI)
- **Pairwise comparison-based methods**
  - Treat each group in the ground truth as a class, and then check the pairwise consistency of the objects in the clustering results
  - Ex. Four possibilities: TP, FN, FP, TN; Jaccard coefficient



Matching-Based Methods



- The matching based methods compare clusters in the clustering results and the groups in the ground truth
- Suppose a clustering method partitions D = {o_1, …, o_n} into *m* clusters C = {C_1, …, C_m}. The ground truth G partitions D into *l* groups G = {G_1, …, G_l}.
- **Purity**: The extent that cluster C_i contains points only from one (ground truth) partition
  - Purity for cluster C_i : $\frac{|C_i \cap G_j|}{|C_i|}$ where G_j matching C_i maximizes $|C_i \cap G_j|$
  - Total purity of clustering *C*:

$$purity = \sum_{i=1}^{m} \frac{|C_i|}{n} \max_{j=1}^{l} \{\frac{|C_i \cap G_j|}{|C_i|}\} = \frac{1}{n} \sum_{i=1}^{m} \max_{j=1}^{l} \{|C_i \cap G_j|\}$$

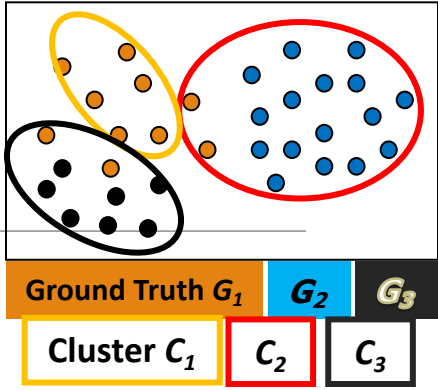
- Example: 11 objects

Object	a	b	c	d	e	f	g	h	i	j	k
Ground truth $\mathcal{G}$	G_1	G_1	G_1	G_1	G_1	G_1	G_2	G_2	G_2	G_2	G_3
Clustering $\mathcal{C}_1$	C_1	C_1	C_1	C_1	C_2	C_2	C_3	C_3	C_3	C_3	C_4
Clustering $\mathcal{C}_2$	C_1	C_1	C_2	C_2	C_2	C_3	C_1	C_2	C_2	C_1	C_3

- Other methods:
  - maximum matching; F-measure

Purity for clustering $\boldsymbol{C_1}$ = 1/11 (4 + 2 + 4 + 1) = 11/11 = 1;
Purity for clustering $\boldsymbol{C_2}$ = 1/11 (2 + 3 + 1) = 6/11

Information Theory-Based Methods (I) Conditional Entropy



- A clustering can be regarded as a compressed representation of a given set of objects
- The better the clustering results approach the ground-truth, the less amount of information is needed. This leads to the use of *conditional entropy.*
- **Entropy of clustering** *C:*

$$H(\mathcal{C}) = -\sum_{i=1}^{m} \frac{|C_i|}{n} \log \frac{|C_i|}{n}$$

Entropy of ground truth *G*:

$$H(\mathcal{G}) = -\sum_{i=1}^{l} \frac{|G_i|}{n} \log \frac{|G_i|}{n}$$

- **Conditional entropy of G given cluster** C_i**:**

$$H(\mathcal{G}|C_i) = -\sum_{j=1}^{l} \frac{|C_i \cap G_j|}{|C_i|} \log \frac{|C_i \cap G_j|}{|C_i|}$$

Conditional entropy of G given clustering *C:*

$$H(\mathcal{G}|\mathcal{C}) = \sum_{i=1}^{m} \frac{|C_i|}{n} H(\mathcal{G}|C_i) = -\sum_{i=1}^{m} \sum_{j=1}^{l} \frac{|C_i \cap G_j|}{n} \log \frac{|C_i \cap G_j|}{|C_i|}$$

- For the last example:

$$H(\mathcal{G}|\mathcal{C}_1) = -(\frac{4}{11}\log\frac{4}{4} + \frac{2}{11}\log\frac{2}{2} + \frac{4}{11}\log\frac{4}{4} + \frac{1}{11}\log\frac{1}{1}) = 0$$

$$H(\mathcal{G}|\mathcal{C}_2) = -(\frac{2}{11}\log\frac{2}{4} + \frac{2}{11}\log\frac{2}{4} + \frac{3}{11}\log\frac{3}{5} + \frac{2}{11}\log\frac{2}{5} + \frac{1}{11}\log\frac{1}{2} + \frac{1}{11}\log\frac{1}{2})$$
$$= 0.297$$

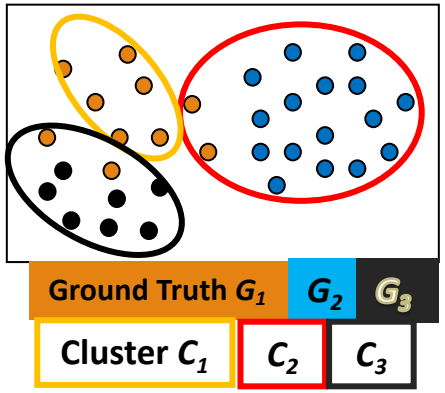
Note: conditional entropy cannot detect the issue that $\boldsymbol{C_1}$ splits the objects in *G* into two clusters

Information Theory-Based Methods (II) Normalized Mutual Information (NMI)

- **Mutual information *I*(*C*, *G*)**:
  - Quantifies the amount of shared info between the clustering *C* and ground-truth partitioning *G*

$$I(\mathcal{C},\mathcal{G}) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log\left(\frac{p_{ij}}{p_{C_i}\cdot p_{G_j}}\right)$$

  - Measures the dependency between the observed joint probability p_{ij} of *C* and *G*, and the expected joint probability $p_{Ci} \cdot p_{Gj}$ under the independence assumption
  - When *C* and *G* are independent, $p_{ij} = p_{Ci} \cdot p_{Gj}$, $I(C, G) = 0$. However, there is no upper bound on the mutual information



- **Normalized mutual information** (NMI)

$$NMI(\mathcal{C},\mathcal{G}) = \sqrt{\frac{I(\mathcal{C},\mathcal{G})}{H(\mathcal{C})}\cdot\frac{I(\mathcal{C},\mathcal{G})}{H(\mathcal{G})}} = \frac{I(\mathcal{C},\mathcal{G})}{\sqrt{H(\mathcal{C})\cdot H(\mathcal{G})}}$$

  - Value range of NMI: [0,1]. Value close to 1 indicates a good clustering

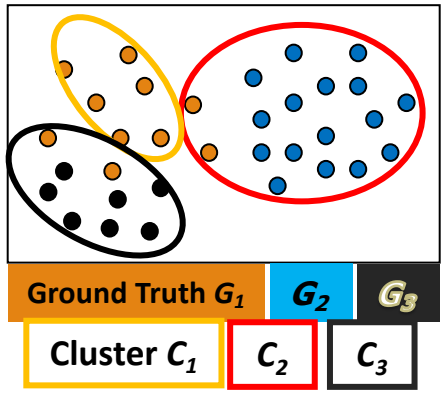
Pairwise Comparison-Based Methods: Jaccard Coefficient

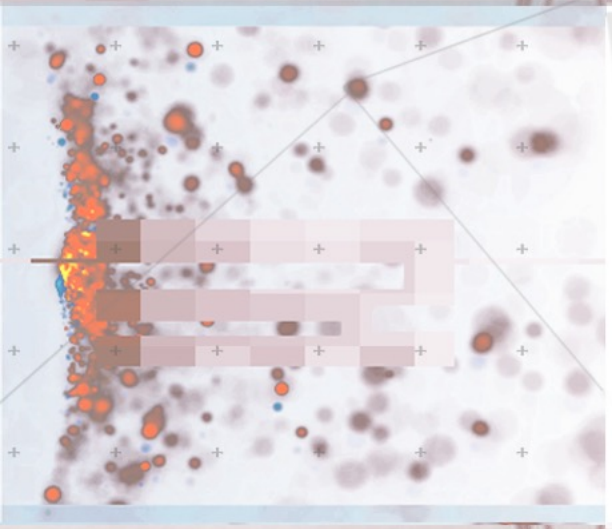
- Pairwise comparison: treat each group in the ground truth as a class
- For each pair of objects (o_i, o_j) in D, if they are assigned to the same cluster/group, the assignment is regarded as positive, o/w, negative.
  - Depending on assignments, we have four possible cases:

