
Unsupervised Machine Learning

Erzsébet Merényi
Department of Statistics
and Department of Electrical and Computer Engineering
Rice University, Houston, Texas

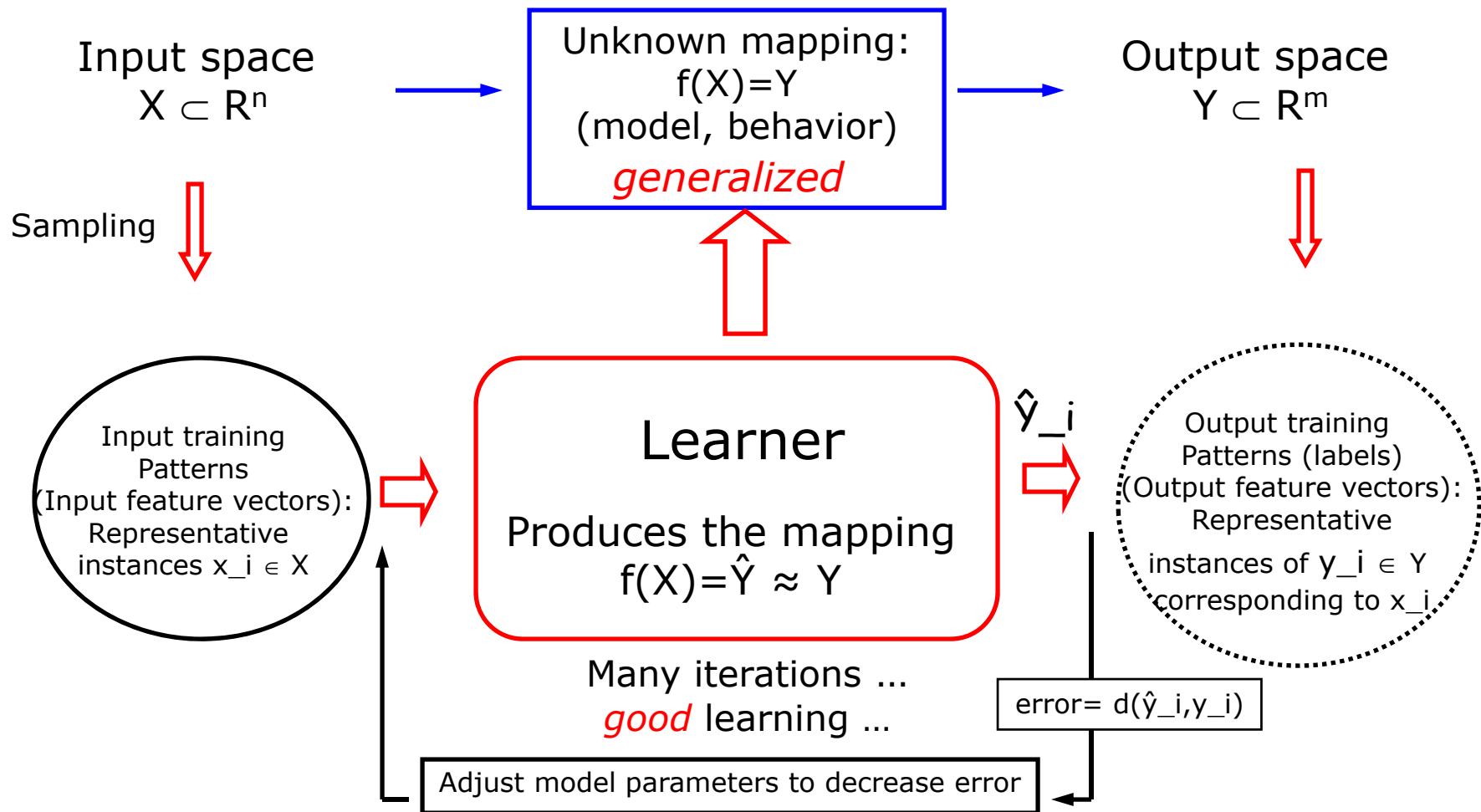


Credits: Parts are joint work with current and former graduate students
Josh Taylor, Patrick O'Driscoll, Brian Bue, Lili Zhang, Kadim Taşdemir, Maj. Michael Mendenhall,
Abha Jain
and many collaborators



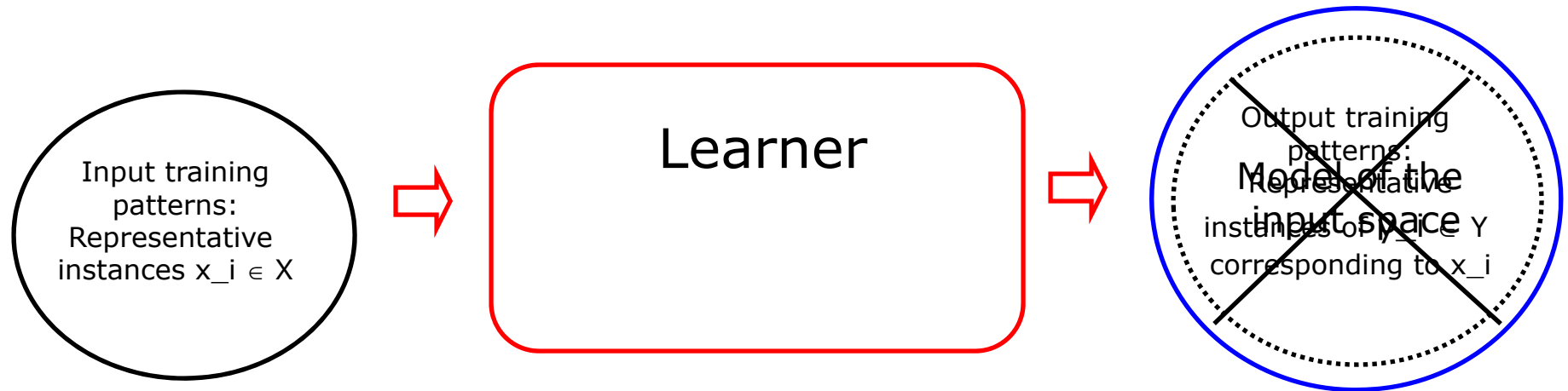
Learning With a Teacher

(supervised learning)



Learning Without a Teacher

(unsupervised learning)



An unsupervised (self-organized) learner captures some internal characteristics of the data space (data manifold): structure, mixing components / latent variables, ...

- Ex: clusters
- Ex: principal components
- Ex: independent components

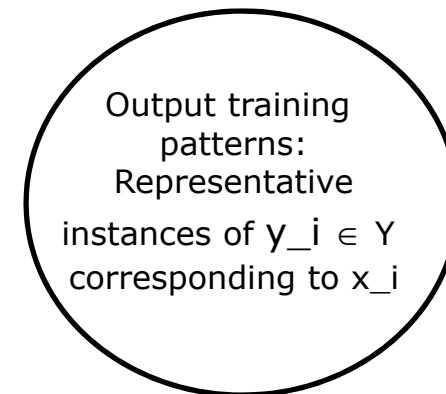
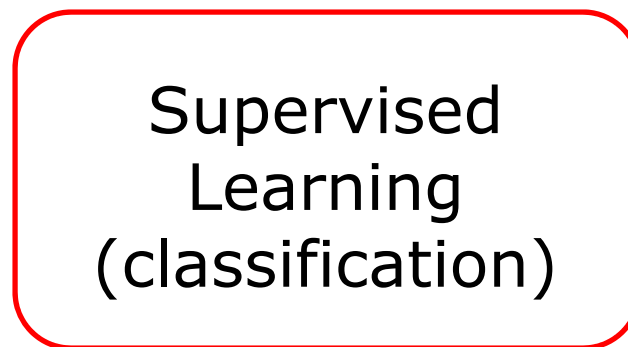
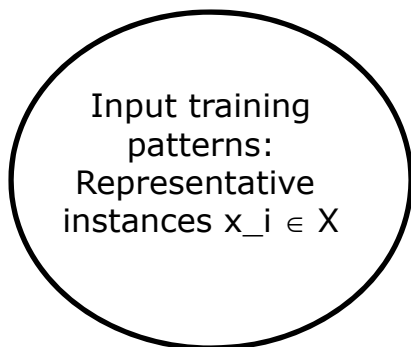
Phases of the Full Learning Process

Unsupervised + Supervised

1.

- Phase 1 allows
- New discovery
 - Detection of mislabeled “known” samples, or missed class
- Prevent confusion of supervised learner -> poor learning results

3.



2.



labeling

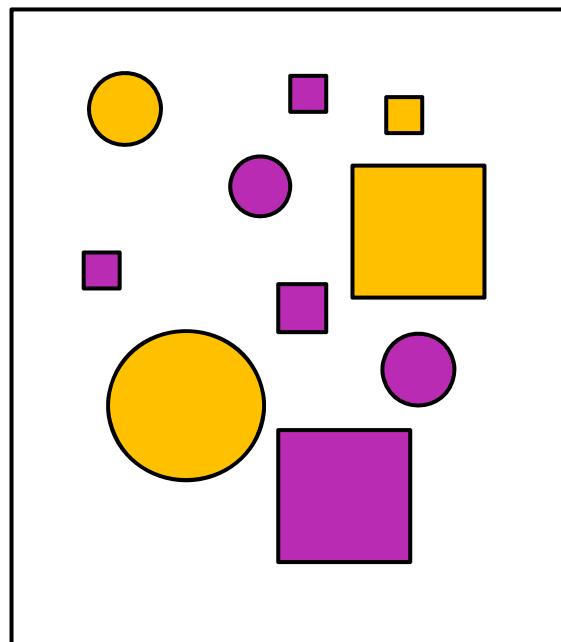


Feature Vector (Data Point) in n-Space

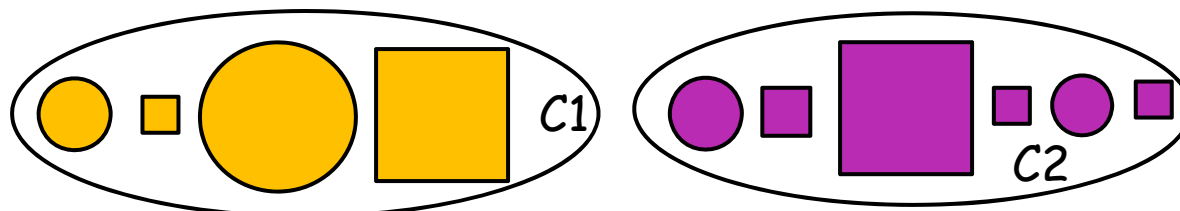
- Input to a learning algorithm
- Vector of descriptors for an object of interest in physical space: $\mathbf{x} \in R^n$
- Ex: Descriptors for a galaxy
 - Image – unfolded to a vector of pixel values
 - Vector of derived statistics: mean brightness, width, eccentricity, RGB color values, ...
 - Spectrum
 - Combinations
- Ex: Descriptors for a dark matter / dark energy phenomenon - ?
- The choice of descriptors is important: must characterize the objects from the point of view of the problem!
- Objects close in physical (problem) space may not be close in feature space – and vica versa
 - Careful with using image (spatial) context – can help; or can lose important discovery of small size in physical space



