Cluster Analysis

Group: #14

Instructor: Prof. Anita Wasilewska

Presenter: Xuanyu Yao, Yinlong Su, Sen Li

Overview

- Cluster Analysis Review
- Density-Based Clustering Methods
- Model-Based Clustering Methods
- Clustering High-Dimensional Data
- Q&A

Citations

- Wasilewska, Anita. (2016). “Introduction to Learning”. The State University of New York at Stony Brook. CSE 537 Spring 2016 Lecture Slides http://www3.cs.stonybrook.edu/~cse634/16L7learningintrod.pdf
- McCallum, A.; Nigam, K.; and Ungar L.H. (2000) "Efficient Clustering of High Dimensional Data Sets with Application to Reference Matching", Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 169-178 doi:10.1145/347090.347123
- A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, Institute for Computer Science, University of Munich, 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)
- Mehotra, K., Mohan, C. K., & Ranka, S. (1997). Elements of Artificial Neural Networks. MIT Press pp. 187-202
- Data Mining Lecture Slides, Chapter 7 Cluster Analysis, Professor Jianyong Wang, Tsinghua University

What is Cluster Analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- **Unsupervised learning**: no predefined classes
- Typical applications
  - As a *stand-alone tool* to get insight into data distribution
  - As a *preprocessing step* for other algorithms

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters with
  - High intra-class similarity
  - Low inter-class similarity
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation

Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality

Density-based Clustering

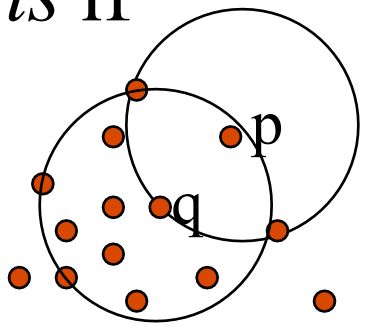
Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - DENCLUE: Hinneburg & D. Keim (KDD'98)

A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise
Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu
Institute for Computer Science, University of Munich
2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)

Density-Based Clustering: Basic Concepts

- Two parameters:
  - *Eps*: Maximum radius of the neighborhood
  - *MinPts*: Minimum number of points in an Eps-neighborhood of that point
- $N_{Eps}(q)$: $\{p \text{ belongs to } D \mid dist(p,q) <= Eps\}$
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to $N_{Eps}(q)$
  - Core point condition:
  - $|N_{Eps}(q)| >= MinPts$

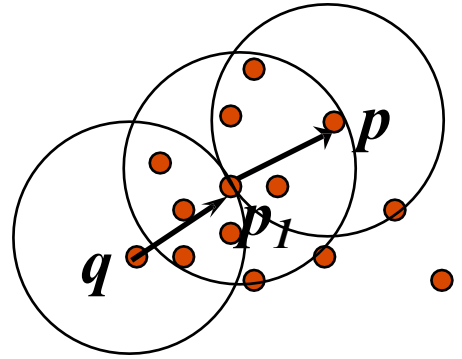


MinPts = 5

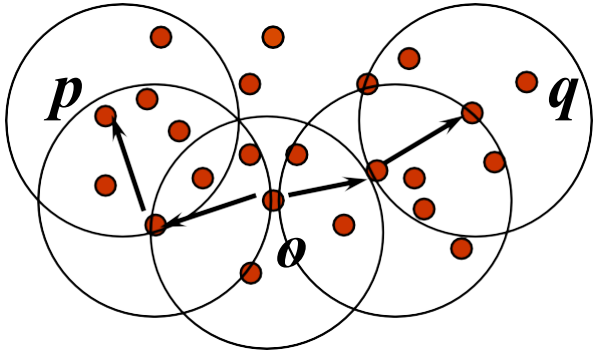
Eps = 1 cm

Density-Reachable and Density-Connected

- Density-reachable:
  - A point p is density-reachable from a point q w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i

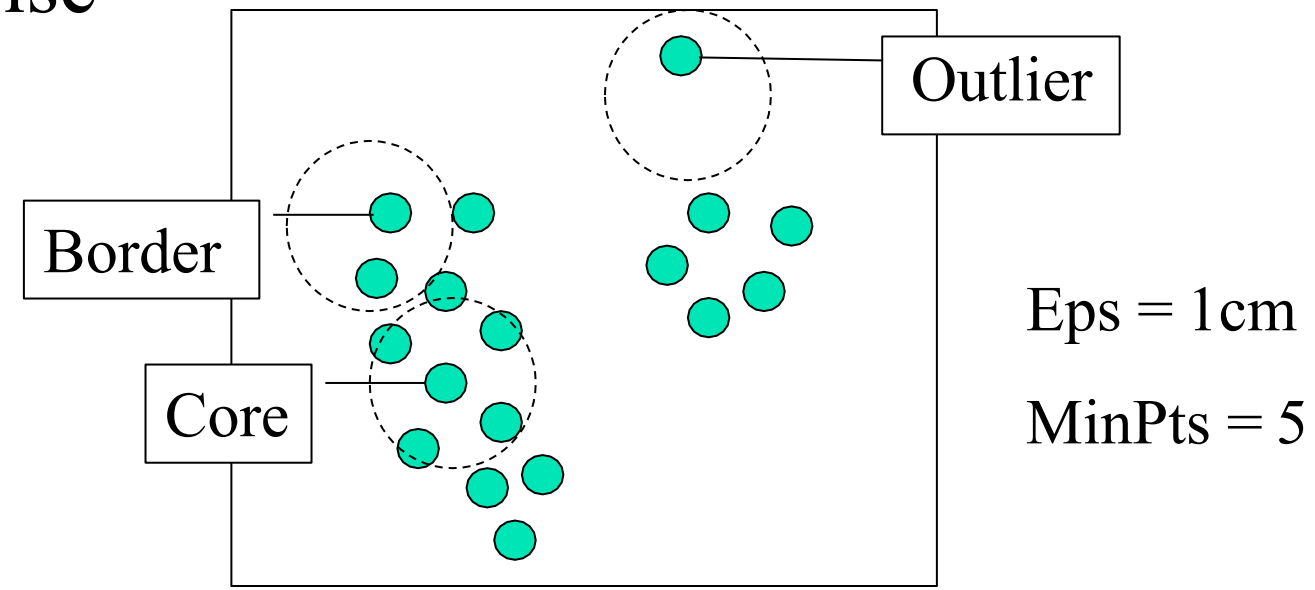


- Density-connected
  - A point p is density-connected to a point q w.r.t. *Eps*, *MinPts* if there is a point o such that both p and q are density-reachable from o w.r.t. *Eps* and *MinPts*



DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: The Algorithm

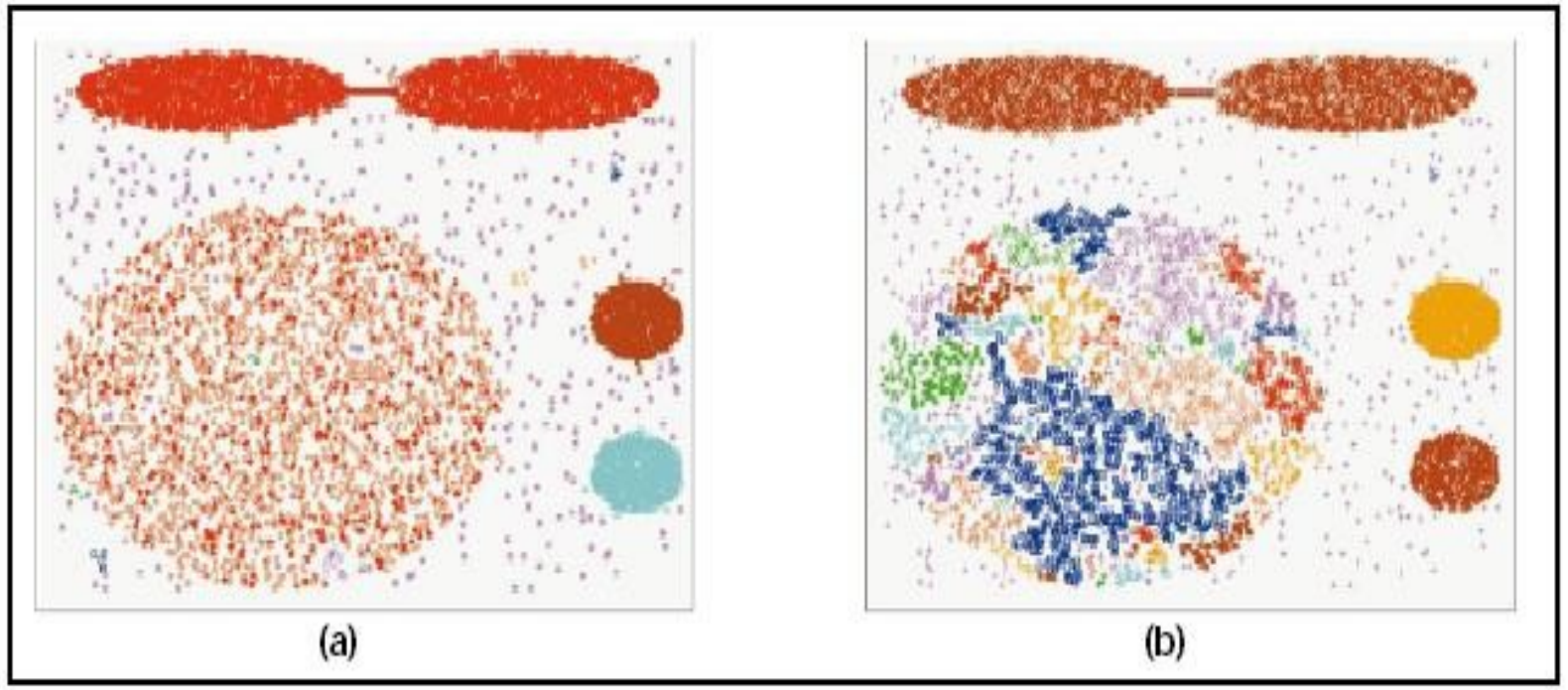
- Arbitrarily select a point *p*
- Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*.
- If *p* is a core point, a cluster is formed.
- If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