Note: Total # of pairs of points $N = \binom{n}{2}$

	$C(o_i) = C(o_j)$	$C(o_i) \neq C(o_j)$
$G(o_i) = G(o_j)$	true positive (TP)	false negative (FN)
$G(o_i) \neq G(o_j)$	false positive (FP)	true negative (TN)

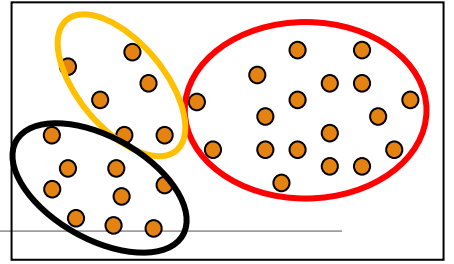
- **Jaccard coefficient:** Ignoring the true negatives (thus asymmetric)
  - *Jaccard* = *TP*/(*TP* + *FN* + *FP*) [i.e., denominator ignores *TN*]
    - *Jaccard* = 1 if perfect clustering
- Many other measures are based on the pairwise comparison statistics:
  - Rand statistic
  - Fowlkes-Mallows measure





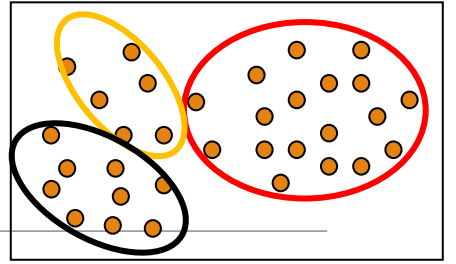
Session 6: Clustering Evaluation: Intrinsic Methods

Intrinsic Methods (I): Dunn Index



- Intrinsic methods (i.e., no ground truth) examine how compact clusters are and how well clusters are separated, based on similarity/distance measure between objects
- **Dunn Index**:
  - The compactness of clusters: the maximum distance between two points that belong to the same cluster: $\Delta = \max_{C(o_i)=C(o_j)}\{d(o_i, o_j)\}$
  - The degree of separation among different clusters: the minimum distance between two points that belong to different clusters: $\delta = \min_{C(o_i)\neq C(o_j)}\{d(o_i, o_j)\}$
  - The Dunn index is simply the ratio: $DI = \frac{\delta}{\Delta}$
    - The larger the ratio, the farther away the clusters are separated comparing to the compactness of the clusters
- Dunn index uses the extreme distances to measure the cluster compactness and inter-cluster separation and it can be affected by outliers

Intrinsic Methods (II): Silhouette Coefficient



- Suppose D is partitioned into *k* clusters: $C_1, \ldots, C_k$. For each object o in D, we calculate
- a(o): avg distance between *o* and all other objects in the cluster to which *o* belongs

$$a(\boldsymbol{o}) = \frac{\sum_{\boldsymbol{o}' \in C_i, \boldsymbol{o} \neq \boldsymbol{o}'} dist(\boldsymbol{o}, \boldsymbol{o}')}{|C_i| - 1}$$

a(o) reflects the compactness of the cluster to which *o* belongs

- b(o): minimum avg distance from *o* to all clusters to which *o* does not belong

$$b(\boldsymbol{o}) = \min_{C_j : 1 \leq j \leq k, j \neq i} \left\{ \frac{\sum_{\boldsymbol{o}' \in C_j} dist(\boldsymbol{o}, \boldsymbol{o}')}{|C_j|} \right\}$$

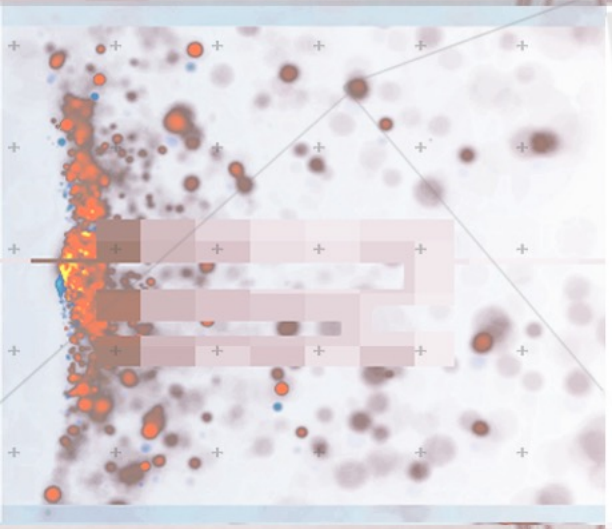
b(*o*) captures the degree to which *o* is separated from other clusters

- Silhouette Coefficient:

$$s(\boldsymbol{o}) = \frac{b(\boldsymbol{o}) - a(\boldsymbol{o})}{\max\{a(\boldsymbol{o}), b(\boldsymbol{o})\}}$$

value range (-1, 1)

  - When the value of *o* approaches 1, the cluster containing *o* is compact and *o* is far away from other clusters, which is the preferable case
  - When the value is negative (i.e., b(o) < a(o)), *o* is closer to the objects in another cluster than to the objects in the same cluster as *o*: a bad situation to be avoided



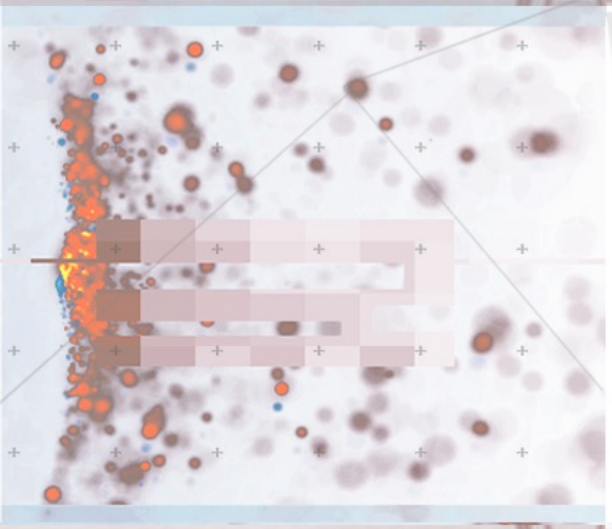
Summary

Summary: Evaluation of Clustering

- Evaluation of Clustering: Basic Concepts
- Clustering Tendency
- Determining the Number of Clusters
- Measuring Clustering Quality: Extrinsic Methods
  - Extrinsic vs. intrinsic methods
  - I: Matching-Based Methods
  - II: Information Theory-Based Methods
  - III: Pairwise Comparison-Based Methods
- Measuring Clustering Quality: Intrinsic Methods

Recommended Readings

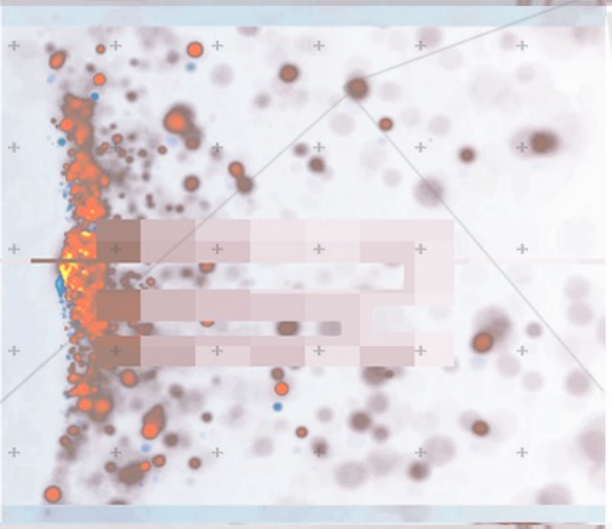
- J. Han, J. Pei, and H. Tong. Data Mining: Concepts and Techniques. Morgan Kaufmann, 4th ed. , 2022
- M. J. Zaki and W. Meira, Jr.. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press, 2014
- L. Hubert and P. Arabie. Comparing Partitions. *Journal of Classification*, 2:193–218, 1985
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988
- M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Info. Systems*, 17(2-3):107–145, 2001
- H. Xiong and Z. Li. Clustering Validation Measures. in (Chapter 23) C. Aggarwal and C. K. Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press, 2014