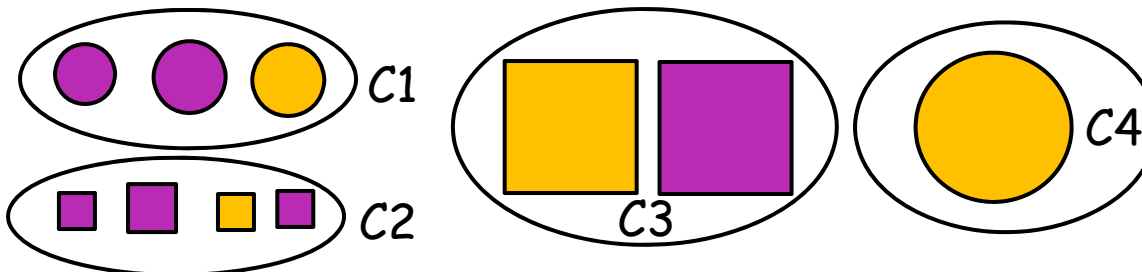
Choice of Feature Vector



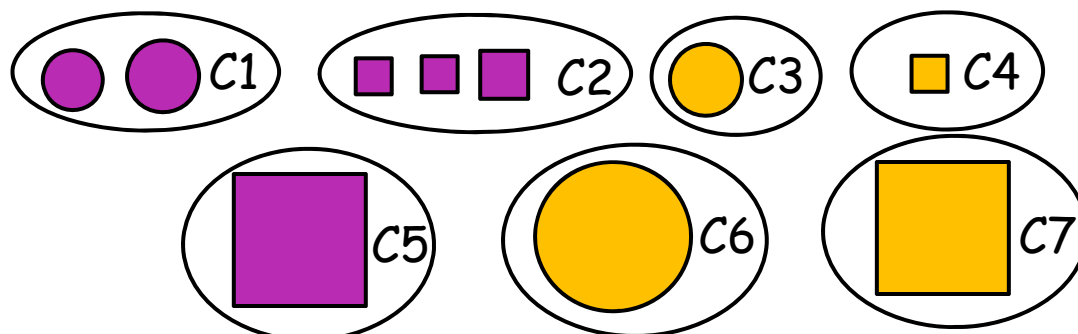
Feature vector: (R,G,B) =>



Feature vector: ("roundness", radius) =>



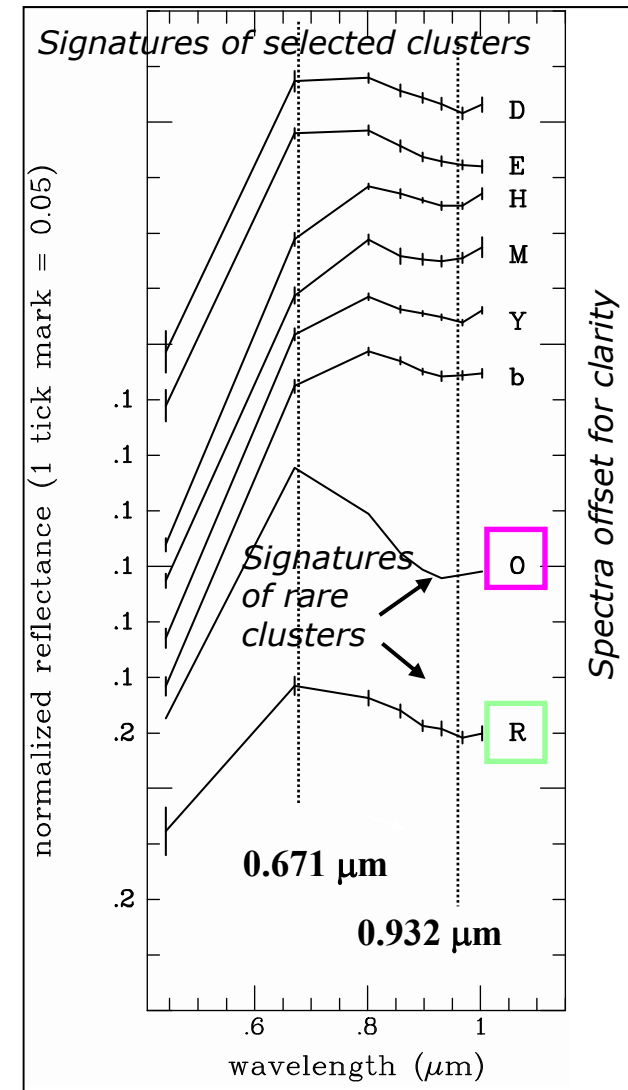
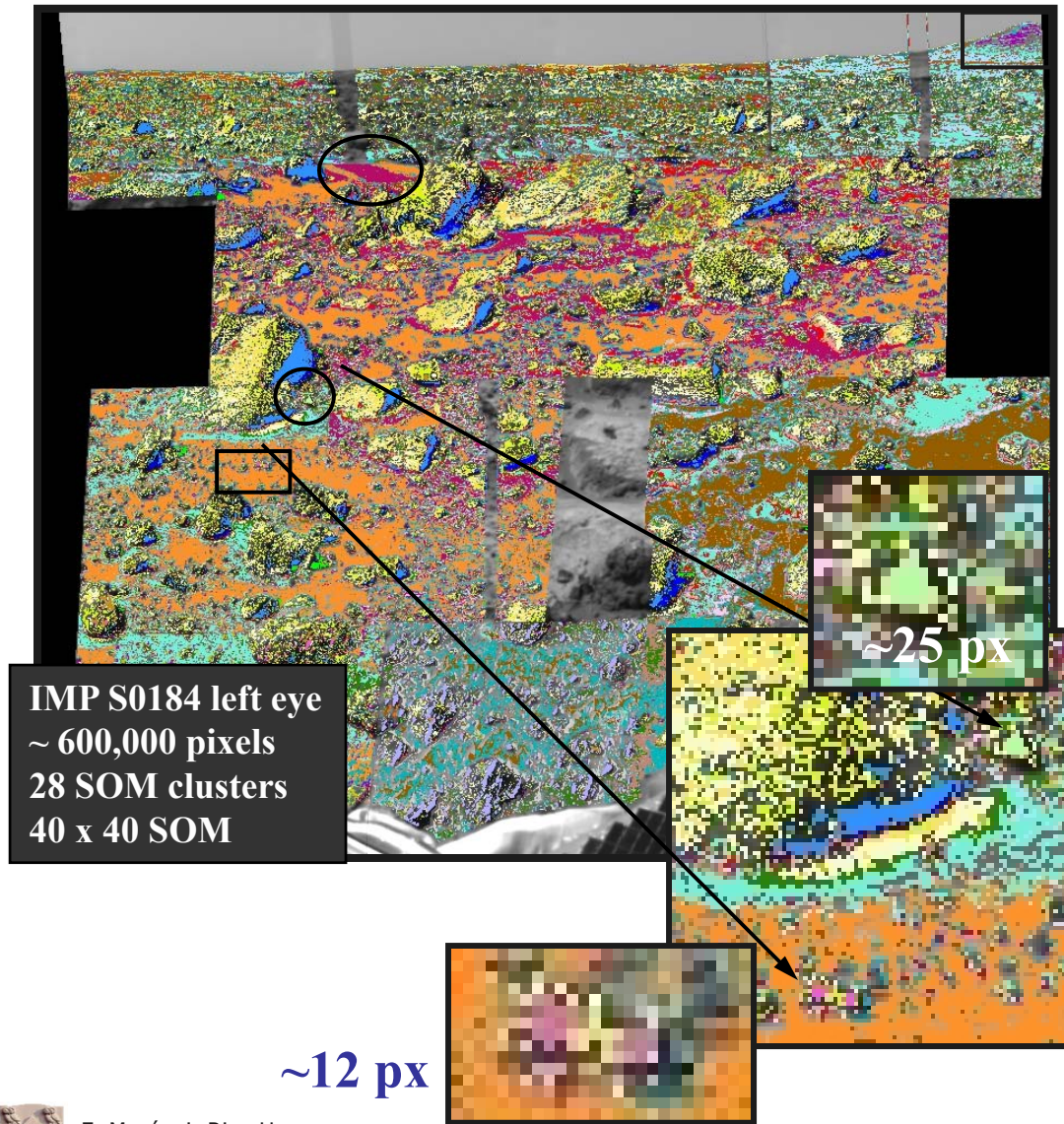
Feature vector: (R,G,B,"curvature", radius) =>



This matches our intuitive categorization better

Finding Clusters of Rare Materials on Mars

Data: VIS-NIR Spectral Imagery, Imager for Mars Pathfinder; Colors: clusters



Farrand et al. *Int'l Mars J.* 2008



E. Merényi, Rice U
erzsebet@rice.edu

Unsupervised Learning

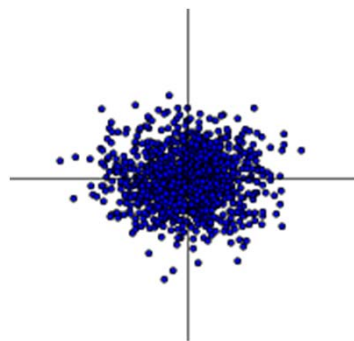
(an incomplete view)

In general, seeks to model the structure of data space from unlabeled data: estimation / identification of the distribution

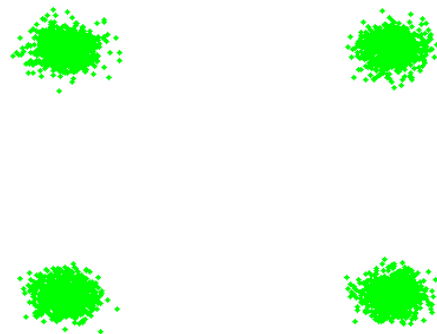
- Finding the (relative) concentration(s) of data points – and topology
- Summarize & explain the key features / relationships in the data

Complexity is the major challenge!

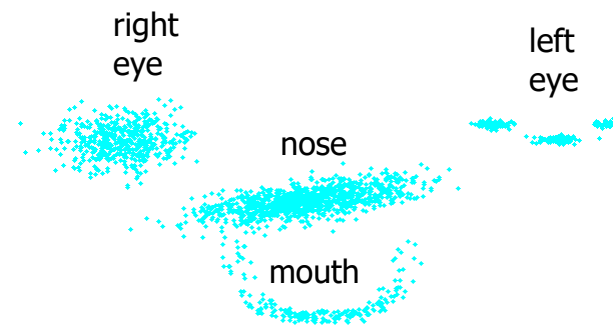
Data sets with same feature dimensionality ($n=2$), same # of points (N) but with increasing structural complexity pose different level of challenge for identifying the structure



Simple



Simple multimodal



"Clown"
(Vesanto & Alhoniemi,
IEEE TNN, 2000)



Complex (Complicated) Data Space

Challenges

- High dimensionality
- Large volume
- Multi-modal (has clusters)
- Highly structured
 - Not linearly separable
 - Widely varying shapes and sizes
 - ... densities (vary within and across clusters)
 - ... proximities
 - ... local dimensionalities

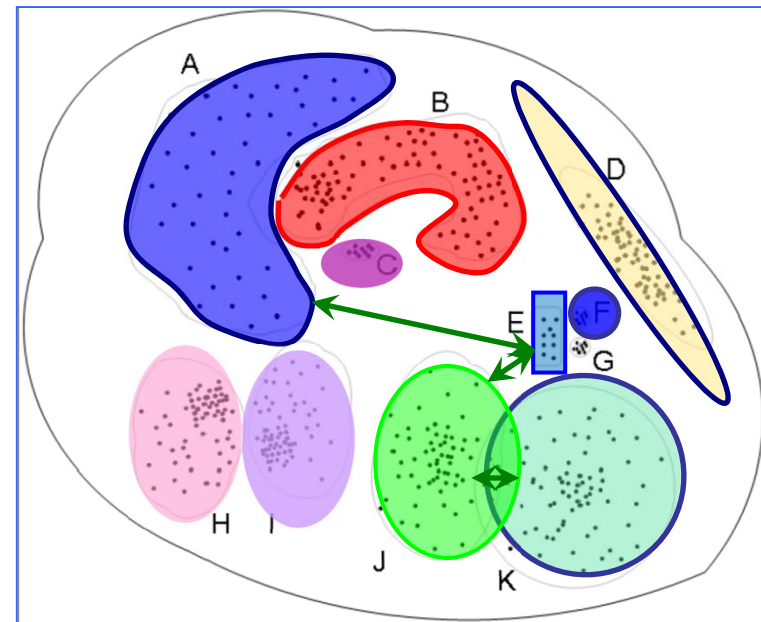
No statistical models

To faithfully learn data relations, and to keep discovery potential, no (or least) assumption should be made about the structure.

Let the data speak.

Imagine in 100 dimensions!

Highly structured data space




Merényi, Taşdemir, Zhang, Springer, LNAI 5400. 2009

Ex: K-means is tuned to capture spherical / ellipsoidal clusters. Can't capture irregulars.

Unsupervised Learning

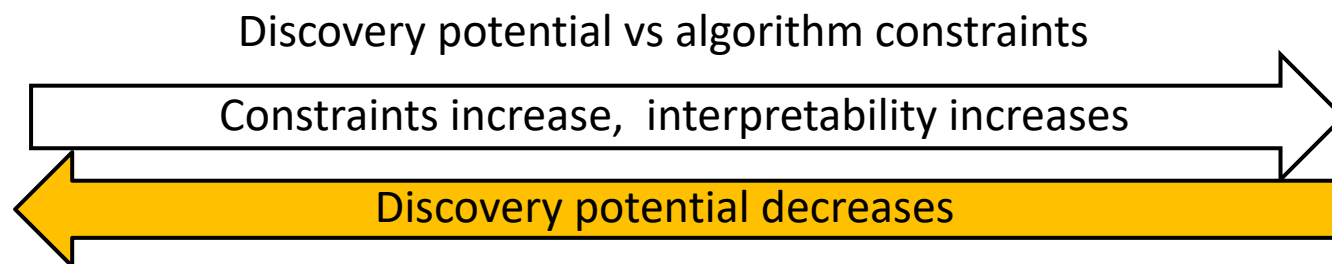
(an incomplete view)

Major approaches

- (Kernel) density estimation / mixture modeling
- Latent variable models such as PCA, ICA, SVD factorization (BSS)
- Anomaly detection (really, any of the others)
- Cluster analysis  Concentrate on this

Various overlaps and correspondences exist across these categories.