DBSCAN: Advantages and Disadvantages

- No need to specify the number of clusters
- No bias towards/against larger clusters
- Can learn arbitrary patterns
- Need to tune parameter Eps and MinPts manually

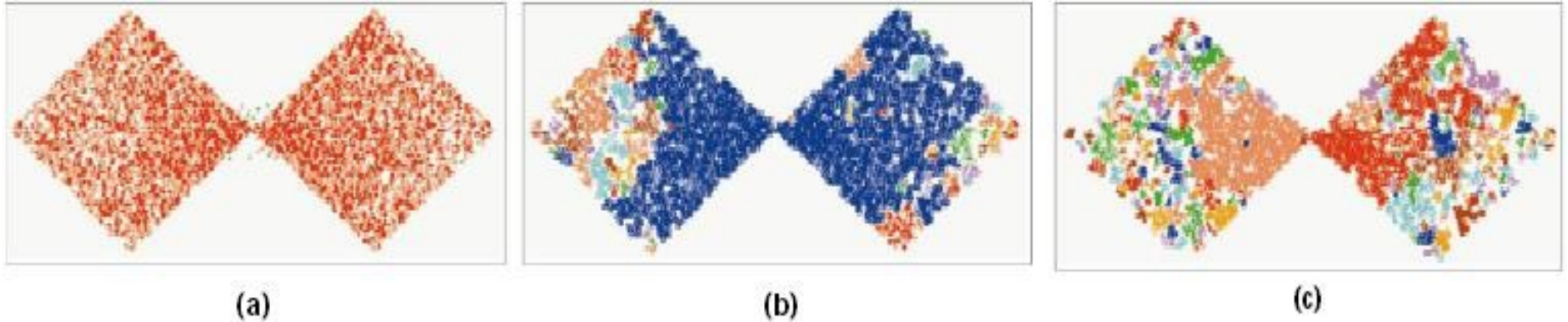
DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

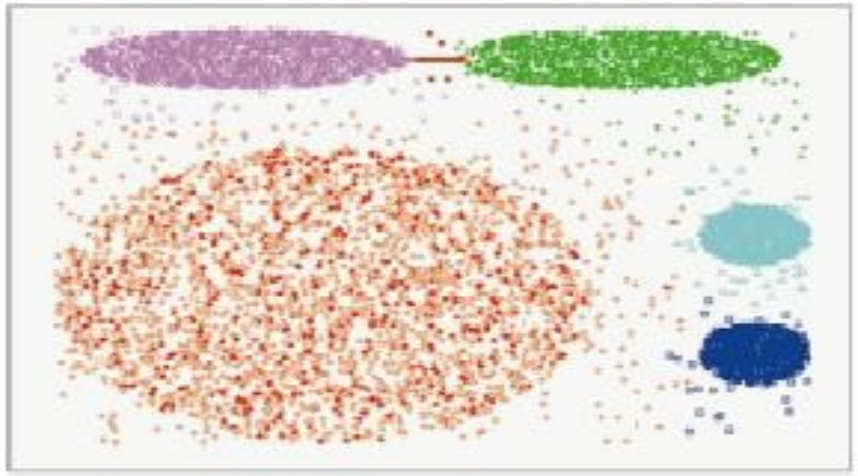


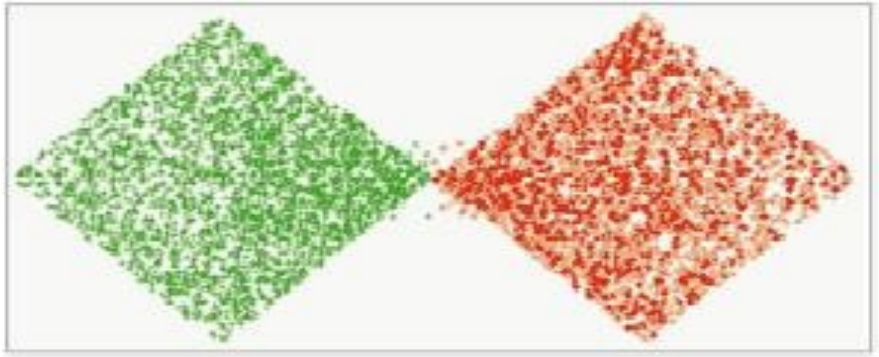
A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise
Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu
Institute for Computer Science, University of Munich
2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)

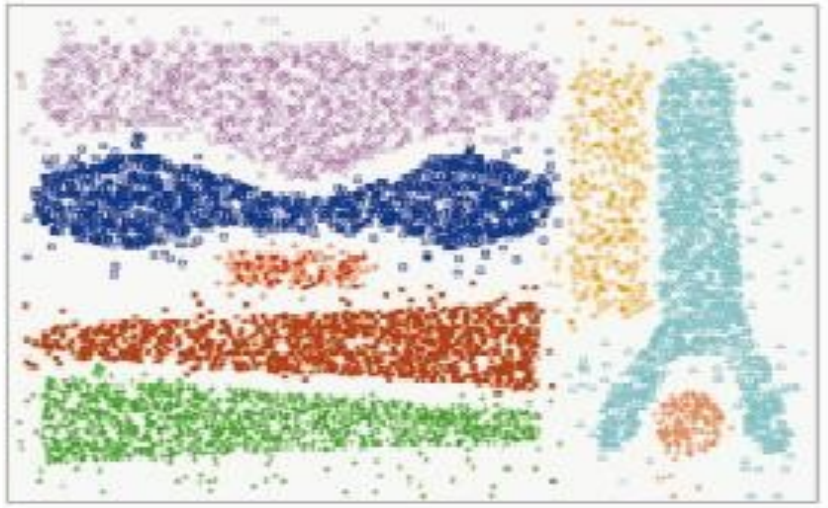
Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



DBSCAN: Sensitive to Parameters (cont.)