Clustering High-Dimensional Data

Clustering High-Dimensional Data

- Challenges of Clustering High-Dimensional Data
- Methods for Clustering High-Dimensional Data
- Subspace Clustering Methods
  - Subspace Clustering I: Subspace Search Methods
  - Subspace Clustering II: Correlation-Based Methods
  - Subspace Clustering III: Bi-Clustering Methods
    - δ-Bi-Clustering
    - δ-pClustering
- Dimensionality Reduction Methods



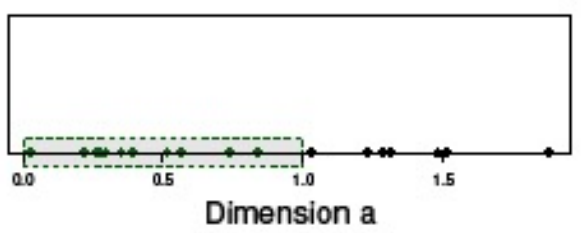
Session 1: Challenges of Clustering High-Dimensional Data

Clustering High-Dimensional Data

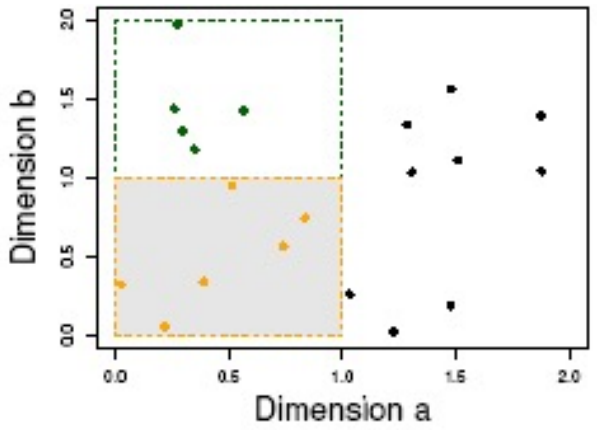
- Why cluster high-dimensional data?
  - How high is high-dimension in clustering?
    - Many clustering algorithms deal with 1-3 dimensions
    - These methods may not work well when the number of dimensions grows to 20
  - Many applications, such as text documents or DNA micro-array data, may need to handles tens of thousands of dimensions
- Major challenges of high-dimensional data clustering
  - Many irrelevant dimensions may mask clusters
  - Distance measure becomes meaningless—due to equidistance
  - Clusters may exist only in some subspaces

The Curse of Dimensionality

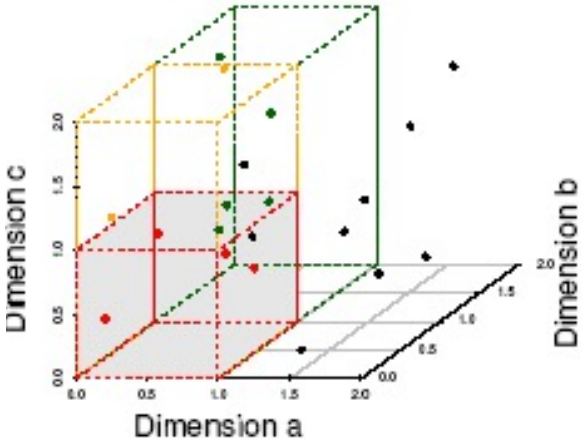
- Data in only one dimension is relatively packed
- Adding a dimension *stretches* the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless—due to equidistance
- Traditional distance measure could be dominated by noises in many dimensions



(a) 11 Objects in One Unit Bin



(b) 6 Objects in One Unit Bin

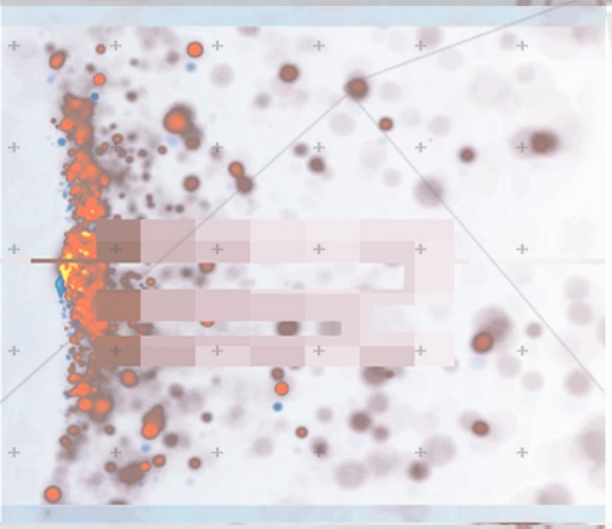


(c) 4 Objects in One Unit Bin

Ack. Graphs adapted from Parsons et al., Subspace clustering for high dimensional data: A review. SIGKDD Explorations 2004)

Curse of Dimensionality: Five Different Aspects

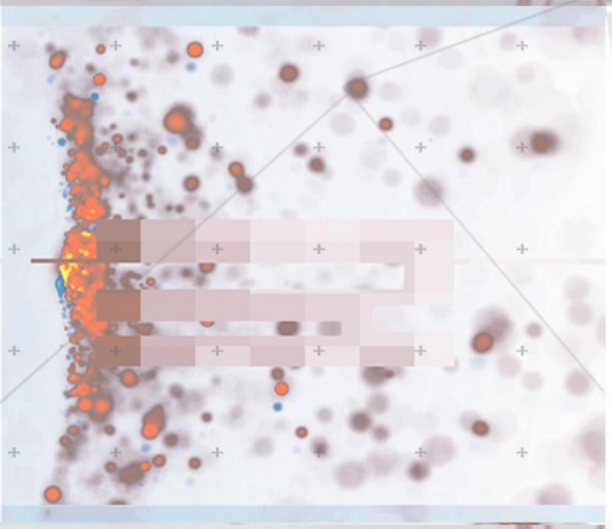
- **Optimization**: The difficulty of global optimization increases exponentially with an increase in the number of dimensions
- **Distance concentration effect** of L_p-norms
  - Distance concentration: Far and close neighbors have similar distances
  - The relative contrast of L_p distances diminishes as dimensionality increases
- **Irrelevant attributes** can interfere with the performance of clustering for that object
  - The relevance of certain attributes may differ for different groups of objects
- **Correlated attributes**: Strong correlation among a subset of attributes can be used to reduce dimensionality
  - The *intrinsic dimensionality* of a dataset can be considerably lower than *embedded dimensionality* (i.e., the number of features of the dataset)
- **Data sparsity**: Data volume in high-dimensional space is extremely sparse



Session 2: Methods for Clustering High-Dimensional Data

Methods for Clustering High-Dimensional Data

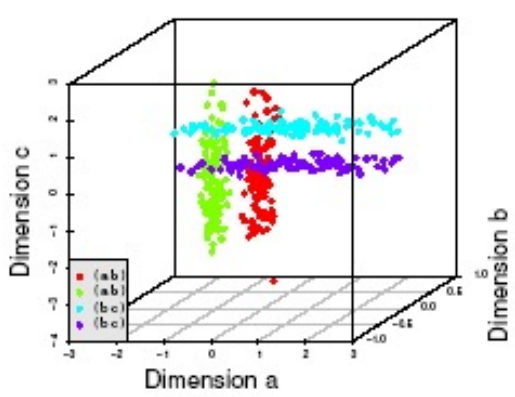
- Methods can be grouped in two categories
  - **Subspace-clustering**: Search for clusters existing in subspaces of the given high dimensional data space
    - CLIQUE, ProClus, and bi-clustering approaches
  - **Dimensionality reduction approaches**: Construct a much lower dimensional space and search for clusters there (may construct new dimensions by combining some dimensions in the original data)
    - Spectral clustering and various dimensionality reduction methods
- Clustering should not only consider dimensions but also attributes (features)
  - **Feature selection:** Useful to find a subspace where the data have nice clusters
  - **Feature transformation:** Effective if most dimensions are relevant
    - PCA (Principal Component Analysis) and SVD (Singular Value Decomposition) are useful when features are highly correlated or redundant