T. Heskes, IEEE TNN 2001: links between mixture modeling, VQ and SOM

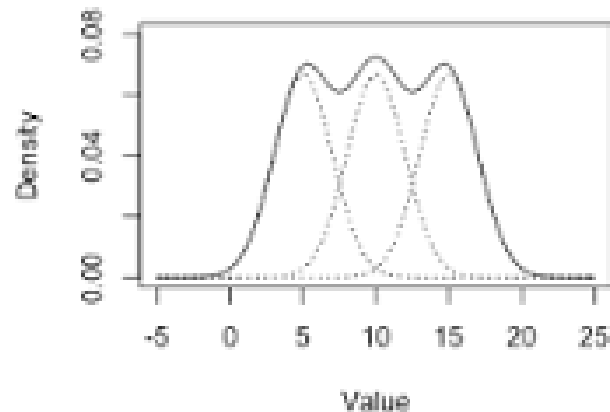


Unsupervised Learning

(an incomplete view)

■ Density estimation / mixture modeling

- Model the data with a weighted sum of functions (linear)
- Predefined functional form
- Predefined # of functions
- EM often used for determining the parameters of fit (parameters of the functions and mixing weights)



Density estimation with mixtures of Gaussians

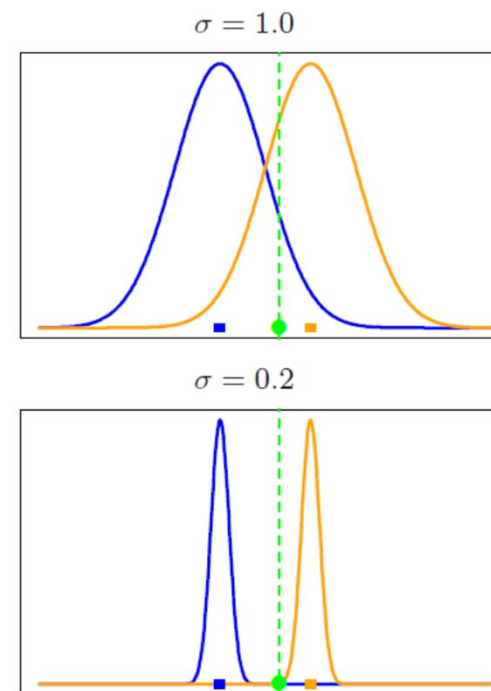


Figure from Hastie et al, 2008

Relation to clustering: Mixture components can be viewed as clusters.

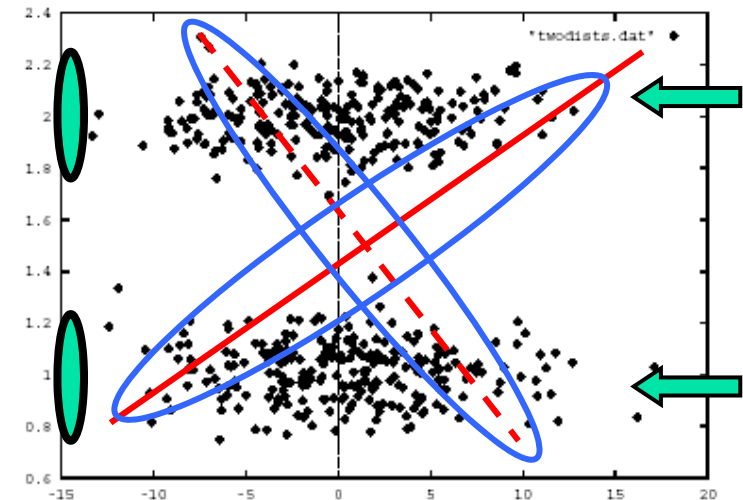


Unsupervised Learning

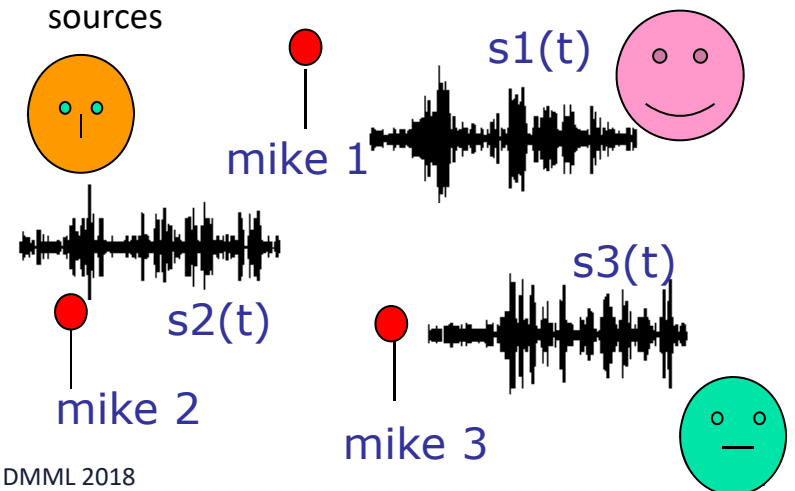
(an incomplete view)

- Latent variable models
 - Also mixtures of “components”, which represent clusters (classes)
 - Components are *not predefined functions*, derived from data, along with mixing weights
 - # of components predefined
 - Mostly linear mixtures
 - Non-linear extensions exist but difficult
- PCA: Finds uncorrelated (lin. Independent) components -> limited to 2nd order stats
 - vast literature, widely available code
 - SVD: More general version of PCA
- ICA: Finds statistically independent components – uses higher order statistics -> finds more interesting structure
 - Different approaches (information theor., neural, statistical, see Ref)

Structure seen by ICA but not by PCA



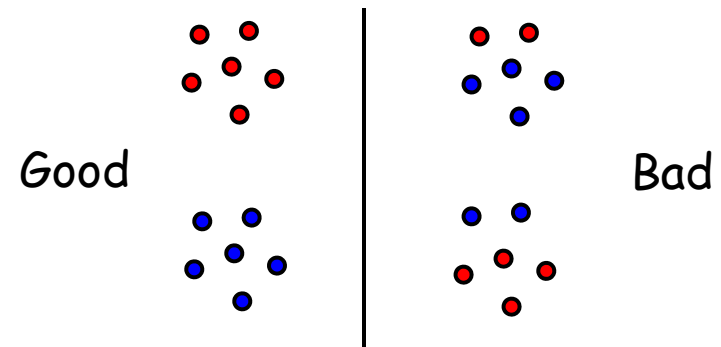
Cocktail Party problem – unmixing unknown sources



Clustering

(somewhat arbitrary, biased)

- **Goal:** To partition the data space into segments (clusters) such that points within a cluster are closer to one another than to any point in any of the other clusters.

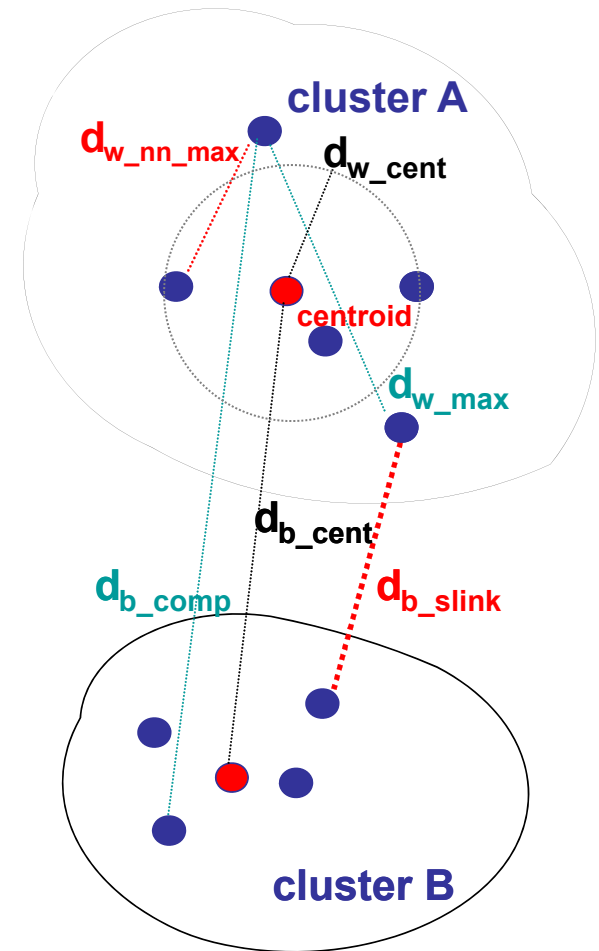


- **Measure of clustering quality without labeled data:** assesses how well the clusters match the natural partitions (chicken – egg?)
- Function of some distortion or intrinsic data relation within and across clusters; depends on the measure of similarity / dissimilarity metric used
 - Metrics often distance-based (similarity = proximity, dissimilarity = distance)
 - Other measures can be used, which are not distances in mathematical sense
 - Ex: Kullback-Leibler divergence; Connectivity measure
- Frequently used: **cluster validity index (CVI)**
 - Review of CVI-s in Bezdek & Pal, 1998; Taşdemir & Merényi, 2011
 - Others: Entropy, modularity, Gap statistic



Cluster Validity Indices (CVI)

- Most CVI-s measure the **ratio** of **separation** between clusters and **scatter** within clusters (aka between clusters and within-cluster distance).
- **Separation** and **scatter** are often calculated from distances
 - Between-cluster distance metrics
 - Centroid linkage
 - Complete linkage
 - Single linkage
 - ...
 - Within-cluster distance metrics:
 - Average distance to cluster centroid, d_{w_cent}
 - Maximum distance between any pair, d_{w_max}
 - Maximum of nearest neighbor distances $d_{w_nn_max}$
 - more ...



Approaches to Index Construction

Classic measures: GDI, DBI – misjudge complex clusterings

- minimum separation/
maximum scatter

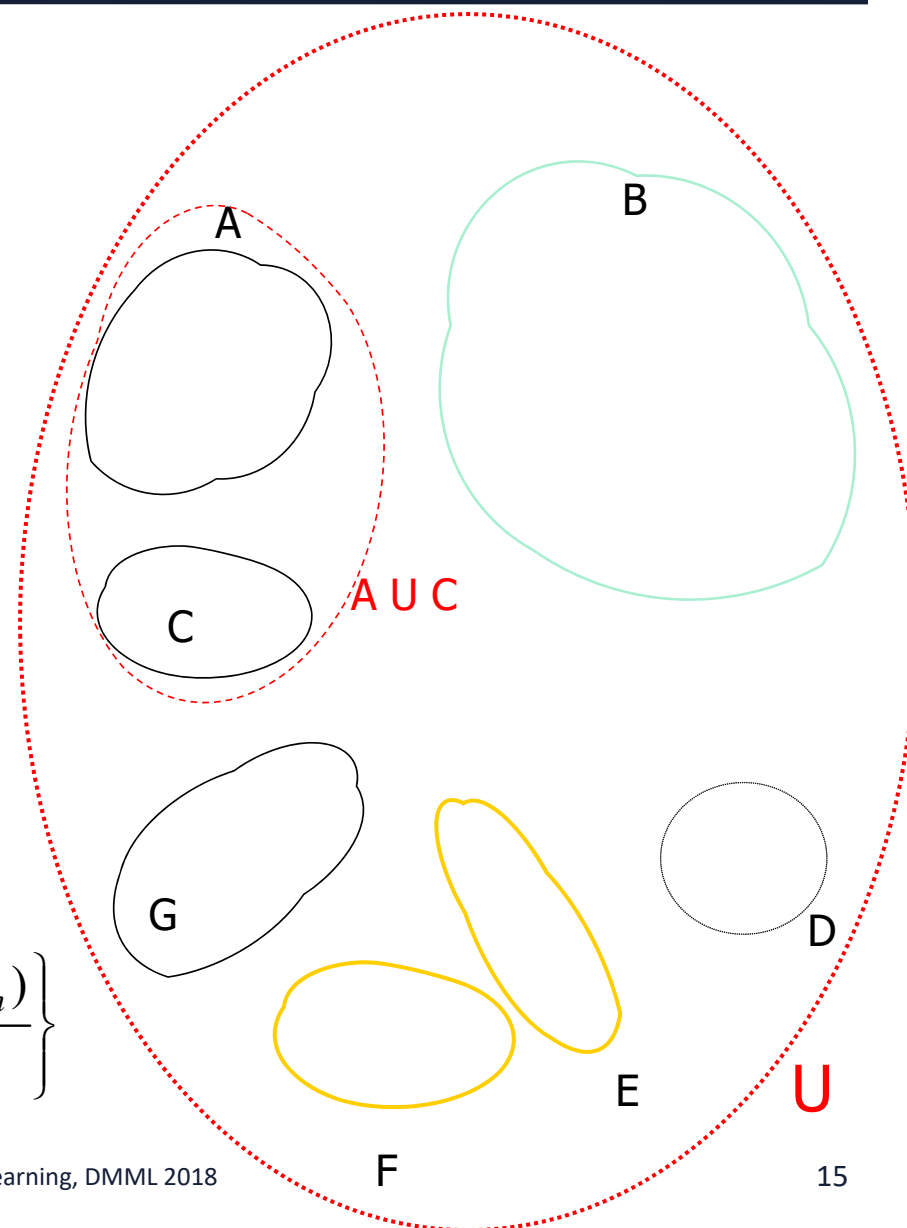
- Ex: GDI

$$GDI(U) = \min_{1 \leq s \leq c} \left\{ \min_{\substack{1 \leq t \leq c \\ t \neq s}} \left\{ \frac{d_{b_i}(C_s, C_t)}{\max_{1 \leq k \leq c} \{d_{w_j}(C_k)\}} \right\} \right\}$$