A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu

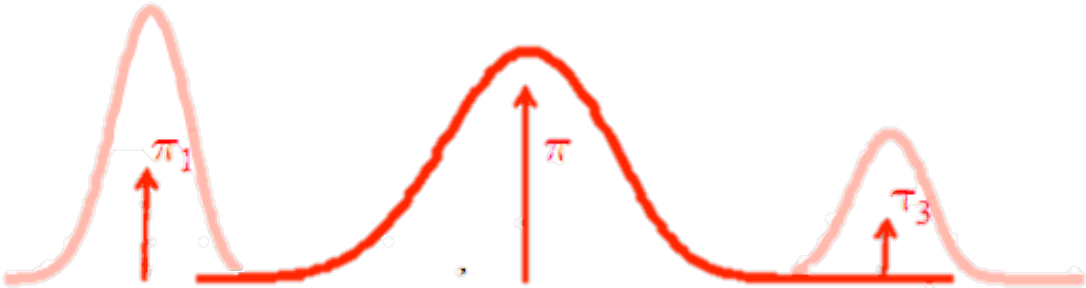
Institute for Computer Science, University of Munich

2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)

Model-based clustering

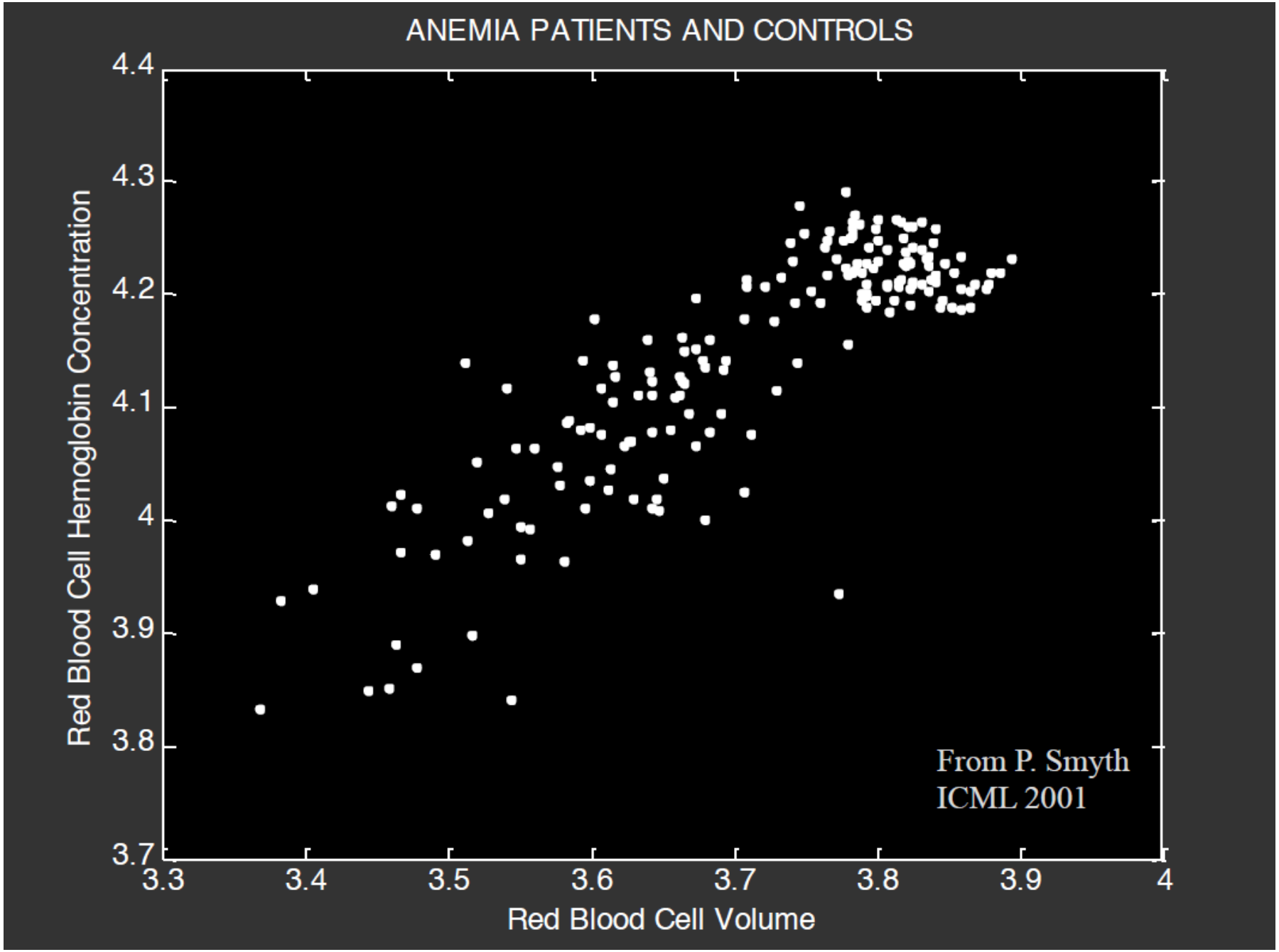
Mixtures of Gaussians

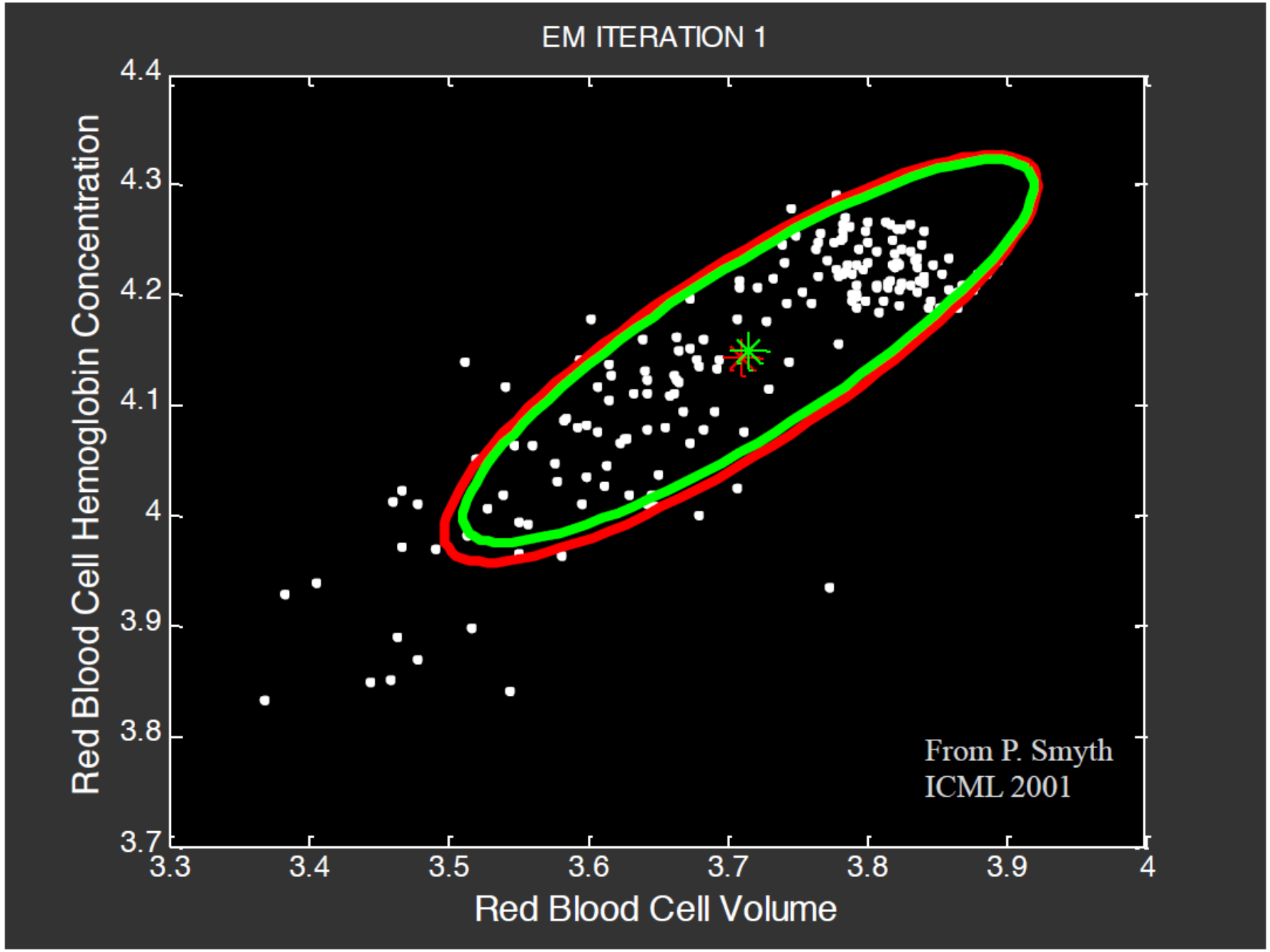
- Gaussian mixture models
  - $p(x)=\sum c\uparrow \pi\downarrow c \mathcal{N}(x;\mu\downarrow c,\sigma\downarrow c)$
    - Mean $\mu\downarrow c$, variance $\sigma\downarrow c$, "size" $\pi\downarrow c$