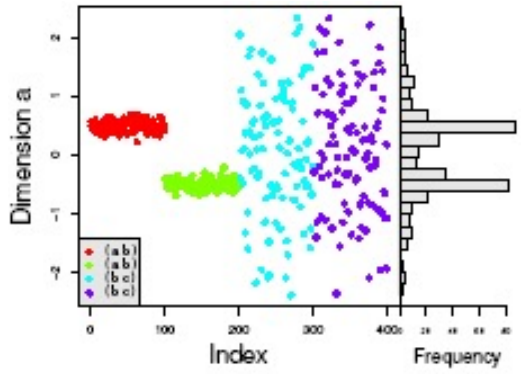


Session 3: Subspace Clustering Methods

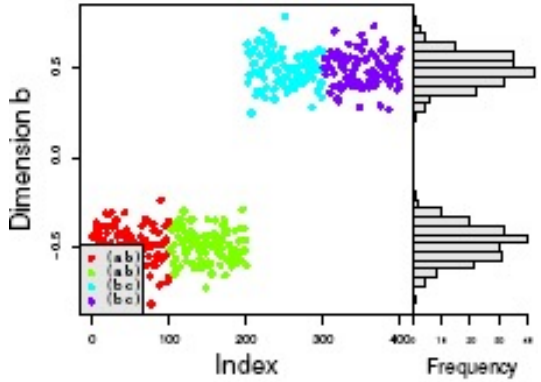
Why Subspace Clustering?

- Clusters may exist only in some subspaces
- Subspace-clustering: Find clusters in all the subspaces

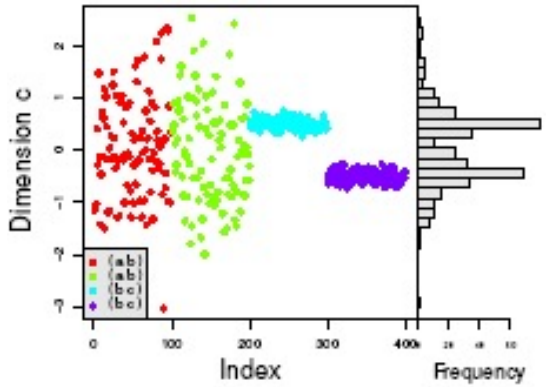




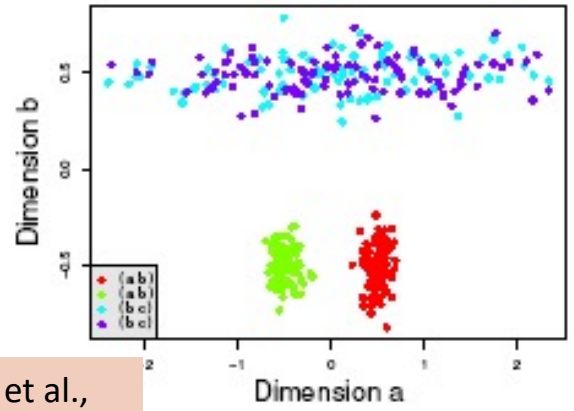
(a) Dimension a



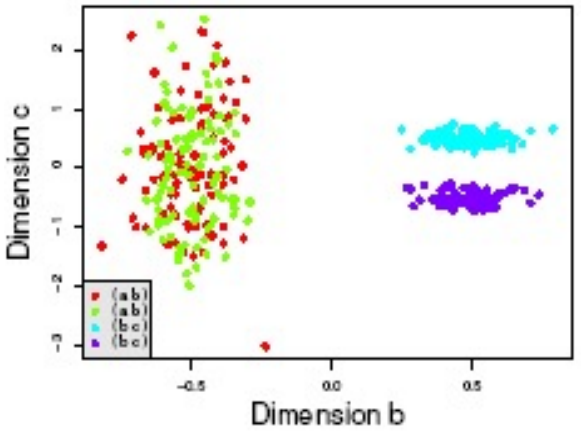
(b) Dimension b



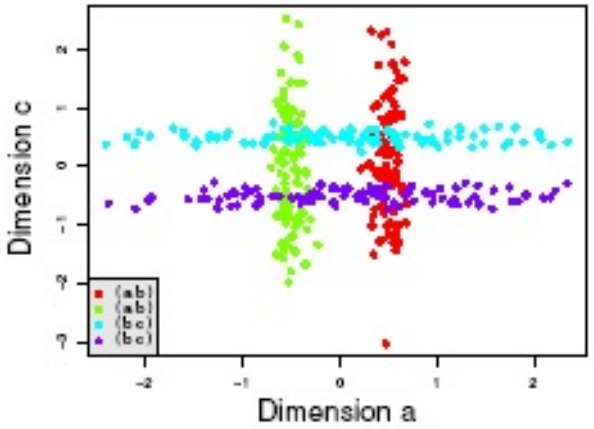
(c) Dimension c



(a) Dims a & b



(b) Dims b & c

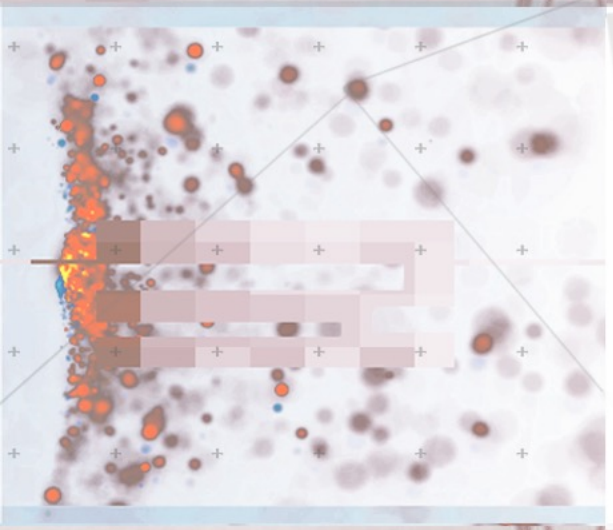


(c) Dims a & c

Ack. Graphs adapted from Parsons et al., Subspace clustering for high dimensional data: A review. SIGKDD Explorations 2004)

Subspace Clustering Methods

- Axis-parallel vs. arbitrarily oriented subspaces
  - Axis-parallel: Subspaces are in parallel with some axes
  - Arbitrarily oriented subspaces
- **Subspace search methods**: Search in axis-parallel subspaces to find clusters
  - Bottom-up approaches
  - Top-down approaches
- **Search and clustering in arbitrarily oriented subspaces**
  - Correlation-based clustering methods
    - E.g., PCA-based approaches
- **Bi-clustering methods**
  - Optimization-based methods
  - Enumeration methods