- average of (scatter/separation)

- Ex: DBI (Davies & Bouldin, 1979)

$$DBI(U) = \frac{1}{|c|} \sum_{k=1}^c \max_{l, l \neq k} \left\{ \frac{d_{w_cent}(C_k) + d_{w_cent}(C_l)}{d_{b_cent}(C_k, C_l)} \right\}$$

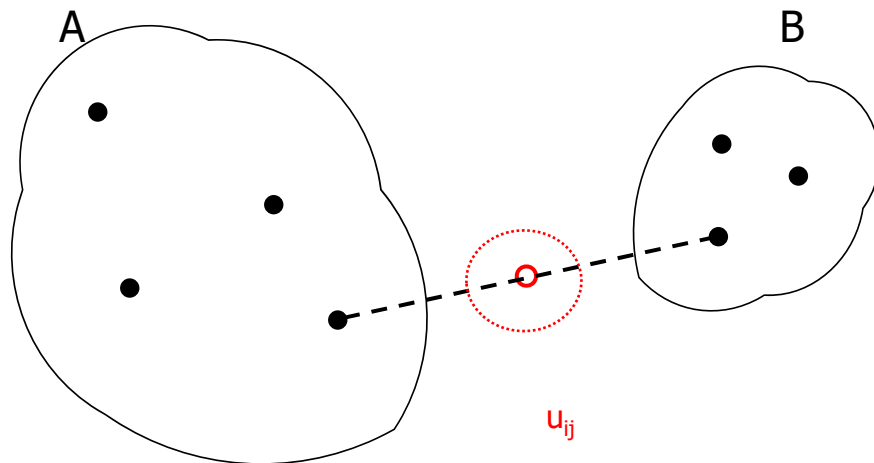


Newer Measures For Separation and Scatter

New indices defined by **the distances** (of data) and **the data distribution**.

- Ex: CDbw (Composite Density between and within clusters) (Halkidi, Vazirgiannis, 2002)

$$CDbw(c) = Intra_dens(c) \cdot Sep(c), c > 1$$



- representatives (prototypes)
- midpoint of the closest prototypes
- stdev: average standard deviation of clusters
- stdev_i: standard deviation of cluster i

$$Intra_dens(c) = \frac{1}{c} \sum_{i=1}^c \frac{1}{r} \sum_{j=1}^r \frac{density(\underline{v}_{ij})}{stdev}$$

$$Sep(c) = \frac{\sum_{\substack{i=1 \\ i \neq j}}^c \sum_{j=1}^c d(clos_rep_i, clos_rep_j)}{c^2 - c} \quad c > 1$$

$$density(\underline{v}_{ij}) = \sum_{l=1}^{n_i} f(x_l, \underline{v}_{ij})$$

$$f(x, \underline{v}_{ij}) = \begin{cases} 0, & \text{if } d(x, \underline{v}_{ij}) > stdev \\ 1, & \text{otherwise} \end{cases}$$

$$f(x, u_{ij}) = \begin{cases} 1, & \text{if } d(x, u_{ij}) < stdev \\ 0, & \text{otherwise} \end{cases}$$



Performance of CVI-s

- Performance of a CVI (whether it is effective measuring the clustering quality) depends on its construction, and on the complexity of the clusters
- Usually good judgment for simple structures; misleading index values for complicated structures – still much work to do

- Taşdemir & Merényi, 2011 evaluate several CVI-s including some more recent ones

Clustering Approaches

- “**Mode finding** (or bump hunting): find multiple convex regions [of the input space X] that contain modes of $\text{Pr}(X)$.
 - This can show if $\text{Pr}(X)$ can be expressed by a mixture of simpler density models each representing a distinct type of observations.
 - Find a smaller set of latent variables (the modes)
 - Can get difficult / intractable in higher dimensions
- **Combinatorial methods** find optimum partitioning wrt some goal function
 - Work directly on the observed data points (do not use probability models)
 - Each data point assigned to one cluster (many-to-one encoding)
 - Predefined # of clusters, K
 - BUT: for N data points and K clusters, the # of possible partitionings (cluster assignments) $S(N,K)$ quickly explodes

Ex: $N=10, K=4 \Rightarrow S(10,4) = 34,105$

Ex: $N=19, K=4 \Rightarrow S(19,4) \approx 10^{10}$!!

$$S(N, K) = \frac{1}{K!} \sum_{k=1}^K (-1)^{K-k} \binom{K}{k} k^N.$$



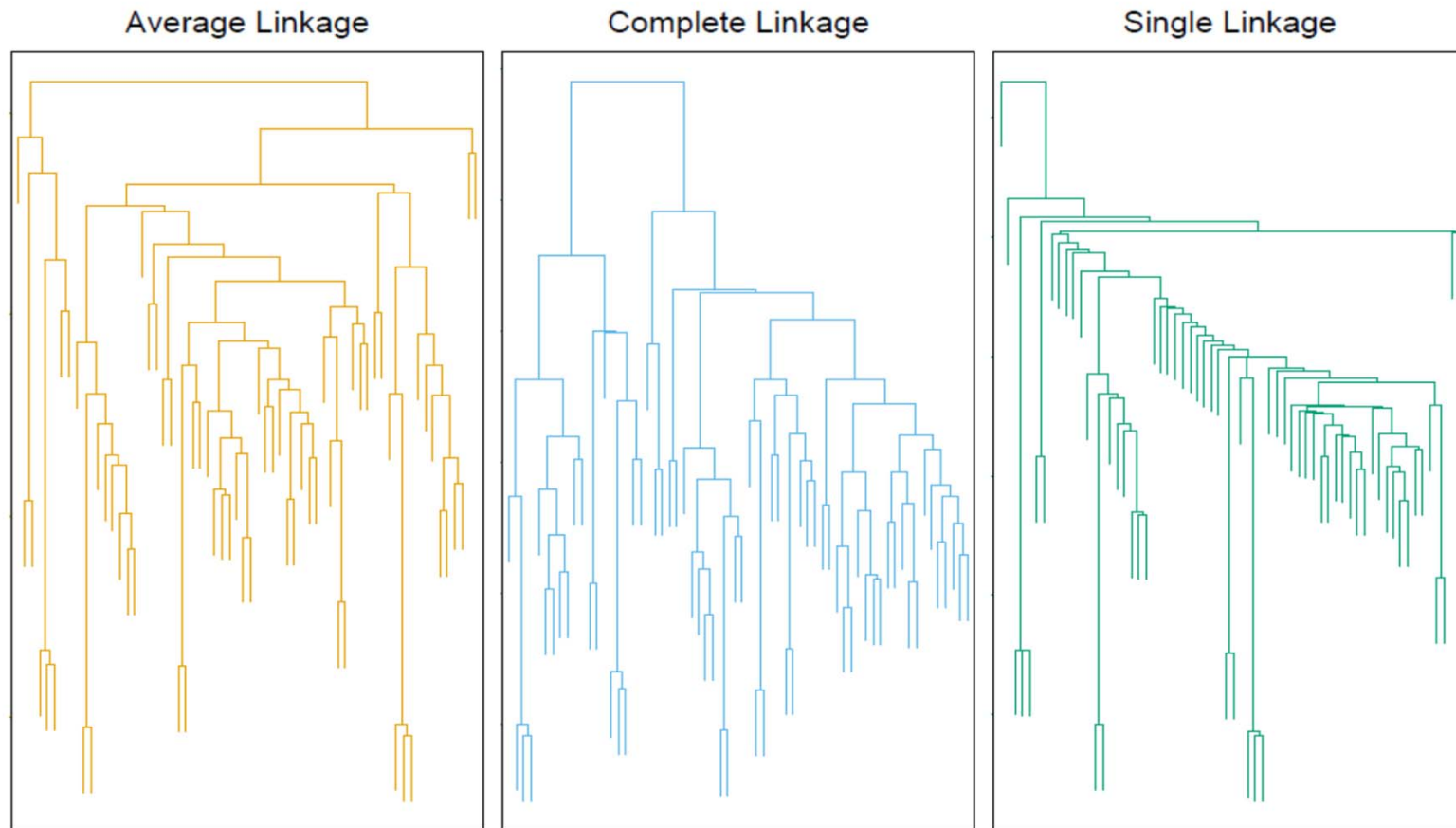
Hierarchical Clustering

Builds a binary tree where each node is a cluster; the children of a node are subclusters

- Work directly on the observed data points (do not use probability models); Assign each data point (n-dim sample) to one cluster
- The tree can be built by agglomerative (bottom-up) method, successively merging the two closest clusters
- or by divisive (top-down) method, successively splitting clusters by some quality criterion (e.g., a CVI)
- # of clusters, K , is NOT predefined, but obtained by cutting the resulting tree (dendrogram), by a quality criterion
- NEGATIVE: can be **computationally intense** (works with pair wise (cluster) similarities; or with CVIs involving the former)
- POSITIVE: Model-free, any similarity measure can be plugged in; Can capture irregular clusters, and a large number of clusters
- BEWARE: The choice of cluster similarity / distance or partitioning quality measure greatly influences the outcome



Outcomes of Clustering the Same Data With Different Similarity Measures



Dendrograms, showing the stages of the clustering. Each was built using a different cluster similarity metric, indicated at the top of the panels.

(Figure from Hastie et al., 2008)



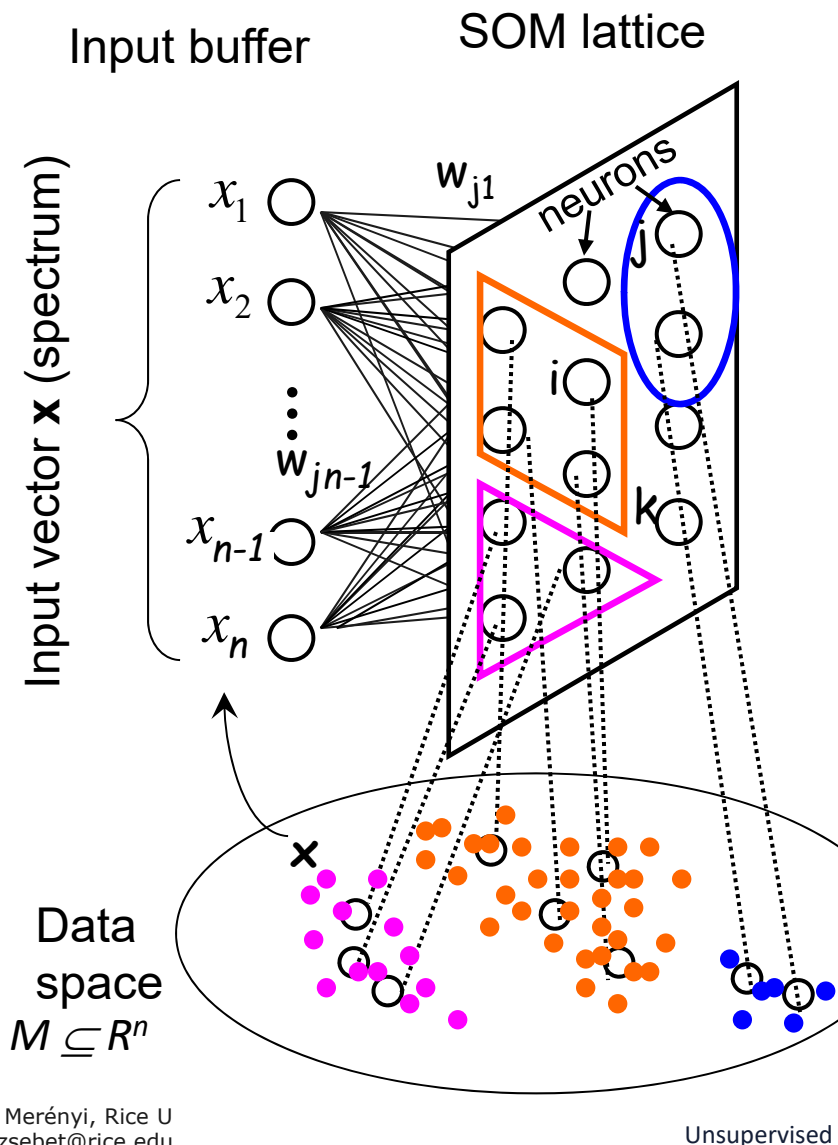
Prototype Based Clustering Approaches

Alleviate computational burden: compute distances to a smaller number of *prototypes* (not between all pairs of data points); this is VQ, coarse grained