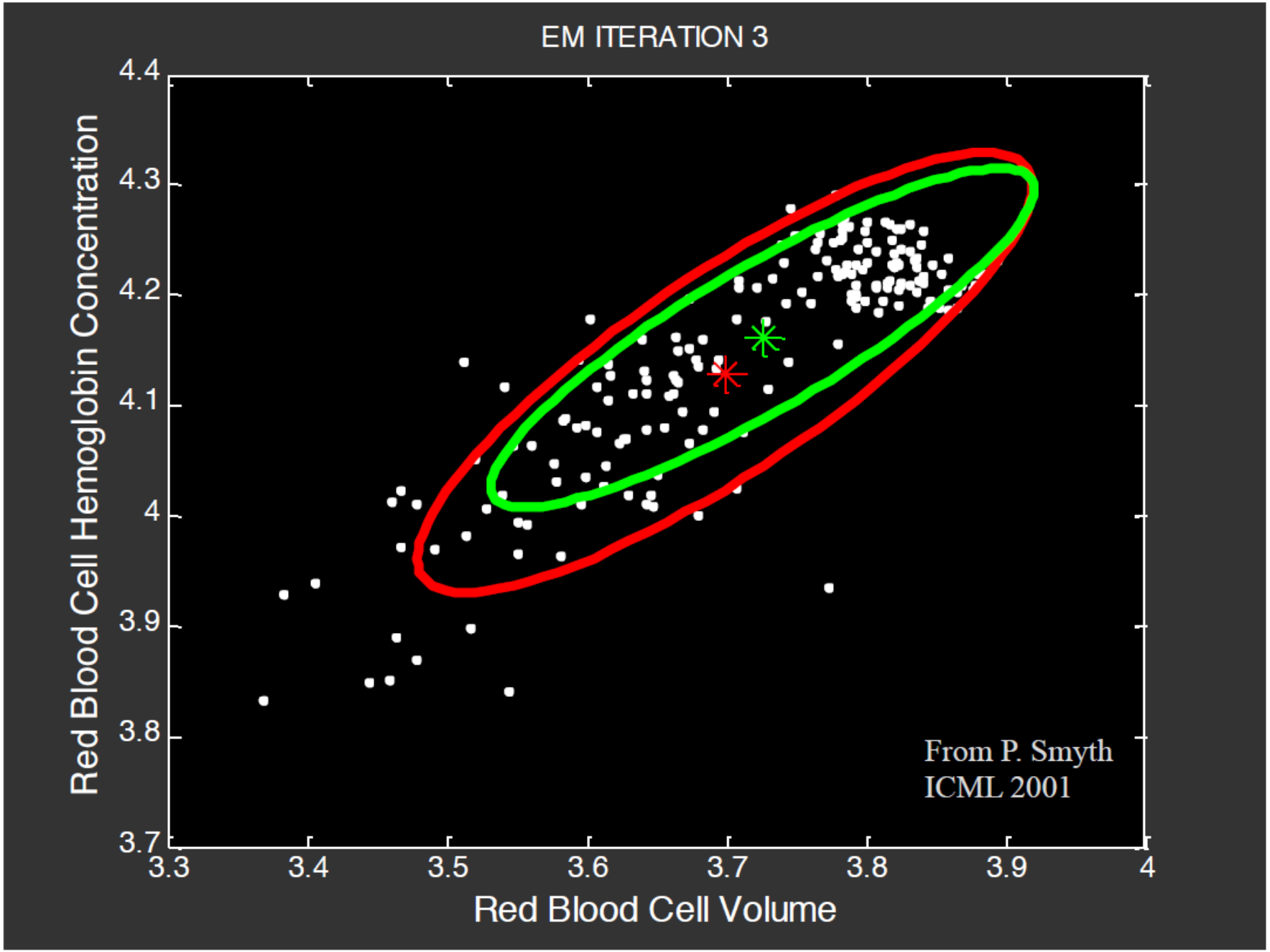


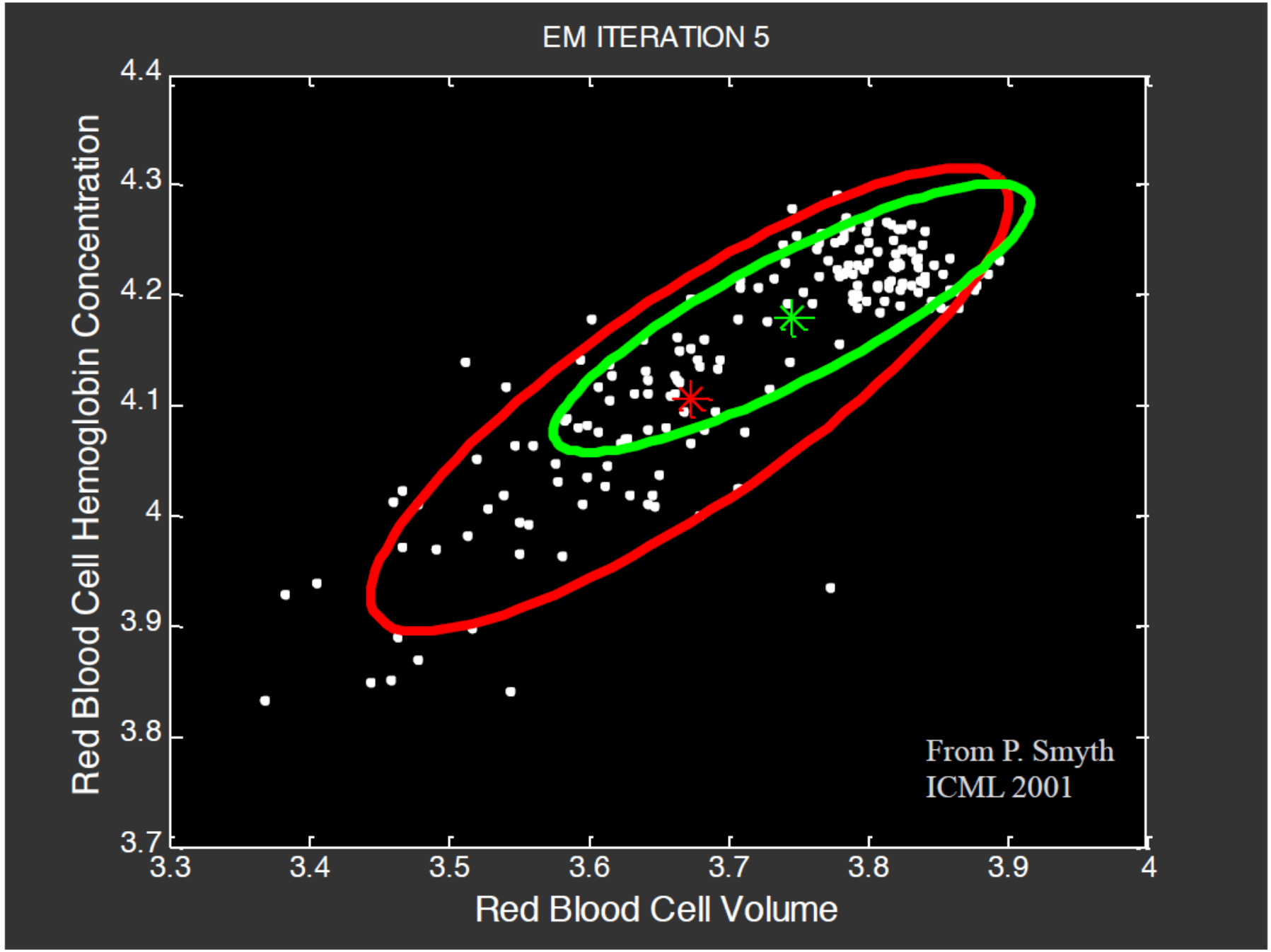
- Multivariate Gaussian models
  - $\mathcal{N}(x;\mu,\Sigma)=1/(2\pi)\uparrow d/2 \ |\Sigma|\uparrow -1/2 \ exp\{-1/2 \ (x-\mu)\uparrow T \Sigma\uparrow -1 \ (x-\mu)\}$