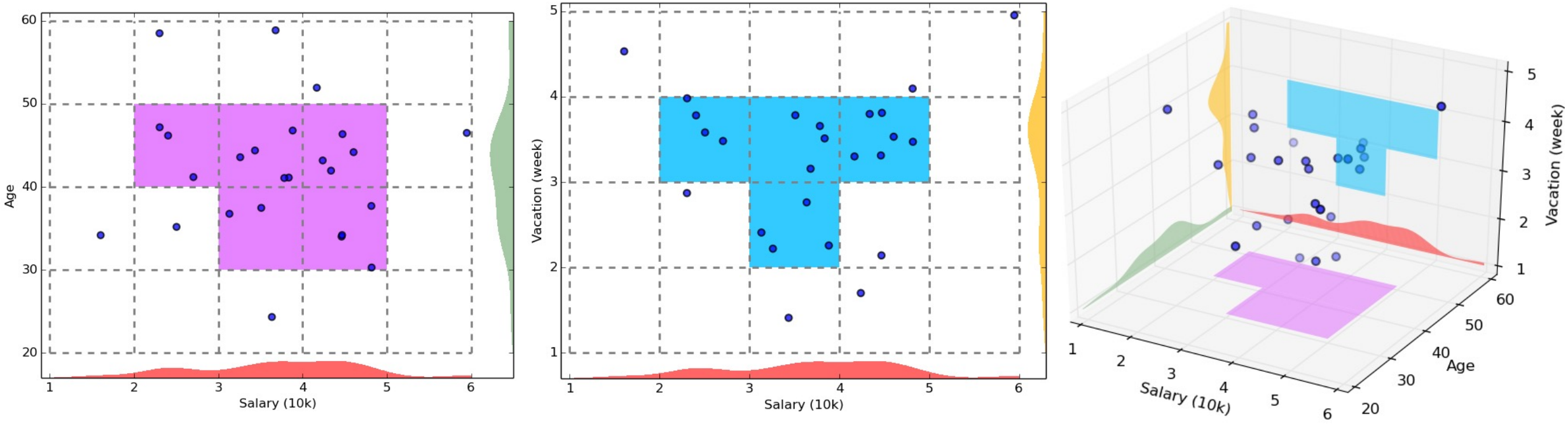


Session 4: Subspace Clustering I: Subspace Search Methods

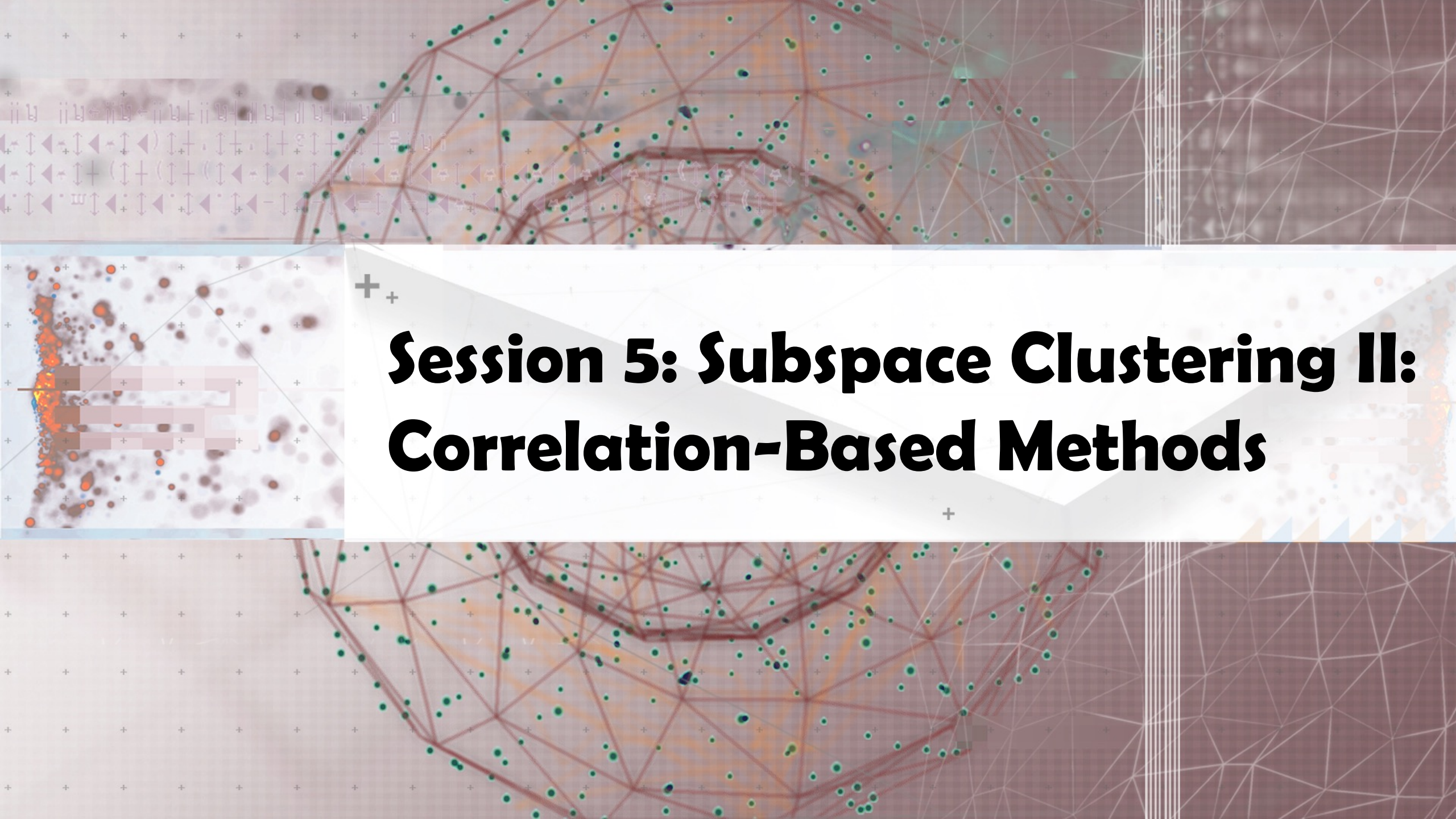
Subspace Clustering: Subspace Search Methods

- Search various subspaces to find clusters
- ***Bottom-up approaches***
  - Start from low-D subspaces and search higher-D subspaces only when there may be clusters in such subspaces
  - Various pruning techniques to reduce the number of higher-D subspaces to be searched
  - Ex. CLIQUE (Agrawal et al. 1998)
- ***Top-down approaches***
  - Start from full space and search smaller subspaces recursively
  - Effective only if the *locality assumption* holds: Restricts that the subspace of a cluster can be determined by the local neighborhood
  - Ex. PROCLUS (Aggarwal et al. 1999): A *k*-medoid-like method

Example of CLIQUE: Density and Grid-Based Subspace Clustering

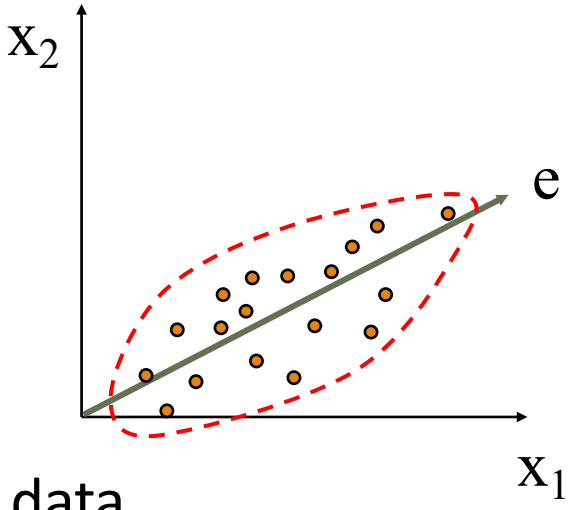


- Start at 1-D space and discretize numerical intervals in each axis into grid
- Find dense regions in each subspace and generate their minimal descriptions (clusters)
- Use the dense regions to find promising candidates in 2-D space (using Apriori principle)
- CLIQUE automatically identifies subspaces of a high dimensional data space and terminates when no more clusters or cluster candidates can be found



Session 5: Subspace Clustering II: Correlation-Based Methods

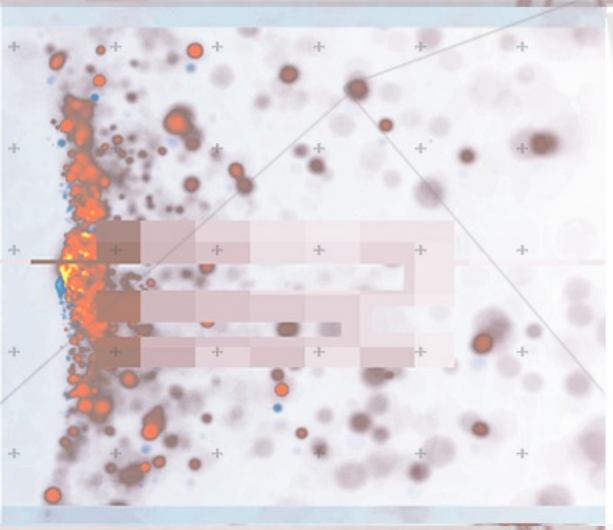
Subspace Clustering: Correlation-Based Methods



- **Subspace search method**
  - Similarity measure is based on distance or density
- **Correlation-based method**: Based on advanced correlation models
  - Ex. PCA (Principal Component Analysis)-based approach
    - Find a projection that captures the largest amount of variation in data
    - Apply PCA to derive a set of new, uncorrelated dimensions (*dimensionality reduction*)
    - Then find clusters in the new space or its subspaces
  - Other space transformation methods
    - Hough transform
    - Fractal dimensions