- **K-means:** iteratively adjusts initial cluster centers (Linde, Buzo, Gray, 1980)
- Computationally inexpensive
- *K is predefined*; optimal # of clusters must be determined by charting a partitioning quality measure (such as a CVI) as a function of K
 - Gap measure (Tibshirani et al, 2001: average within-cluster scatter compared to same of uniform distribution; the ideal K is where the “gap” is maximum. The gap ignores the between-clusters distances!
- Model-free but *favors spherical clusters* (each prototype is center of one cluster – implicitly assumes spherical clusters)
- Very sensitive to the initial choice of cluster centers
- Experience: works well for simple data; but not for high-D, complex data

Learn the data structure with Self-Organizing Maps

Machine learning analog of biological neural maps in the brain



Two simultaneous actions:

- **Adaptive** Vector Quantization (VQ): puts the prototypes in the “right” locations => allows summarization of N data vectors by **$O(\sqrt{N})$ prototypes**; while encoding salient properties
- Ordering of the prototypes on the SOM grid according to similarities; *only SOMs do this.*

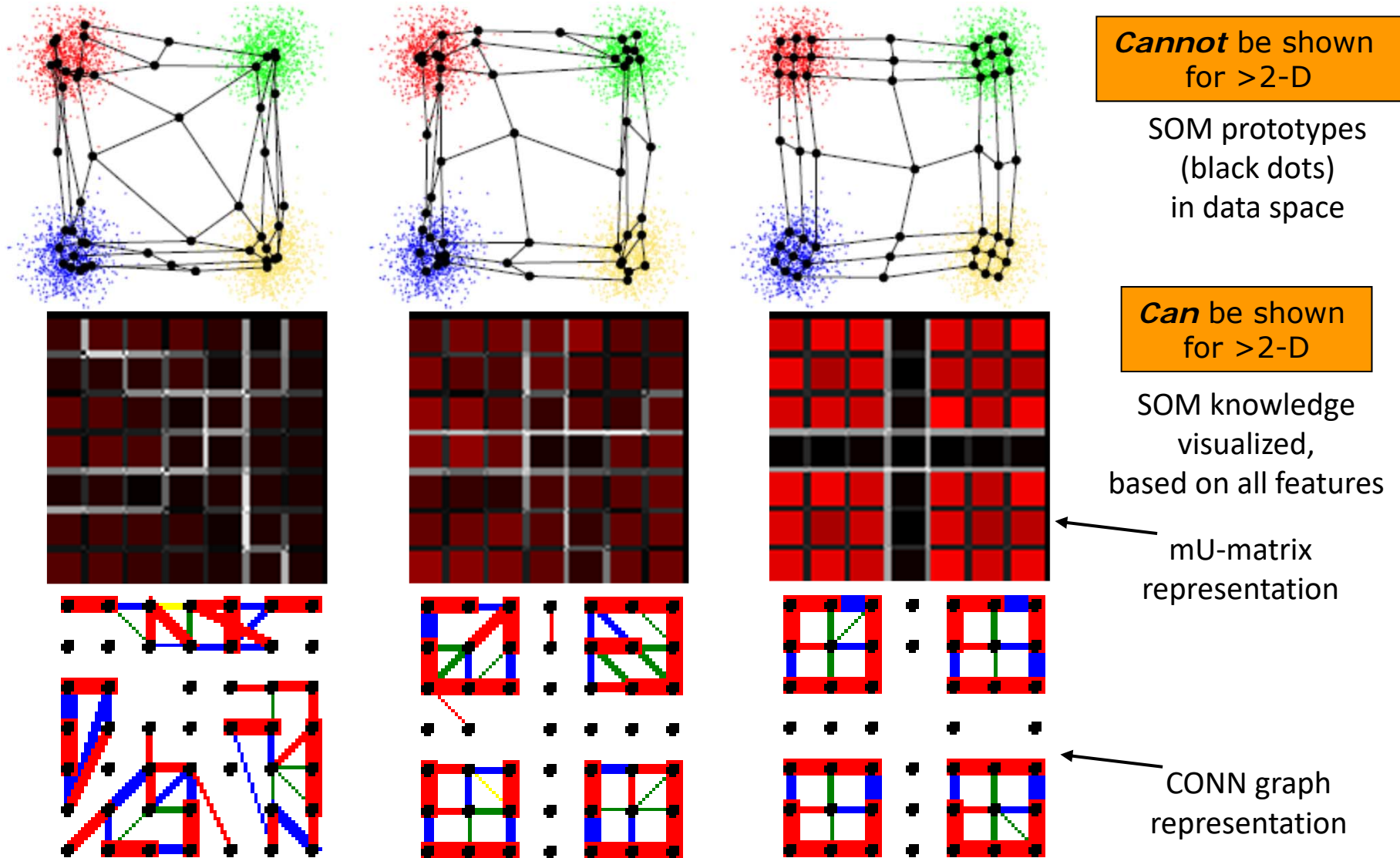
I.e., SOM learns the structure (the distribution) AND expresses the topology (similarity relations) on a low-dimensional lattice.

Finding the prototype groups: post-processing – segmentation of the SOM based on representations of the SOM’s knowledge



Toy example: unsupervised SOM learning of 4 Gaussian clusters

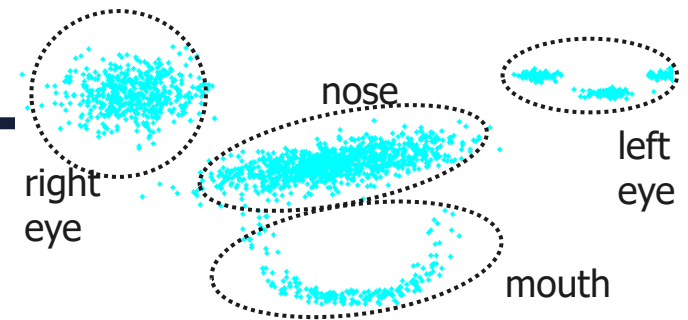
Evolution of prototypes, and visualization



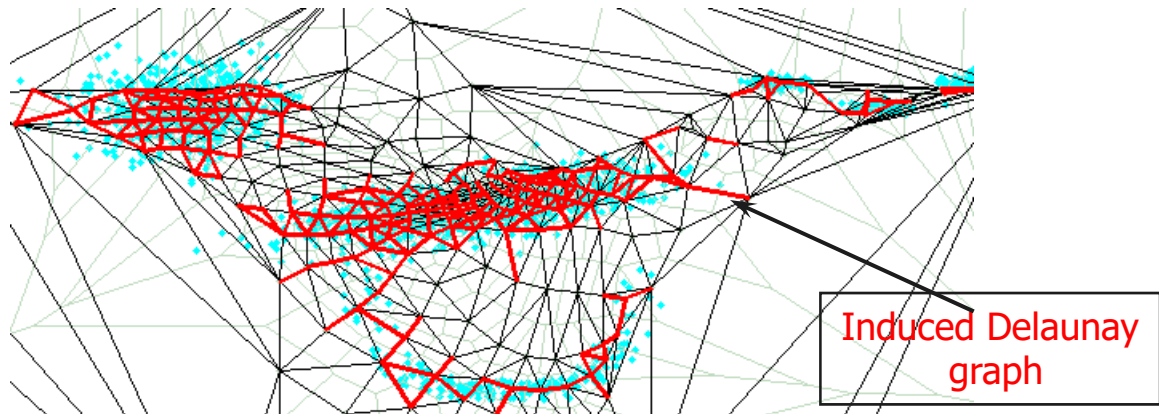
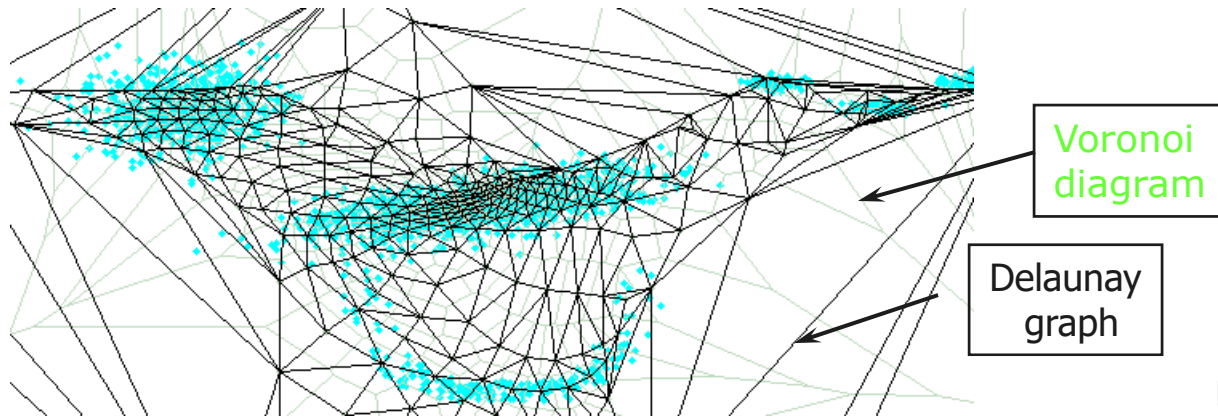
Graph representation of SOM knowledge: Induced Delaunay graph

Well-learned SOM prototypes (black vertices), nicely follow the data distribution.

Placement of prototypes is crucial! (Assume correct learning.)



2-D "Clown" data
(Data: Vesanto and Alhoniemi, 2000)



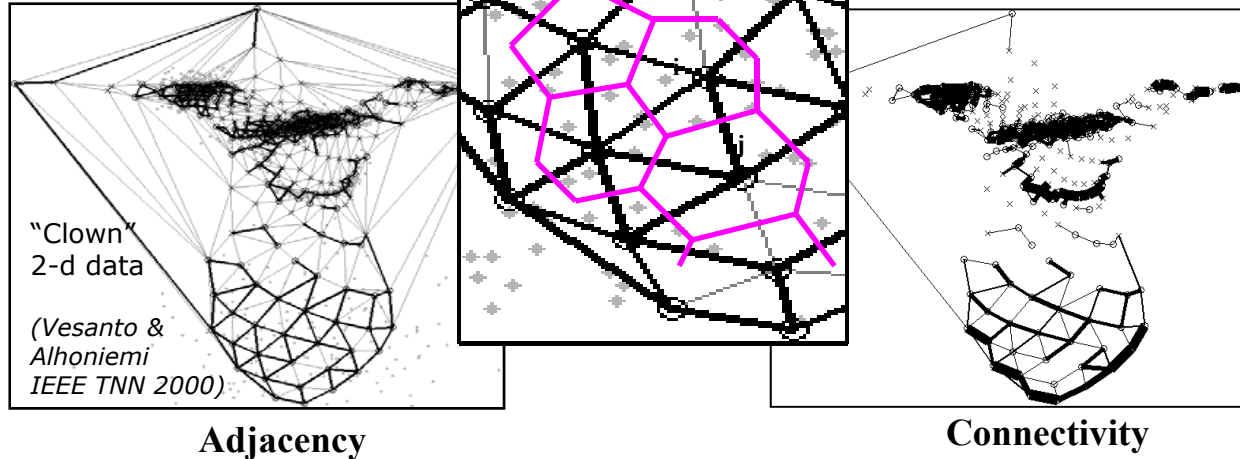
(Figures from Taşdemir and Merényi, 2009)

- Martinetz and Schulten, 1994:
- The induced Delaunay graph perfectly represents topology - but how to get it in high-D space?
 - Competitive Hebbian learning (neural maps) produces the induced Delaunay graph (with one mild condition)
- To get it: Connect two prototypes if they are closest and 2nd closest match for a data vector

Connectivity (CONN) similarity measure and graph

(Taşdemir & Merényi, IEEE TNN 2009)