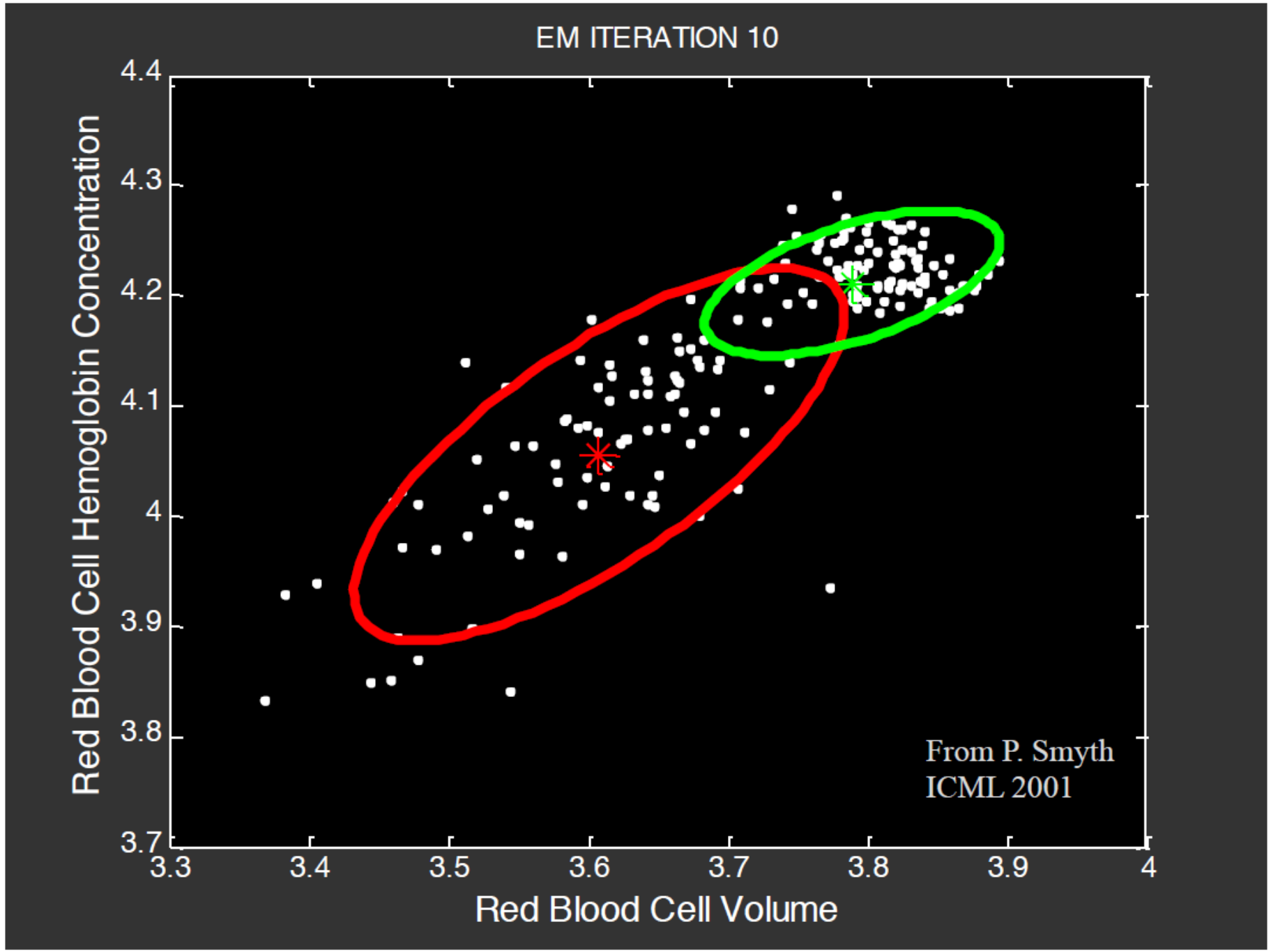
G. J. McLachlan and K. E. Bkasford. *Mixture Models: Inference and Applications to Clustering*. John Wiley & Sons, 1988.

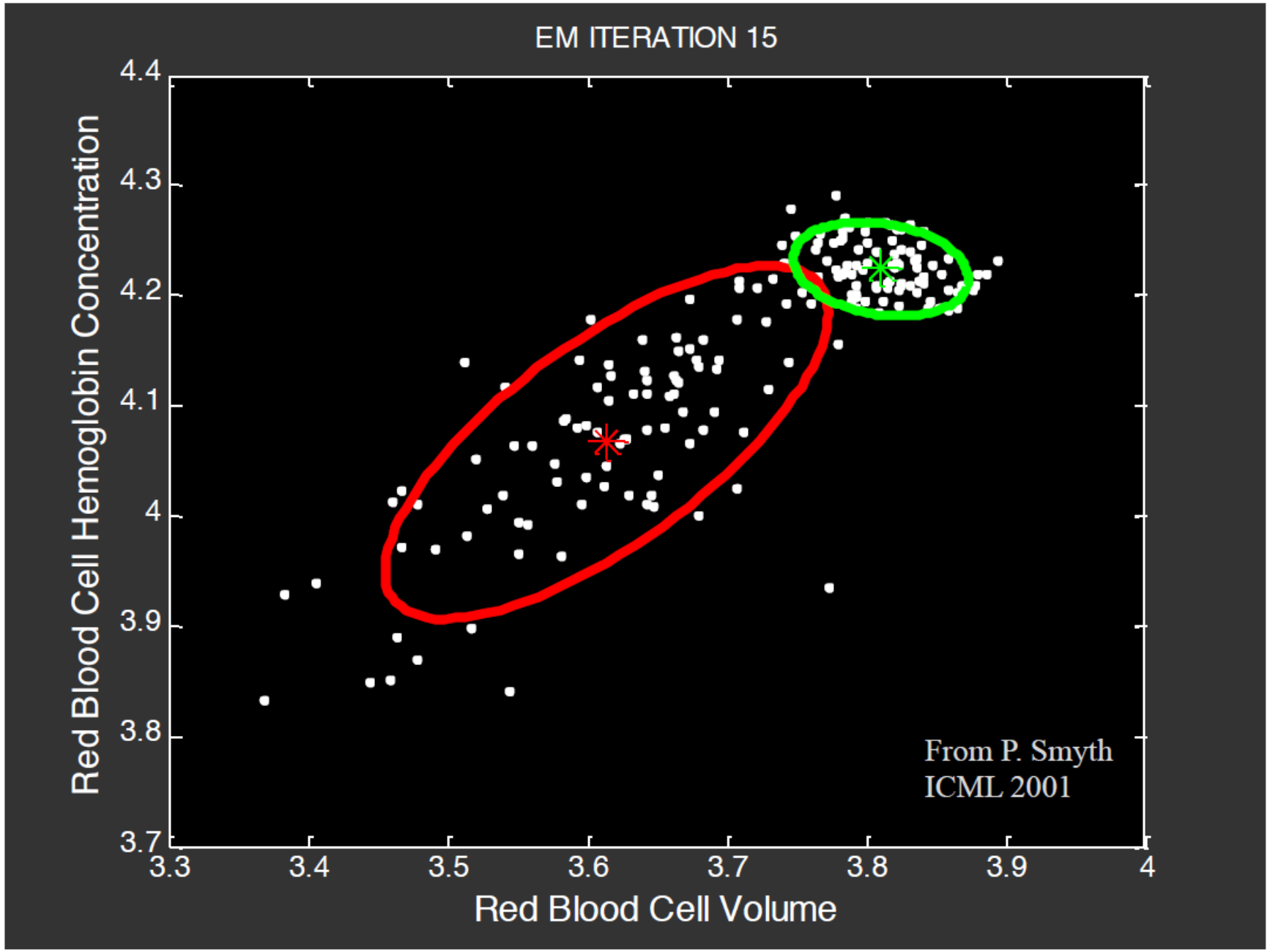

ANEMIA PATIENTS AND CONTROLS
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