Simple Illustration of Principal Component Analysis

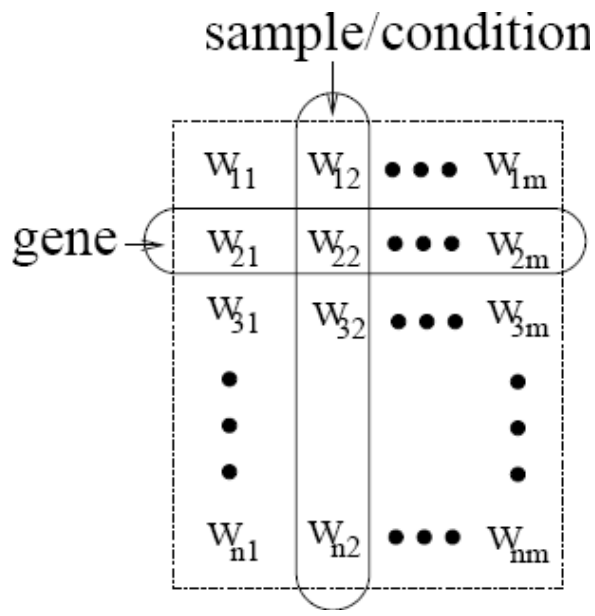
- Given *N* data vectors (numeric data only) from *n*-dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
- Normalize input data: Each attribute falls within the same range
- Compute *k* orthogonal (i.e. linearly uncorrelated) (unit) vectors, i.e., ***principal components***
- Each input vector is a linear combination of the *k* **principal component vectors**
- The principal components are sorted in order of decreasing "significance" or strength
- Eliminate the *weak components* (i.e., those with low variance)
  - That is, using the strongest principal components, it is possible to reconstruct a good approximation of the original data



Session 6: Subspace Clustering III: Bi-Clustering Methods

Subspace Clustering (III): Bi-Clustering Methods

- Bi-clustering: Cluster both objects and attributes simultaneously (treating objects and attributes in a symmetric way)
- Four requirements:
  1. Only a small set of objects participate in a cluster
  2. A cluster only involves a small number of attributes
  3. An object may participate in multiple clusters or does not participate in any cluster at all
  4. An attribute may be involved in multiple clusters or is not involved in any cluster at all
- Ex 1. *Gene expression* or *microarray data*
  - A gene-sample/condition matrix: Each element in the matrix, a real number, records the expression level of a gene under a specific condition
- Ex. 2. Clustering customers and products



$$\begin{matrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{matrix}$$

Types of Bi-Clusters

- Let $A = \{a_1, ..., a_n\}$ be a set of genes, $B = \{b_1, ..., b_m\}$ a set of conditions
- A bi-cluster: A submatrix where genes and conditions follow some consistent patterns
- 4 types of bi-clusters (ideal cases)
  - Bi-clusters with constant values:
    - for any i in I and j *in* J, $e_{ij} = c$

10	10	10	10	10
20	20	20	20	20
50	50	50	50	50
0	0	0	0	0

  - Bi-clusters with constant values in rows:
    - $e_{ij} = c + \alpha_i$
    - Also, it can be constant values in columns

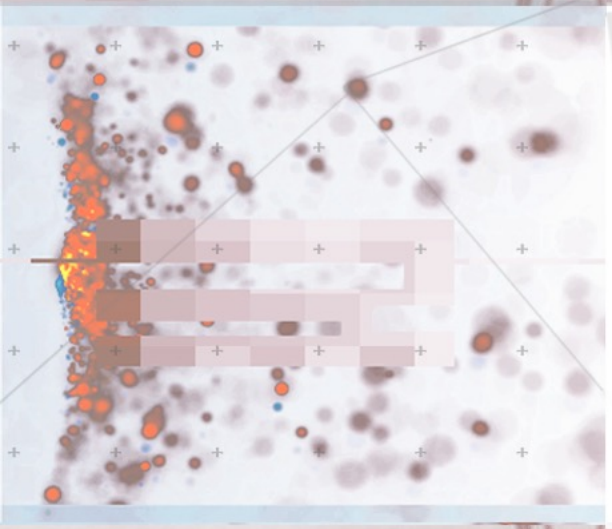
10	50	30	70	20
20	60	40	80	30
50	90	70	110	60
0	40	20	60	10

  - Bi-clusters with *coherent values* (i.e., *pattern-based clusters*)
    - $e_{ij} = c + \alpha_i + \beta_j$
  - Bi-clusters with *coherent* evolutions in rows
    - $(e_{i1j1} - e_{i1j2})(e_{i2j1} - e_{i2j2}) \geq 0$
    - i.e., only interested in the up- or down- regulated changes across genes or conditions without constraining on the exact values

10	50	30	70	20
20	100	50	1000	30
50	100	90	1200	80
0	80	20	100	10

Bi-Clustering Methods

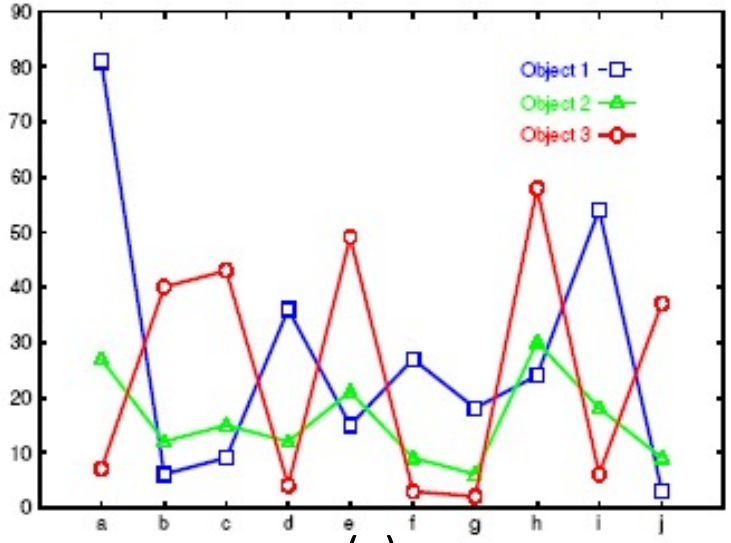
- Real-world data is noisy: Try to find approximate bi-clusters
- Methods: Optimization-based methods vs. enumeration methods
- **Optimization-based methods**
  - Try to find submatrices one at a time to achieve the best significance as a bi-cluster
  - Due to the cost in computation, greedy search is employed to find local optimal bi-clusters
  - Ex. δ-bicluster Algorithm (Cheng and Church, ISMB'2000)
- **Enumeration methods**
  - Use a tolerance threshold to specify the degree of noise allowed in the bi-clusters to be mined
  - Then try to enumerate all submatrices as bi-clusters that satisfy the requirements
  - Ex. δ-pCluster Algorithm (H. Wang et al. SIGMOD'2002, MaPle: Pei et al., ICDM'2003)



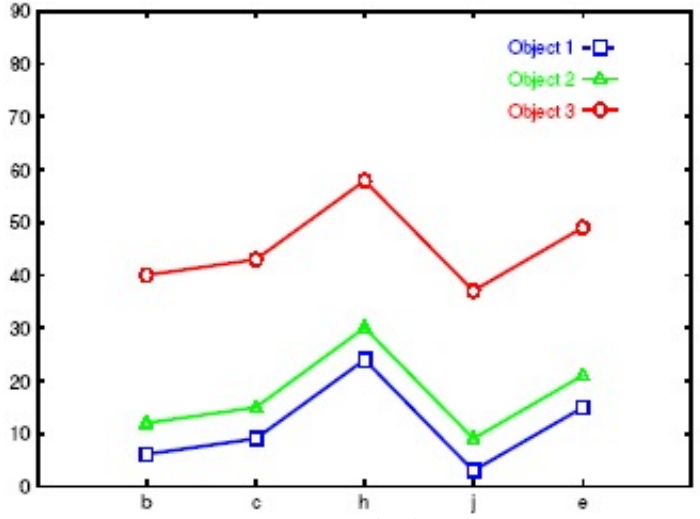
Session 7: Bi-Clustering I: δ-Bi-Clustering