Induced
Delaunay
graph
- binary

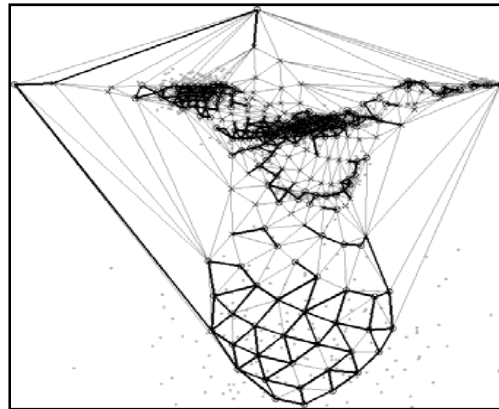


CONN graph:
Weighted
induced
Delaunay
graph

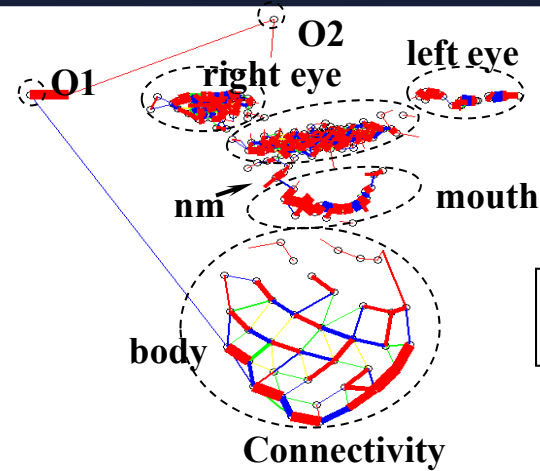


Connectivity (CONN) similarity measure and graph

(Taşdemir & Merényi, IEEE TNN 2009)

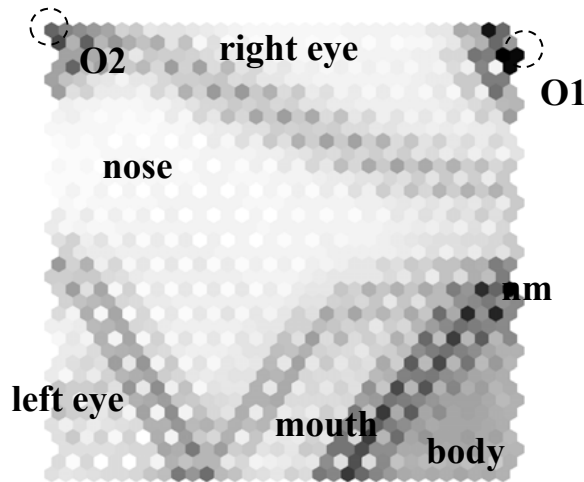


Adjacency



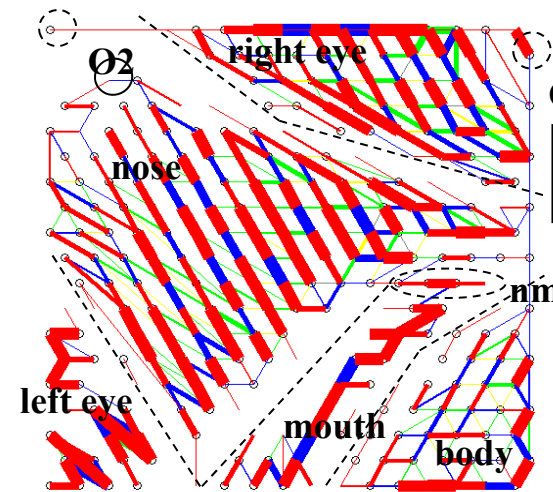
Connectivity

Cannot be shown for data dim > 2



U-matrix ($\sum \|w_i - w_j\|$)
overlain the SOM grid

Figure adapted from Vesanto & Alhoniemi IEEE TNN 2000



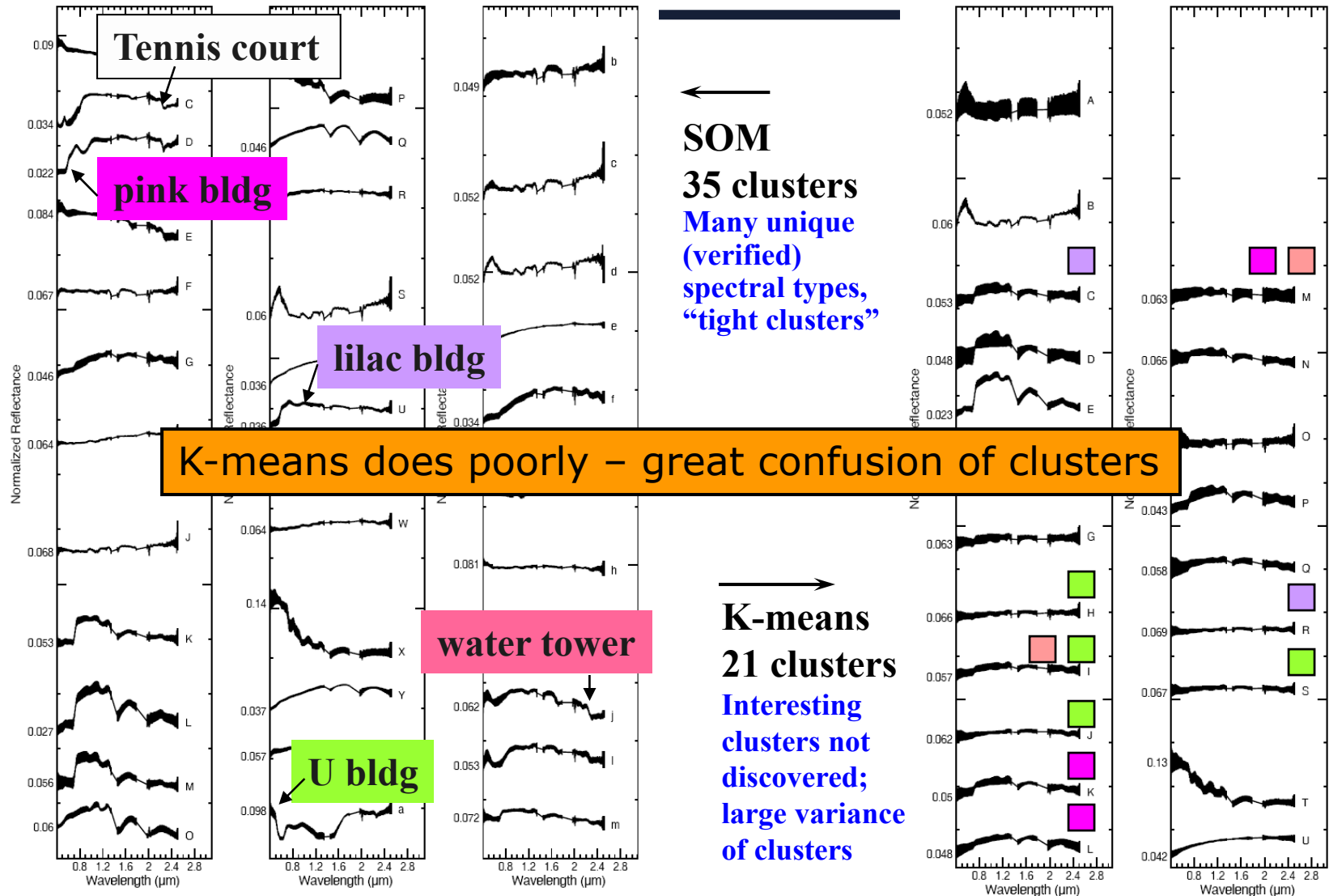
Can be shown For data dim > 2

Bonus: CONN shows topology violations

CONNectivity matrix draped over SOM grid: The SOM / CONN portrait of the Clown

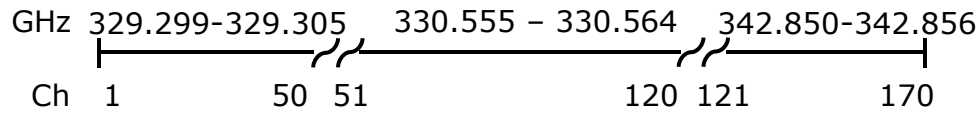
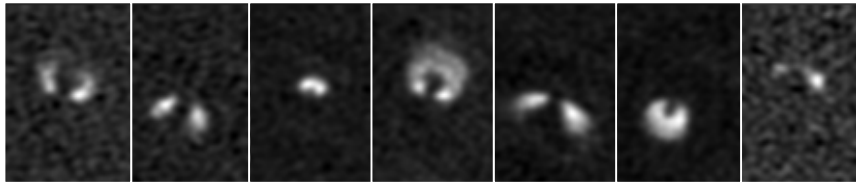
SOM vs K-means: Spectral Statistics of Clusters

Data: Ocean City, 200-band Hyperspectral Image of Urban Area

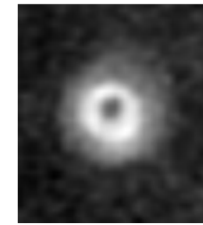


Example: ALMA hyperspectral image – spectral variations

Image planes from ALMA Band 7, protoplanetary disk HD 142527

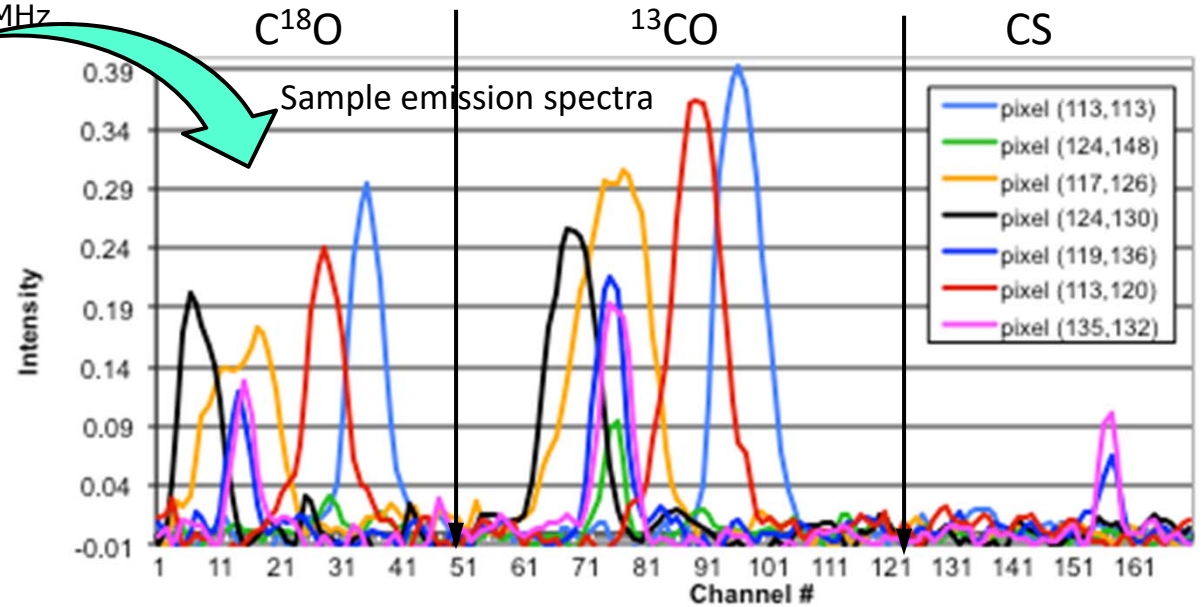
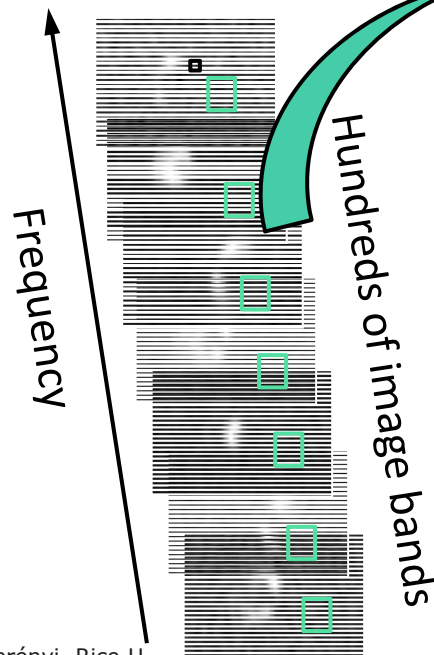


170 channels: C¹⁸O, ¹³CO, CS lines stacked
Spectral resolution: 0.122 MHz



Continuum image

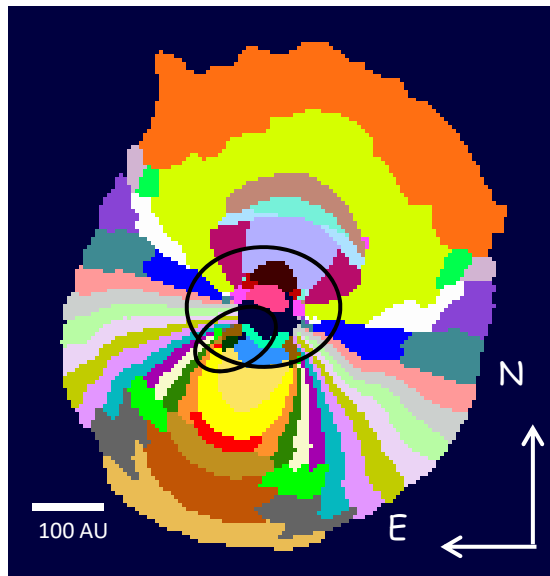
Cluster the spectral signatures to map regions of distinct kinematic and compositional behavior.