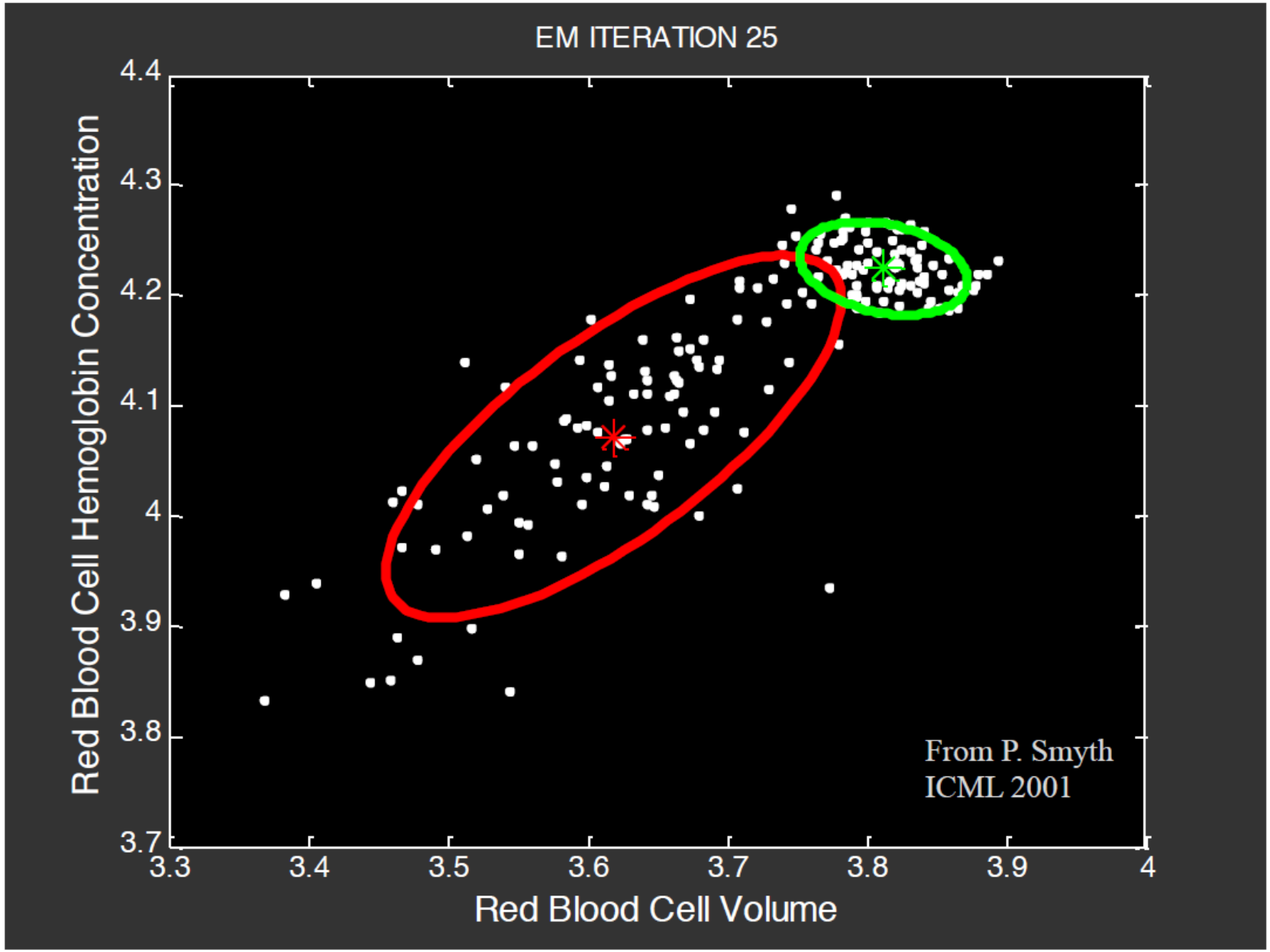

EM ITERATION 1
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

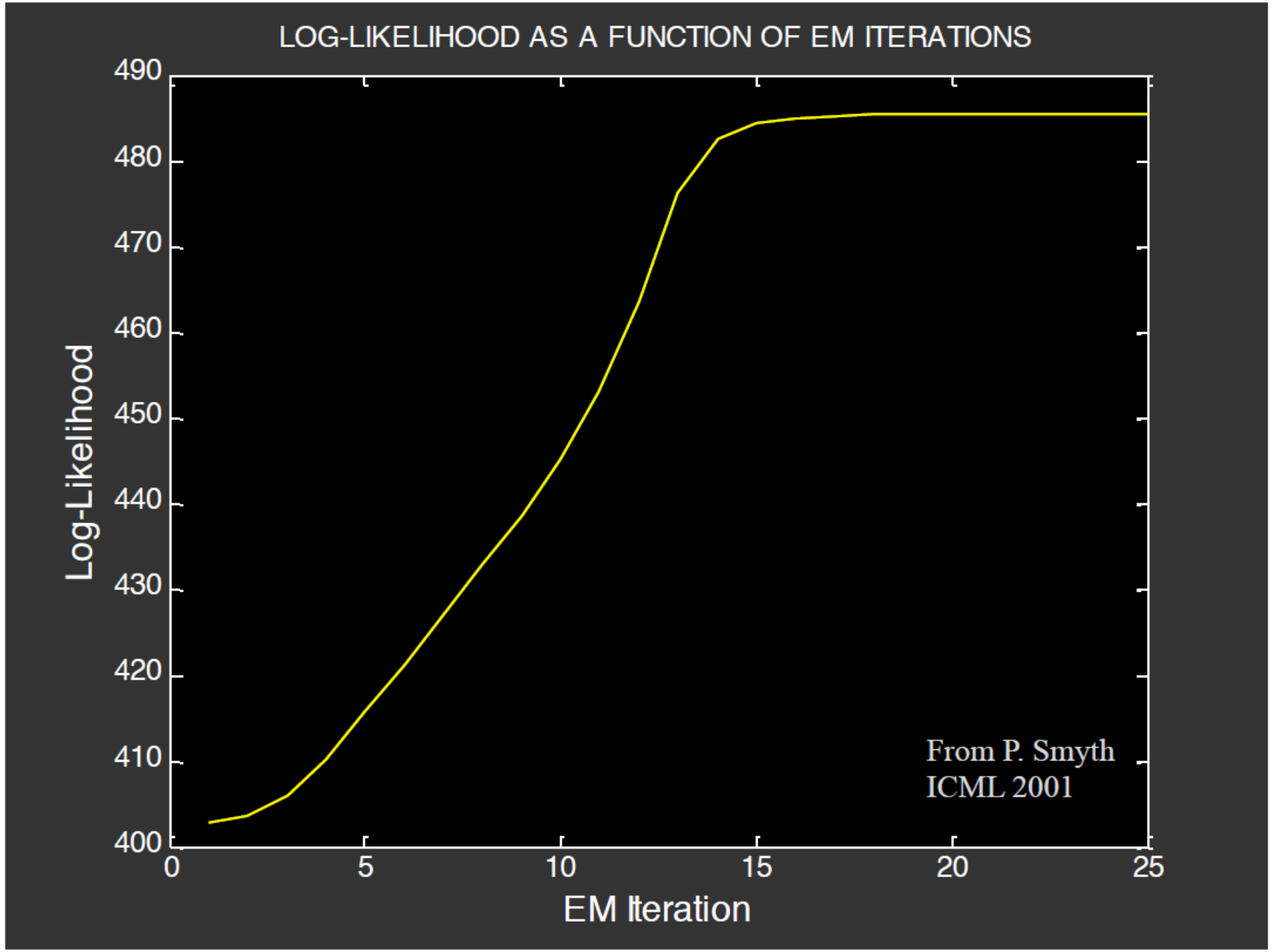

EM ITERATION 3
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 5
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 10
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 15
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 25
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


LOG-LIKELIHOOD AS A FUNCTION OF EM ITERATIONS
Log-Likelihood
EM Iteration
490
480
470
460
450
440
430
420
410
400
0
5
10
15
20
25
From P. Smyth
ICML 2001

Self-Organizing Maps (SOMs)

- Developed by professor Kohonen
- A type of artificial neural network (ANN) that is trained using unsupervised learning to produce a **low-dimensional** (typically two-dimensional), discretized representation of the input space of the training samples, called a map.