Bi-Clustering for Micro-Array Data Analysis

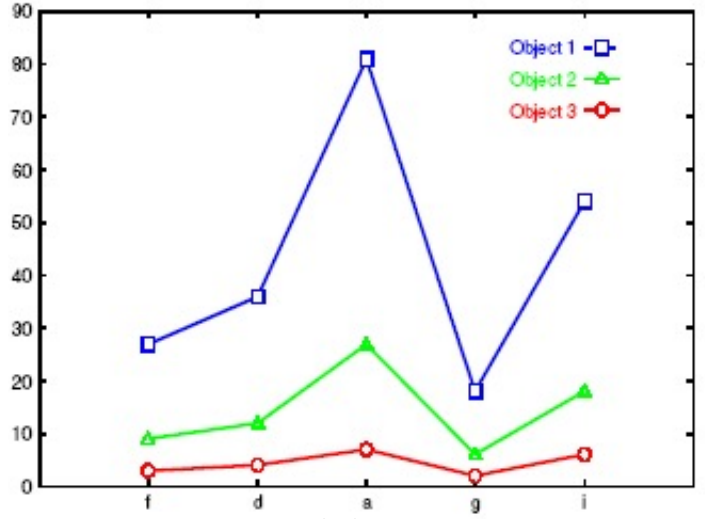
- Figure (a): Micro-array "raw" data shows three objects (e.g., tissues) and their gene values in a multi-dimensional space: Difficult to find their patterns
- Figures (b) and (c): Some subsets of dimensions form nice shift and scaling patterns
- No globally defined similarity/distance measure
- Clusters may not be exclusive
  - A gene can appear in multiple clusters



(a)



(b)



(c)

Bi-Clustering (I): δ-Bi-Cluster

- For a submatrix *I x J*
  - *T*he **mean of the *i*-th row**: $e_{iJ} = \frac{1}{|J|}\sum_{j \in J} e_{ij}$
  - The **mean of the *j*-th column**: $e_{Ij} = \frac{1}{|I|}\sum_{i \in I} e_{ij}$
  - The **mean of all elements in the submatrix**: $e_{IJ} = \frac{1}{|I||J|}\sum_{i \in I, j \in J} e_{ij} = \frac{1}{|I|}\sum_{i \in I} e_{iJ} = \frac{1}{|J|}\sum_{j \in J} e_{Ij}$
- The **quality of the submatrix as a bi-cluster** can be measured by the *mean squared residue* value

$$H(I \times J) = \frac{1}{|I||J|}\sum_{i \in I, j \in J} (e_{ij} - e_{iJ} - e_{Ij} + e_{IJ})^2$$

- A submatrix *I x J* is **δ-bi-cluster** if $H(I \times J) \leq \delta$ where $\delta \geq 0$ is a threshold
  - When $\delta = 0$, *I x J* is a perfect bi-cluster with coherent values
  - By setting $\delta > 0$, a user can specify the tolerance of average noise per element against a perfect bi-cluster
    - residue(e_{ij}) = $e_{ij} - e_{iJ} - e_{Ij} + e_{IJ}$

Bi-Clustering (I): The δ-Bi-Cluster Algorithm

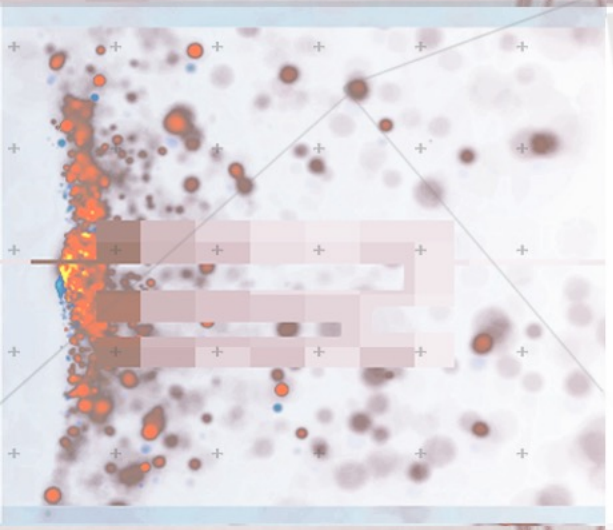
- **Maximal δ-bi-cluster**
  - A δ-bi-cluster *I x J* s.t. no other δ-bi-cluster *I' x J' which* contains *I x J*
- Computing is costly: Use heuristic greedy search to obtain local optimal clusters
- Two phase computation: *Deletion phase* and *addition phase*
  - **Deletion phase:**
    - Start from the whole matrix, iteratively remove rows and columns while the mean squared residue of the matrix is over δ
    - At each iteration, for each row/*column*
      - Compute the *mean squared residue:*

$$d(i) = \frac{1}{|J|}\sum_{j\in J}(e_{ij} - e_{iJ} - e_{Ij} + e_{IJ})^2 \qquad d(j) = \frac{1}{|I|}\sum_{i\in I}(e_{ij} - e_{iJ} - e_{Ij} + e_{IJ})^2$$

      - Remove the row or column of the largest mean squared residue

Bi-Clustering (I): The δ-Bi-Cluster Algorithm (Cont.)

- Two phase computation: *Deletion phase* and *addition phase* (continued)
  - **Addition phase:**
    - Expand iteratively the δ-bi-cluster *I x J* obtained in the deletion phase as long as the δ-bi-cluster requirement is maintained
    - Consider all the rows/columns not involved in the current bi-cluster *I x J* by calculating their mean squared residues
    - A row/column of the smallest mean squared residue is added into the current δ-bi-cluster
- It finds only one δ-bi-cluster, thus needs to run multiple times
  - By replacing the elements in the output bi-cluster by random numbers
- A quite costly search process



Session 8: Bi-Clustering II: δ-pClustering

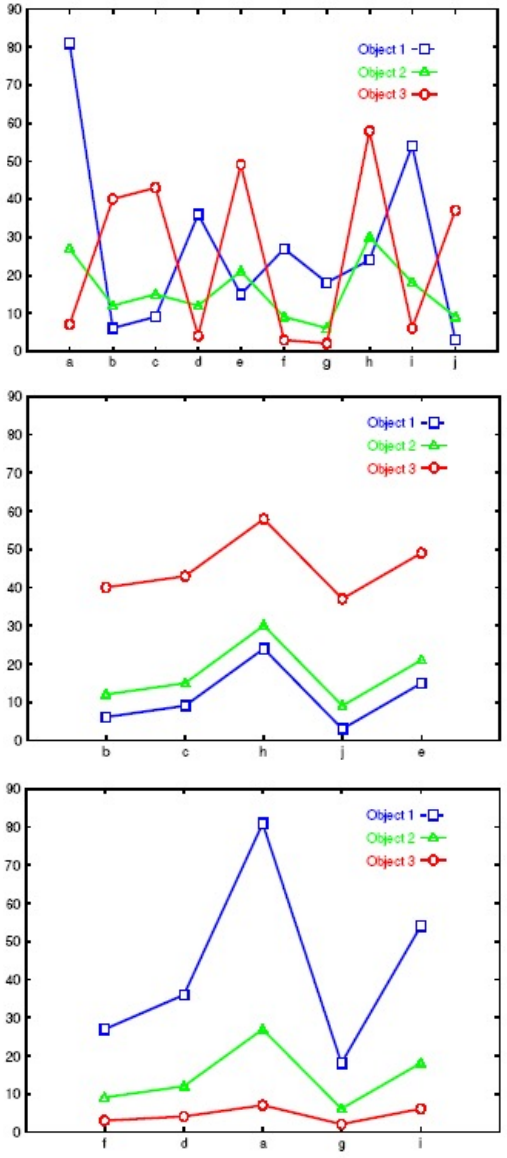
Bi-Clustering (II): δ-pCluster: Clustering by Pattern Similarity