ALMA spectra from combined C¹⁸O, ¹³CO, CS lines, showing differences in composition, Doppler shift, temperature

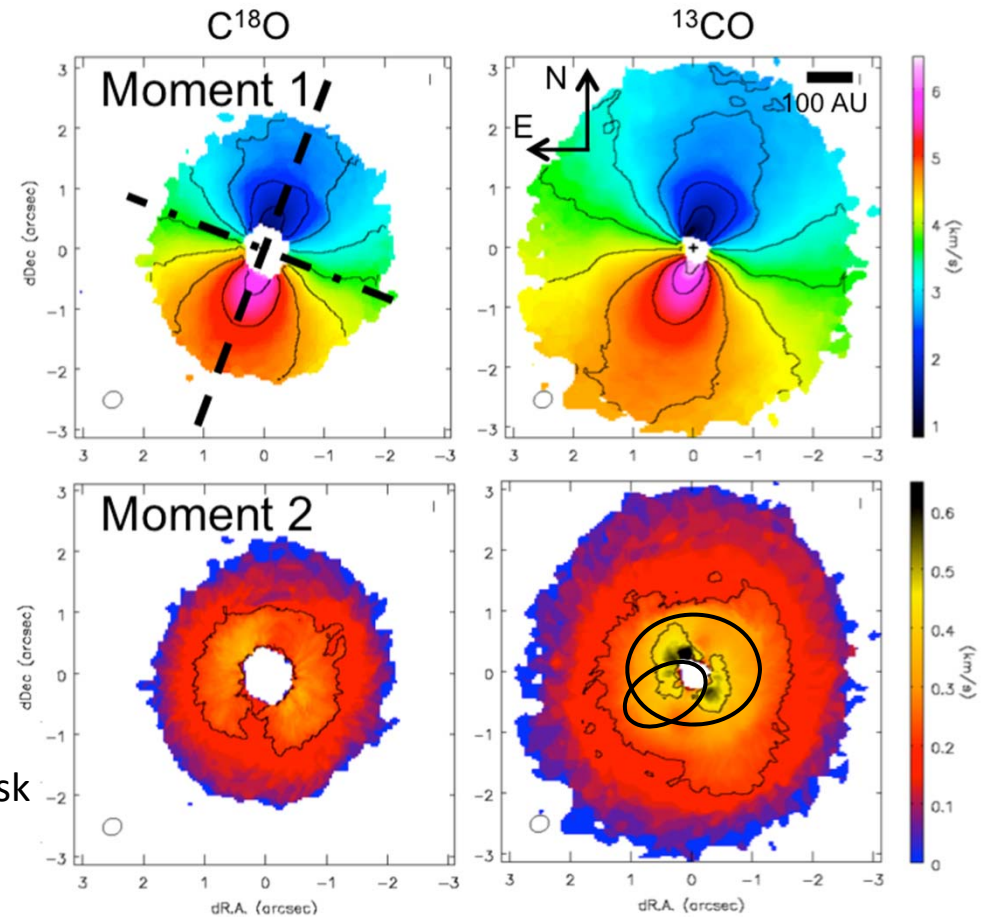
NeuroScope structure discovery from ALMA data HD 142527 protoplanetary disk (data: Isella 2015)

NeuroScope cluster map from stacked $C^{18}O$, ^{13}CO lines, 100 + 100 channels as input feature vectors



The emerging structure of the protoplanetary disk based on all channels of two molecular tracers, visualized in one 2-D view

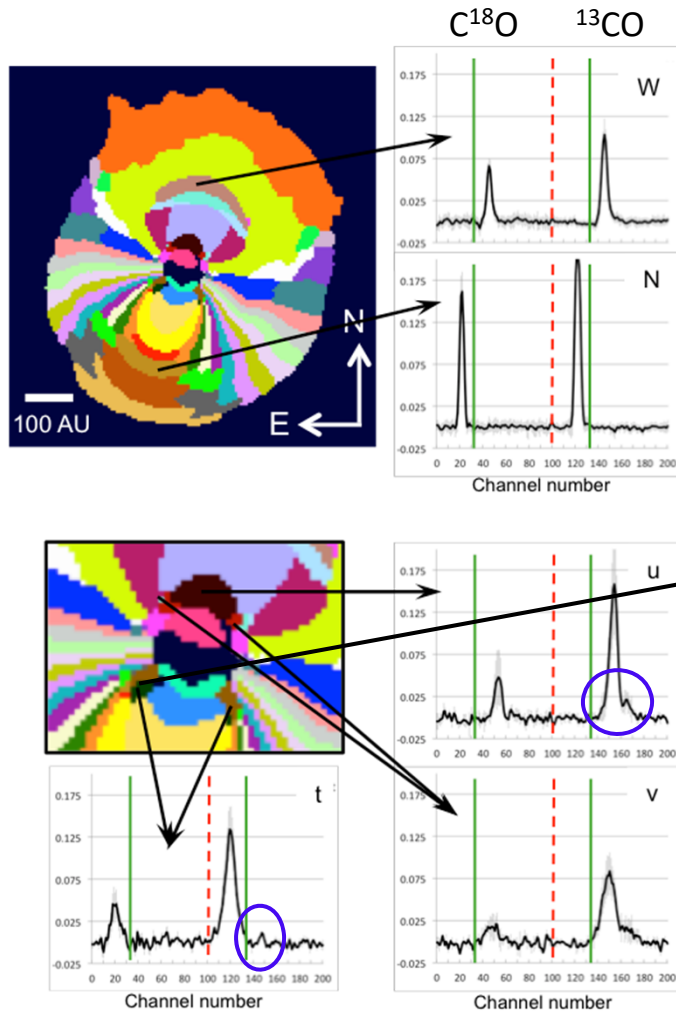
Coloring of clusters is arbitrary, not a heat map!



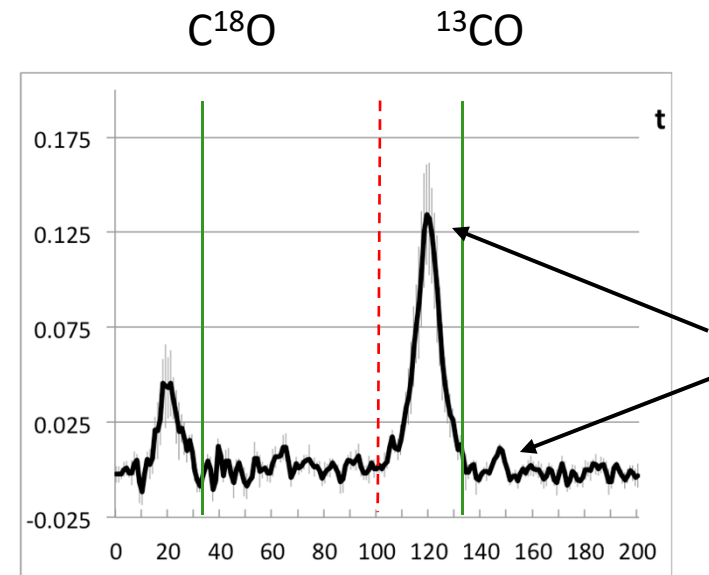
(Merényi, Taylor, Isella, Proc. IAU 325, 2016)

Clusters found in HD142527

Data: ALMA image cube of HD142527 (Isella, 2015)



Mean cluster signatures alert to interesting areas.



Two distinct peaks, shifted opposite from rest frequency. Two gas components moving in different directions.

More discovery within one molecular line

(Merényi, Taylor, Isella, Proc. IAU 325, 2016)

More discovery from the combination of lines



E. Merényi, Rice U
 erzsebet@rice.edu

Unsupervised Learning, DMML 2018

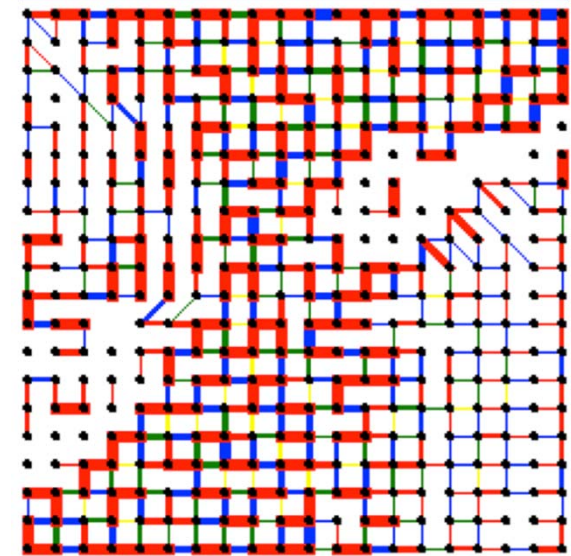
Our Approach To Structure Discovery

Step 1: Learn the data manifold with SOMs - easy, reliable, little tuning needed, automatic, unsupervised.

- Use all input features – keep the discovery potential
- No assumption except lose upper limit of potential clusters (to allocate enough SOM prototypes)
- Use Conscience SOM (CSOM) for maximum entropy learning (best matching of the data distribution)

Step 2: cluster the SOM prototypes – can be hard

- Need good knowledge representation, sensitive similarity measure, like the CONN graph, and visualization.
- Interactive cluster extraction (based on recipe) is best so far. DOES NOT SCALE.
- We look to modern graph-segmentation methods ...



CSOM / CONN portrait of
the ALMA cube of HD142527 ₃₁

Clustering By Graph-Segmentation

Community Finding

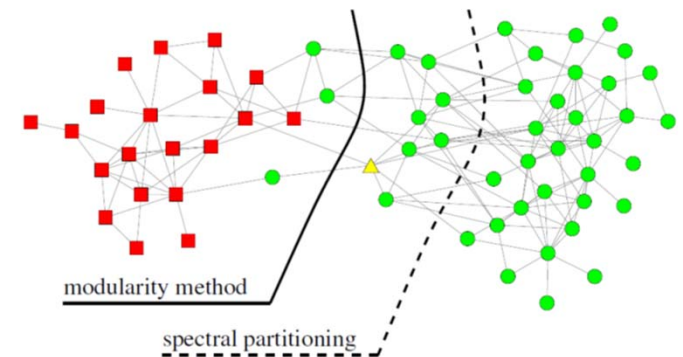
- Works with a pair wise adjacency (proximity) matrix \mathbf{A} of the data as edges in a graph where nodes represent data points
- Cut the graph “optimally”. Ex:
 - **Spectral partitioning** – cut the graph Laplacian matrix, \mathbf{L} , to minimize the cut size, subject to equal-size partitions (!)

Cut size = # edges across different clusters; can be expressed as a weighted sum of eigenvalues of \mathbf{L} . Optimization assigns large weights to terms with small(est) eigenvalue under norm. constraint.

- Cut by (2nd, approximate) “**leading eigenvector**” of the modularity matrix \mathbf{B} (devisive); optimizes a modularity function based on \mathbf{B}

Works better than spectral partitioning.