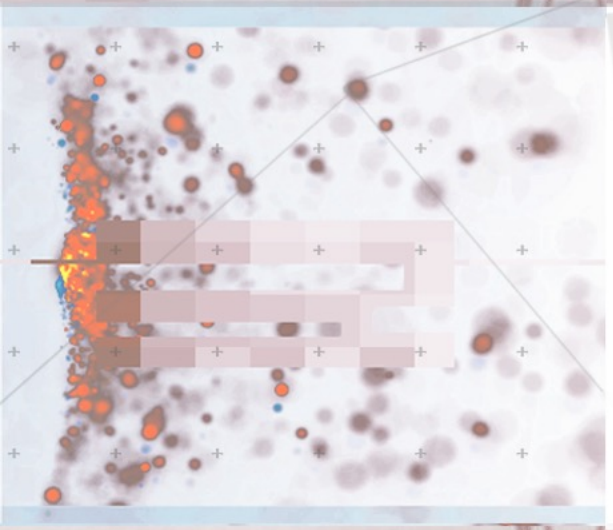
- Clustering by pattern similarity (δ-pClusters) [H. Wang, et al., SIGMOD'02]
- A submatrix *I x J* is a bi-cluster with (perfect) coherent values if and only if $e_{i1j1} - e_{i2j1} = e_{i1j2} - e_{i2j2}$
  - For any 2 *x* 2 submatrix of *I x J*, $p\text{-score}\begin{pmatrix} e_{i_1j_1} & e_{i_1j_2} \\ e_{i_2j_1} & e_{i_2j_2} \end{pmatrix} = |(e_{i_1j_1} - e_{i_2j_1}) - (e_{i_1j_2} - e_{i_2j_2})|$
- A submatrix *I x J* is a **δ-pCluster** (pattern-based cluster) if the *p*-score of every 2 *x* 2 submatrix of *I x J* is at most δ, where δ ≥ 0 is a threshold specifying a user's tolerance of noise against a perfect bi-cluster
- The *p*-score controls the noise on every element in a bi-cluster, while the mean squared residue captures the average noise
- **Monotonicity**: If *I* x *J* is a δ-pCluster, every *x* x *y* (*x,y* ≥ 2) submatrix of *I* x *J* is also a δ-pCluster
- A δ-pCluster is **maximal** if no more rows or columns can be added to still make it retain as a δ-pCluster—We only need to compute all maximal δ-pClusters

More on δ-pClustering and Efficiency Improvement (MaPle)

- Additional advantages of *δ-p*Clusters:
  - Containing no outliers: Due to the averaging effect, *δ*-bi-cluster may contain outliers but still within *δ*-threshold
  - For scaling patterns, taking logarithmic on $\frac{d_{xa} / d_{ya}}{d_{xb} / d_{yb}} < \delta$

    will lead to the same *p*-score form
- Further improving mining efficiency (MaPle: Pei et al. ICDM'03)
- Framework: A pattern-growth approach in frequent pattern mining (Algorithm is similar to mining frequent closed itemsets)
- For each condition combination J, find the maximal subsets of genes I such that I x J is a *δ-p*Clusters
  - If I x J is not a submatrix of another *δ-p*Clusters
  - then I x J is a maximal *δ-p*Cluster

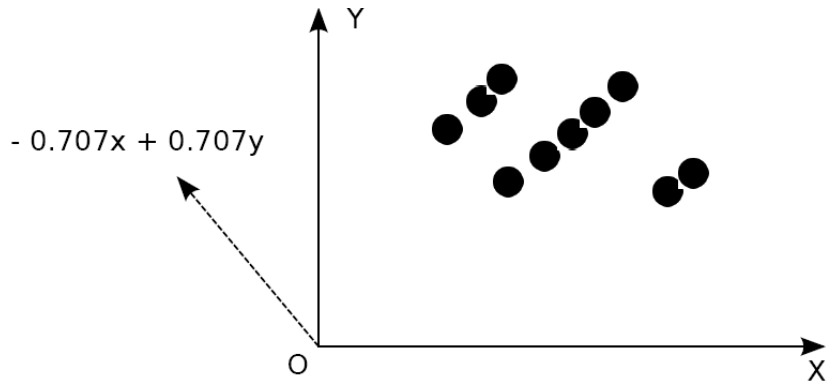




Session 9: Dimensionality Reduction Methods

Dimensionality Reduction

- Dimensionality reduction
  - In some situations, it is more effective to construct a new space instead of using some subspaces of the original data
  - Ex. To cluster the points in the figure, any subspace of the original one, X and Y, cannot help since all the three clusters projected to X and Y axes will overlap
    - Upon constructing a new dimension such as the dashed one, the three clusters become apparent as the points are projected into the new dimension

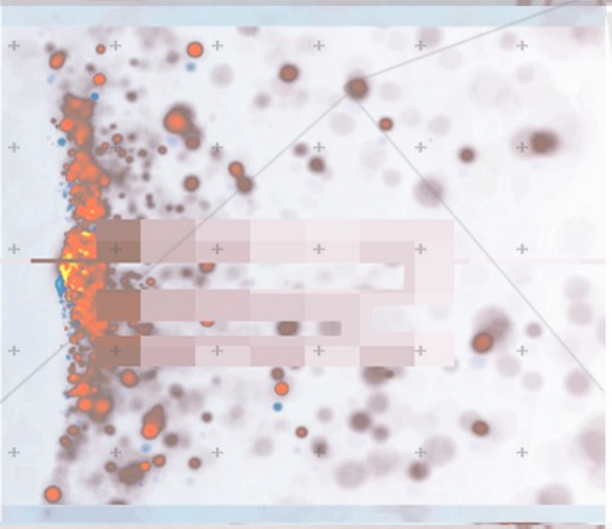


Dimensionality-Reduction Methods

- Feature selection and extraction may not focus on clustering structure finding
- Dimensionality reduction: Reduce dimensionality by mathematical transformation
  - **Nonnegative matrix factorization (NMF)** Will briefly outline the idea in the next slide
    - One high-dimensional sparse nonnegative matrix factorizes approximately into two low-rank matrices
  - **Spectral clustering** To be covered in Lecture 10
    - Uses the *spectrum* of the similarity matrix of the data to perform dimensionality reduction for clustering in fewer dimensions
    - Combining feature extraction and clustering
    - **Typical spectral clustering methods**
      - Normalized Cuts (Shi and Malik, CVPR'97 or PAMI'2000)
      - The Ng-Jordan-Weiss algorithm (NIPS'01)

Clustering by Nonnegative Matrix Factorization (NMF)

- Nonnegative matrix factorization (NMF)
  - A nonnegative matrix $A_{n\times d}$ (e.g., word frequencies in documents) can be approximately factorized into two nonnegative lower rank matrices $U_{n\times k}$ and $V_{k\times d}$ *(k << d, n)*:
    - $A_{n\times d} \approx U_{n\times k}\, V_{k\times d}$ (or, $A \approx U\, V$)
  - Residue matrix R represents the noise in the underlying data: $R = A - U\, V$
- Constrained optimization: Determine U and V so that the sum of the square of the residuals in R is minimized
- U and V simultaneously provide the clusters on the rows (docs) and columns (words): Another kind of co-clustering
  - $U_{n\times k}$: the components of each of n objects mapped into each of k newly created dimensions
  - $V_{k\times d}$: Each of k newly created dimensions in terms of the original d dimensions
- Advantage: Interpretability of NMF—A data point can be expressed as a nonnegative linear combination of the concepts in the underlying data



Summary

Summary: Clustering High-Dimensional Data

- Challenges of Clustering High-Dimensional Data
- Methods for Clustering High-Dimensional Data
- Subspace Clustering Methods
  - Subspace Clustering I: Subspace Search Methods
  - Subspace Clustering II: Correlation-Based Methods
  - Subspace Clustering III: Bi-Clustering Methods
    - δ-Bi-Clustering
    - δ-pClustering
- Dimensionality Reduction Methods

Recommended Readings

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. *SIGMOD'98*
- C. C. Aggarwal, C. Procopiuc, J. Wolf, P. S. Yu, and J.-S. Park. Fast Algorithms for Projected Clustering. *SIGMOD'99*
- Y. Cheng and G. Church. Biclustering of Expression Data. *ISMB'00*
- H.-P. Kriegel, P. Kroeger, and A. Zimek. Clustering High Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering. *TKDD'*09
- S. C. Madeira and A. L. Oliveira. Bi-clustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1, 2004
- L. Parsons, E. Haque, and H. Liu. Subspace Clustering for High Dimensional Data: A Review. *ACM SIGKDD Explorations*, 6(1):90–105, 2004.
- J. Pei, X. Zhang, M. Cho, H. Wang, and P. S. Yu. Maple: A Fast Algorithm for Maximal Pattern-Based Clustering. *ICDM'03*
- H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by Pattern Similarity in Large Data Sets. *SIGMOD'02*
- A. Zimek. Clustering High-Dimensional Data (Chapter 9), in C. Aggarwal and C. K. Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press, 2014