- **Fast & Greedy** – agglomerative, also optimizes the same modularity function



(From Newman, 2006)

$$A_{ij} = \begin{cases} 1 & \text{If vertices } i, j \text{ connected} \\ 0 & \end{cases}$$

$$L_{ij} = \begin{cases} \sum_j A_{ij}, & i = j \text{ degree of vertex } i \\ j & \\ -1 & i \neq j \text{ and } i, j \text{ connected} \\ 0 & \end{cases}$$

$$B_{ij} = A_{ij} - P_{ij}.$$

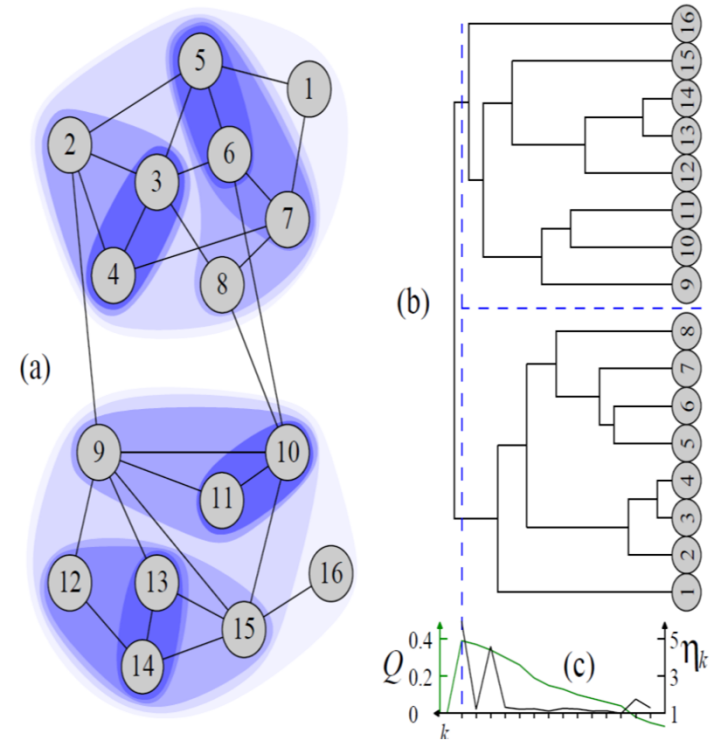
P_{ij} : Probability of edge (i,j) in a “null-model”



Clustering By Graph-Segmentation

Community Finding

- Cut the graph “optimally”. Ex: (cont’d)
 - **Walktrap** – uses random walk to derive a similarity measure based on the distribution of destination states of vertices i and j , after t steps. Then uses this measure in agglomerative hierarchical clustering.
- Does not use the modularity criterion for tree building, but uses to evaluate afterwards
- **Infomap** – also based on random walk, but forms an entropy-based cost function from the within- and between-clusters transitions.
- Many more ... review in Fortunato (2010)
- Available in the `igraph` package, 0 to 2 parameters – good for automation
- BUT: extremely resource hungry
 - N data points $\Rightarrow O(N^2)$ edges
 - 1000 x 1000 px image $\Rightarrow 10^{12}$ edges!!!



(From Pons and Latapy, 2006)

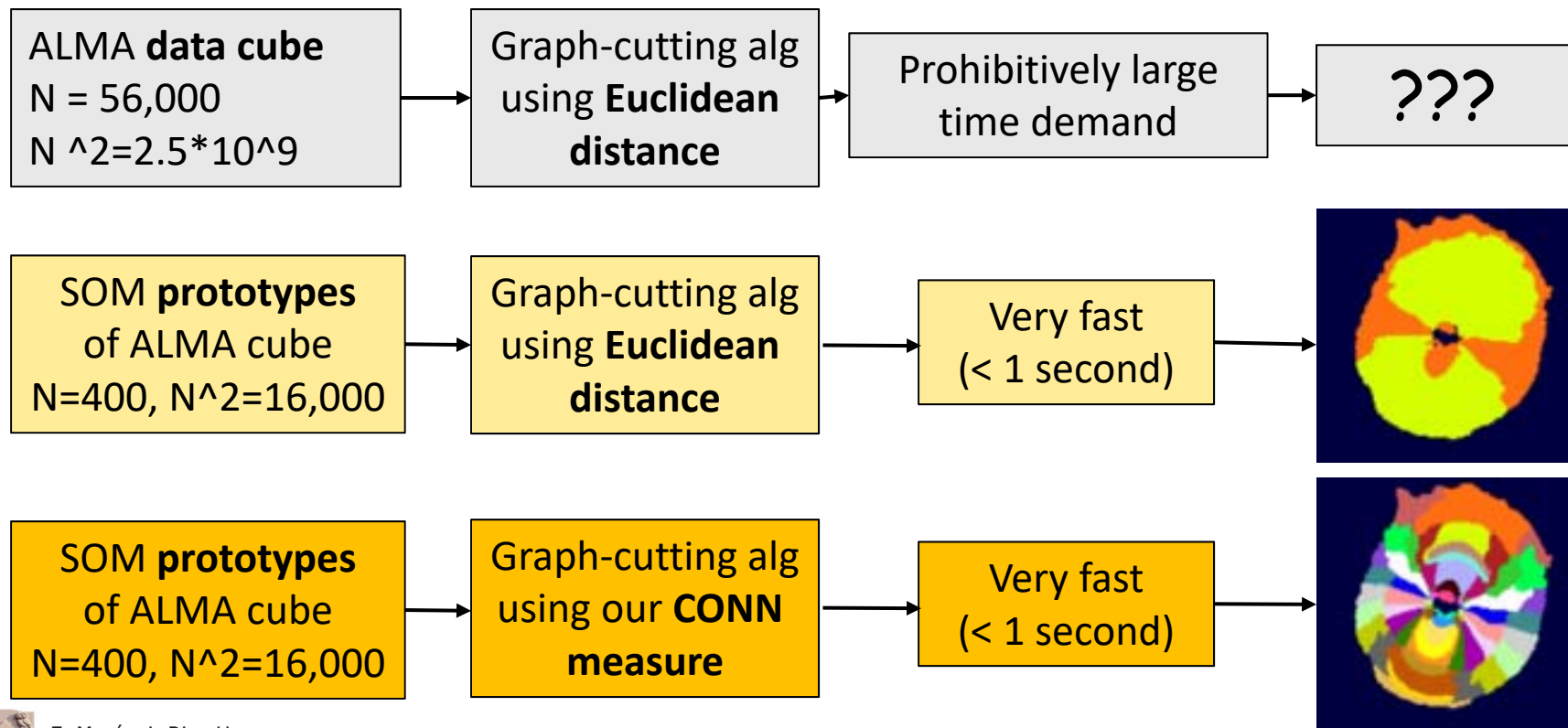
Notice that the peaks of the modularity (goal) function Q indicate that relevant partitionings may exist on multiple scales.



Automation For Segmentation of the SOM

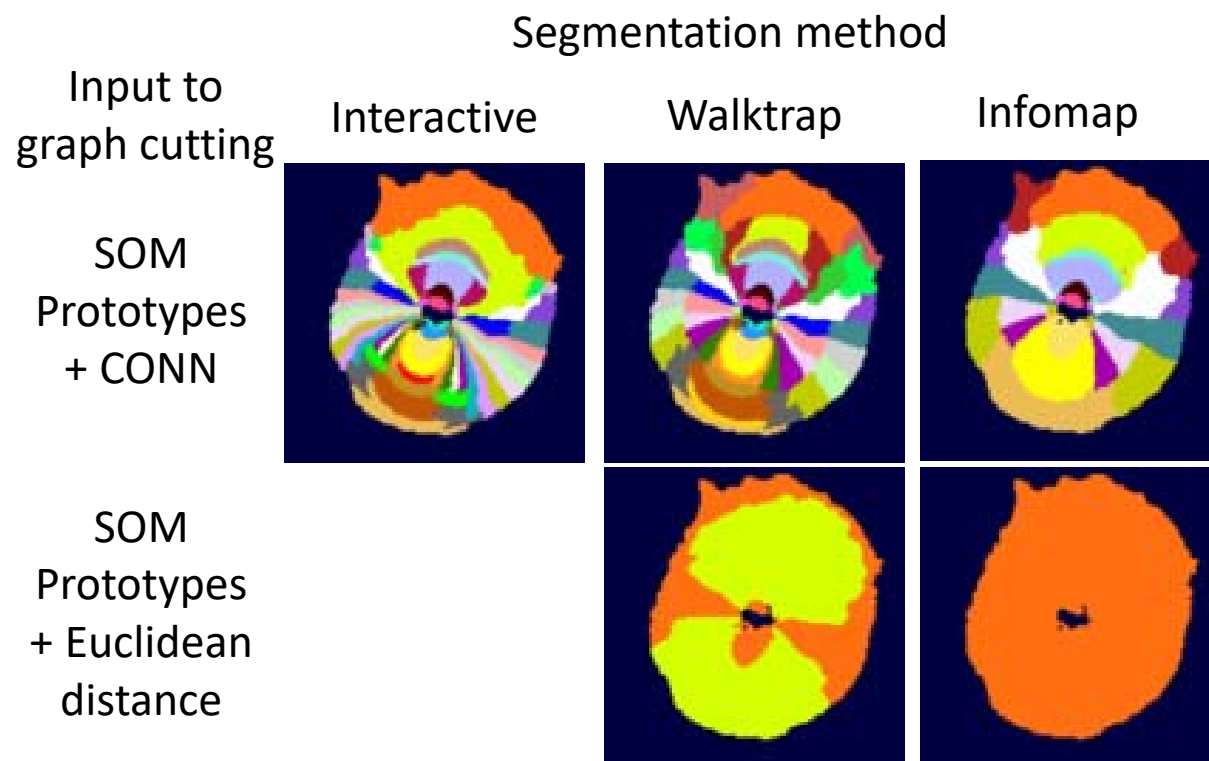
Graph-segmentation informed by SOM and CONN

- ☺ Graph-cutting methods: automatic, only 1 or 2 parameters, some have none *
- ☹ Can't deal with many data points. N vectors $\Rightarrow N^2$ edges. For this small ALMA image (56,000 vectors), over 10^9 edges !!!
- ☺ ☺ Use the intelligently summarized data (SOM prototypes) as input
- ☺ ☺ ☺ Plus CONN similarity measure *(Merényi, Taylor, Isella, Proc. IAU 325, 2016)*



Interactive vs automated results

- Walktrap (Pons & Latapy, 2005) and Infomap (Rosvall & Bergstrom) – two best results with default setting (`igraph` package), 1 or 2 parameters.
- Details don't quite match, but differences reasonable. Graph-segmentation of SOM + CONN finds relevant structure, and FAST.



- Next: explore non-default parameters, for improvement
- Interpret differences

Mass-processing perspectives for pipelines

(numbers for the ALMA example)

- Do SOM learning in parallel hardware : < 5 sec
 - Dedicated mid-level FPGA implementation, could be much faster for more \$\$
- Cluster the SOM prototypes automatically with SOM+CONN input to graph-segmentation algorithms: < 1 sec
- Scales linearly with # of samples, and (within large range) with # of feature dimensions

Other benefits:

- Applicable to disparate data combined from different spectral windows or instruments
- Applicable to chaotic sources (GMCs, galaxy clusters, etc.)

Conclusions

- **Rich data** (e.g., spectral resolution for ALMA) **offer a magnifying lens for the underlying physical processes** (kinematics of atomic and molecular gas and the distribution of solid particles in the ALMA example).
- **Capabilities to exploit the richness and subtleties of features** (details of the feature vectors) **can enlarge the discovery space.**
- Combining proper methods and metrics brings magnitudes of algorithmic speed-up, support large-scale, automated processing.
- **For DM search**, stacked measurements (images) taken at different frequencies, and/or other (possibly disparate) data can be input to ML, for increased discovery potential.



References

- T. Hastie et al. The elements of statistical learning. Springer, 2008
- A. Hyvarinen, Independent Component Analysis, 2001
- M. Van Hulle, Faithful Representations and Topographic Maps, Wiley & Sons, 2001
- D. Wunsch and Xu, Survey Of Clustering Algorithms, IEEE TNN 16:3, pp 645-678, 2005
- Lin, Buzo, and Gray, An Algorithm for Vector Quantizer Design, IEEE Trans. Com, Com-28:1pp 84-95, 1980
- R. Tibshirani et al., Estimating the number of clusters in a dataset via the gap statistic, Journal of the Royal Statistical Society, Series B. 32(2): 411–423, 2001.
- Bezdek & Pal, Some New Indexes of Cluster Validity. IEEE Tran. Sys. Man and Cyb. Part B, 28:3 1998
- [Taşdemir, K., and Merényi, E. \(2011\) A Validity Index for Prototype Based Clustering of Data Sets with Complex Structures. IEEE Trans. Sys. Man and Cyb., Part B. 02/2011; Vol. 41, No. 4, pp 1039 - 1053. DOI: 10.1109/TSMCB.2010.2104319](#)
- Merényi, E., Taşdemir, K., Zhang, L. (2009) [Learning highly structured manifolds: harnessing the power of SOMs](#). Chapter in “Similarity based clustering”, Lecture Notes in Computer Science (Eds. M. Biehl, B. Hammer, M. Verleysen, T. Villmann), Springer-Verlag. LNAI 5400, pp. 138 – 168.
- Taşdemir, K, and Merényi, E. (2009) [Exploiting the Data Topology in Visualizing and Clustering of Self-Organizing Maps](#). IEEE Trans. Neural Networks 20(4) pp 549 – 562.
- Merényi, E., Taylor, J. and Isella, A. (2016), Deep data: discovery and visualization. Application to hyperspectral ALMA imagery. *Proceedings of the International Astronomical Union*, 12(S325), 281-290. doi:10.1017/S1743921317000175



References

- M.E. J. Newman, Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev.* 2006
- P. Pons and M. Latapy, Computing Communities in Large Networks Using Random Walks. *J. Graph Algorithms and Applications*, 10:2 pp 191-218. 2006
- Rosvall, M. and Bergstrom, C. (2008) Maps of random walks on complex networks reveal community structure. *Proc. National Academy of Science* 105, pp 1118-1123, Jan 2008.
- S. Fortunato, Community detection in graphs. arXiv, 2010. (103 pages)