T. Kohonen, "The self-organizing map," in *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464-1480, Sep 1990.

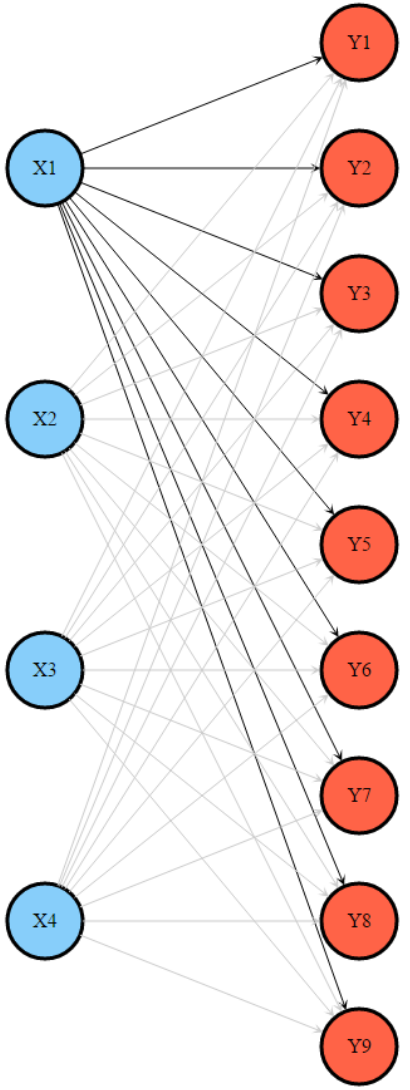
Self-Organizing Maps (SOMs)

- Unsupervised learning
- Competitive learning network
- Provides a topology preserving mapping from high-D to map units
- Detect features inherent to the problem, provide feature map
- Generalization capability

T. Kohonen, "The self-organizing map," in *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464-1480, Sep 1990.

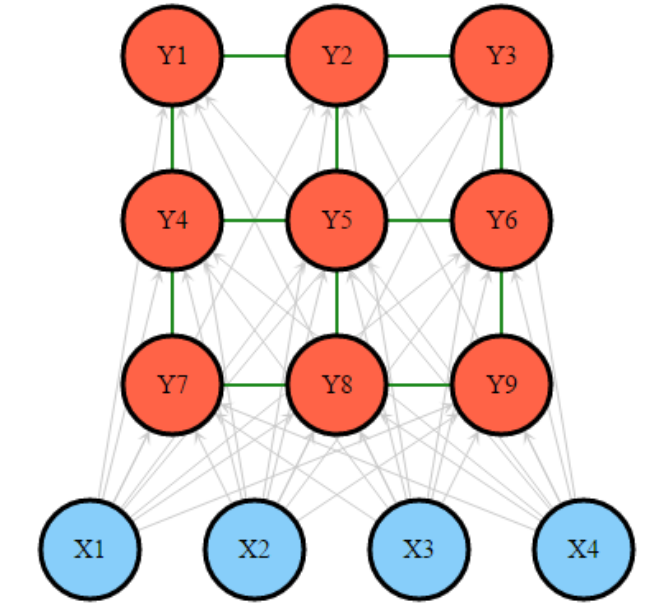
Network Architecture

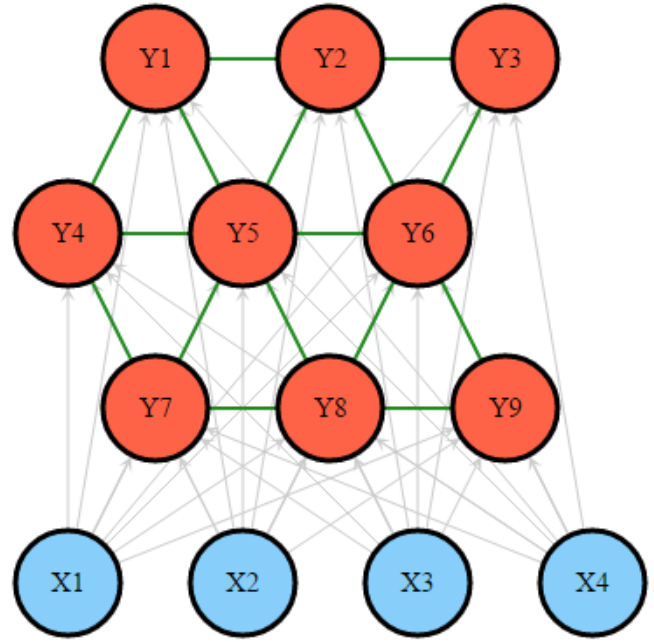
- Two layers of units
  - Input: n units (length of training vectors)
    - $V\downarrow 1 , V\downarrow 2 , V\downarrow 3 , \cdots , V\downarrow n$
  - Output: m units (number of categories)
    - Each node k has a weight vector $[w\downarrow 1k , w\downarrow 2k , w\downarrow 3k , \cdots , w\downarrow nk]$
- Input units fully connected with weights to output units
- Output neurons are ordered according to the topology of the map
  - Usually rectangular or hexagonal



Output Layer Topology

- Intra-layer("lateral") connections
  - Within output layer
  - Defined according to some topology
  - No weight between these connections, but used in algorithm for updating weights
- Often view output in spatial manner
  - Eg. A 1D or 2D arrangement
  - 2D: Rectangular or Hexagonal





SOM Algorithm

- Select output layer network topology
  - Initialize current neighborhood distance, $D(0)$, to a positive value
- Initialize weights from inputs to outputs to small random values
- Let $t=1$
- While computational bounds are not exceeded do
  - 1) Select an input sample $i↓l$
  - 2) Compute the square of the Euclidean distance of $i↓l$ from weight vectors ($w↓j$) associated with each output node

$$\sum k=1↑n▒(i↓l,k-w↓j,k\ (t))↑2$$

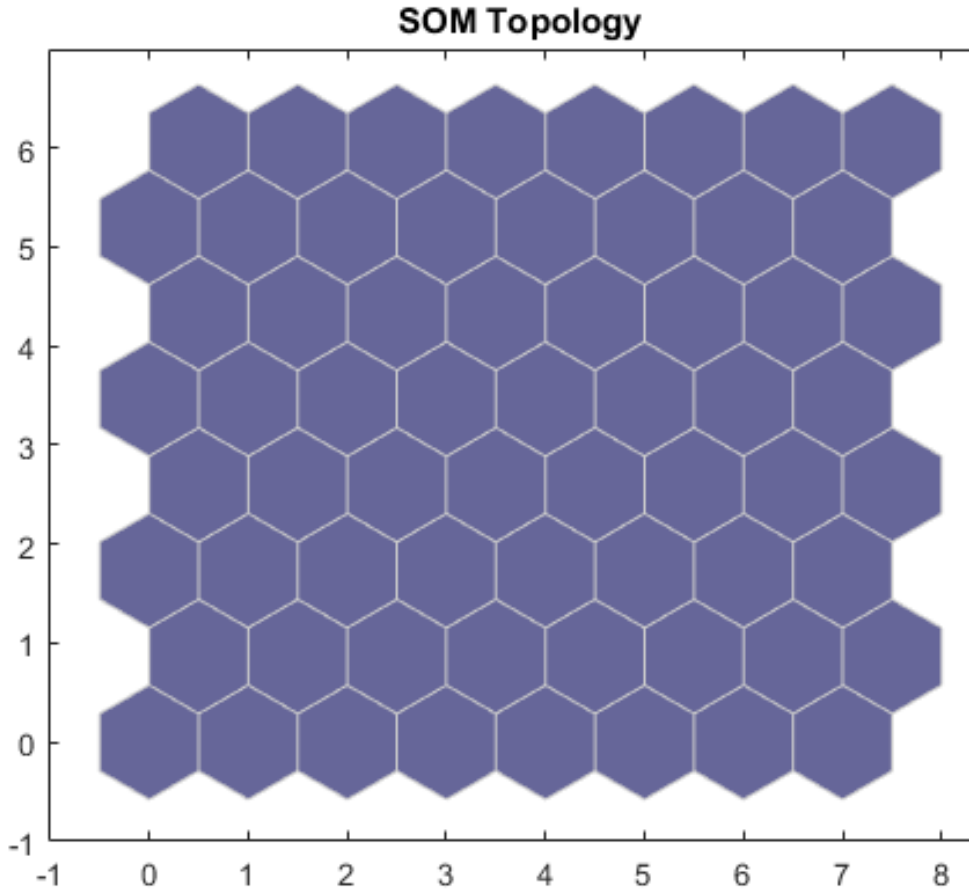
  - 3) Select output node $j↑*$ that has weight vector with minimum value from step 2)
  - 4) Update weights to all nodes within a topological distance given by $D(t)$ from $j↑*$, using the weight update rule:

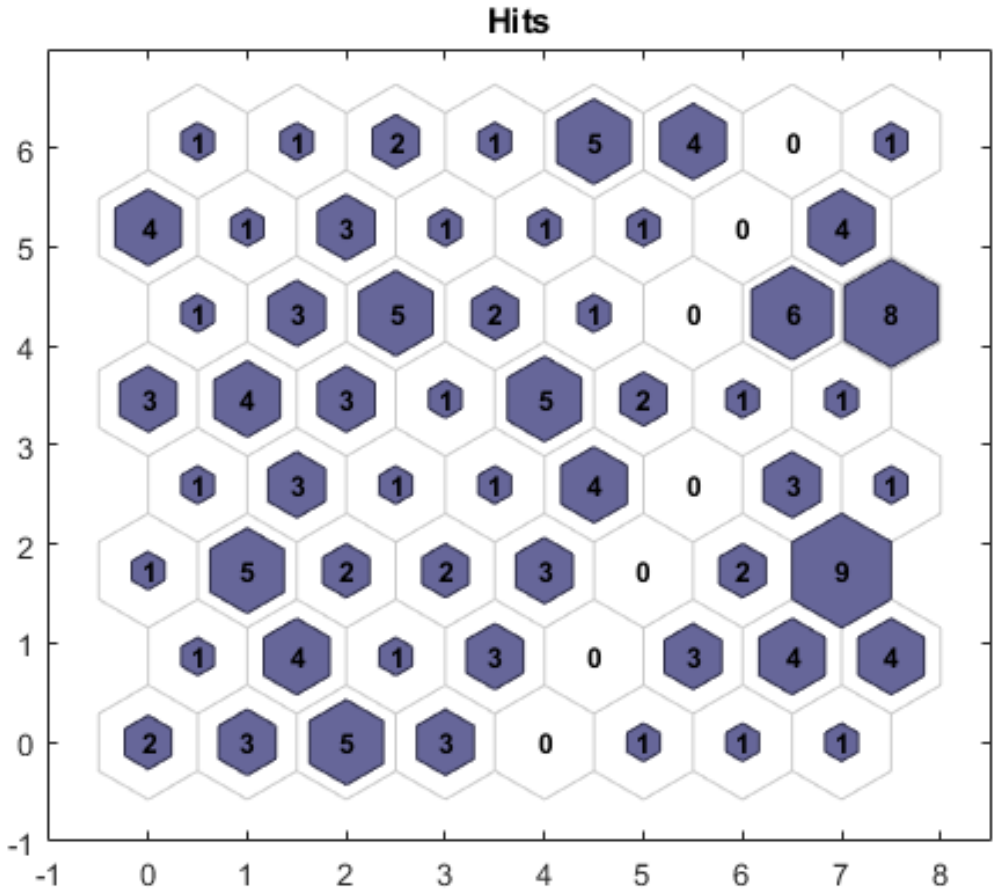
$$w↓j\ (t+1)=w↓j\ (t)+\eta(t)(i↓l-w↓j\ (t))$$

  - 5) Increment t
- Endwhile

Mehotra, K., Mohan, C. K., & Ranka, S. (1997). Elements of Artificial Neural Networks. MIT Press pp. 187-202

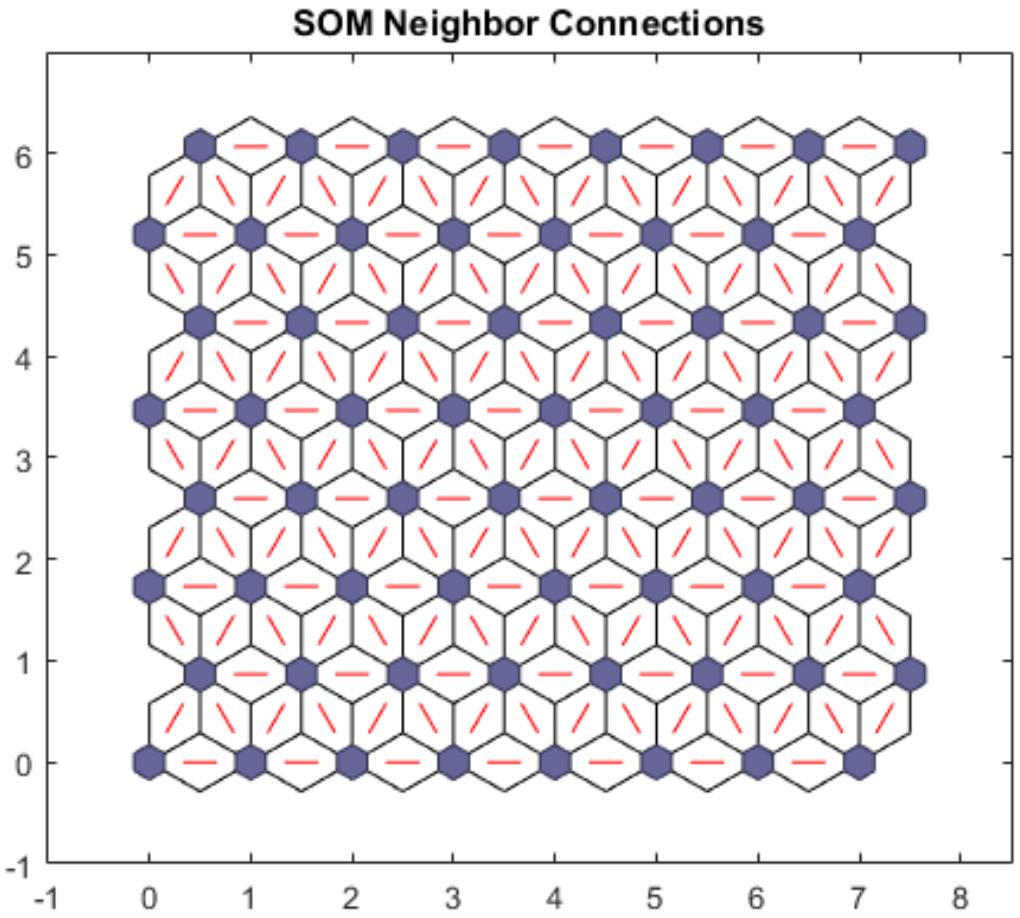
SOM Example: Iris Flower Dataset

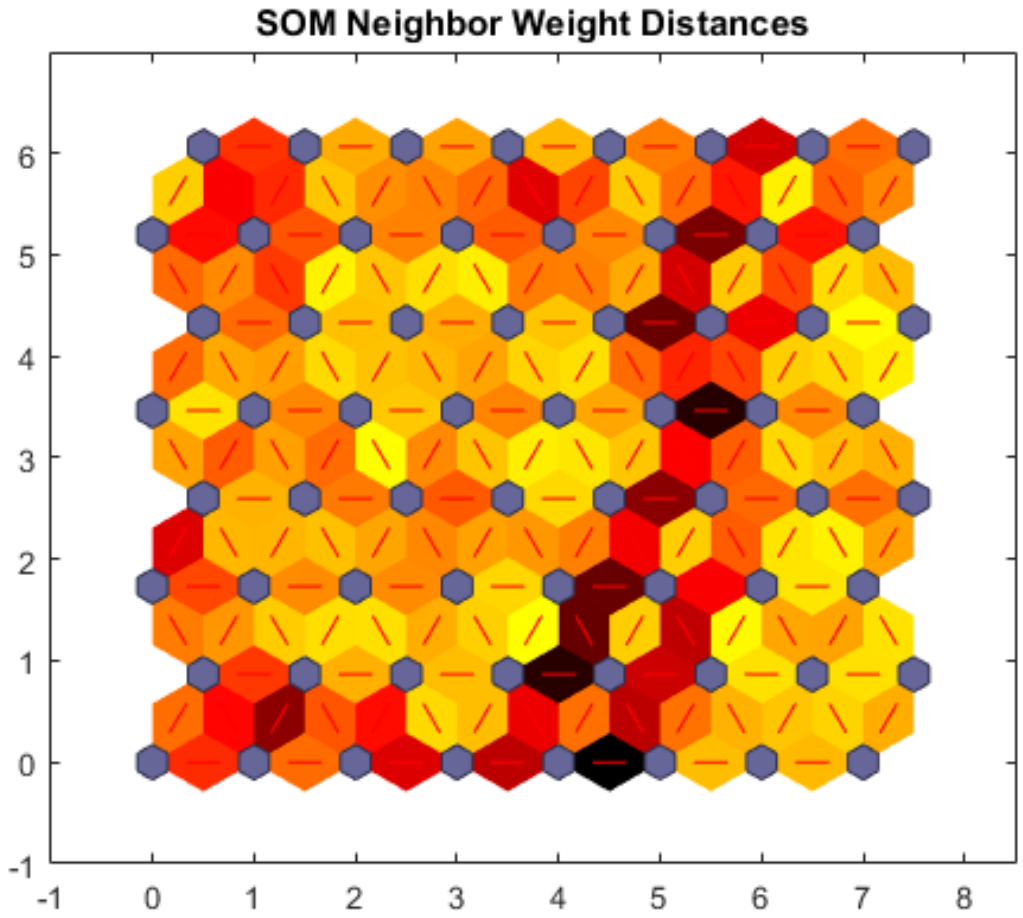




Source: technical Report on SOM Toolbox 2.0 for Matlab

SOM Example: Iris Flower Dataset

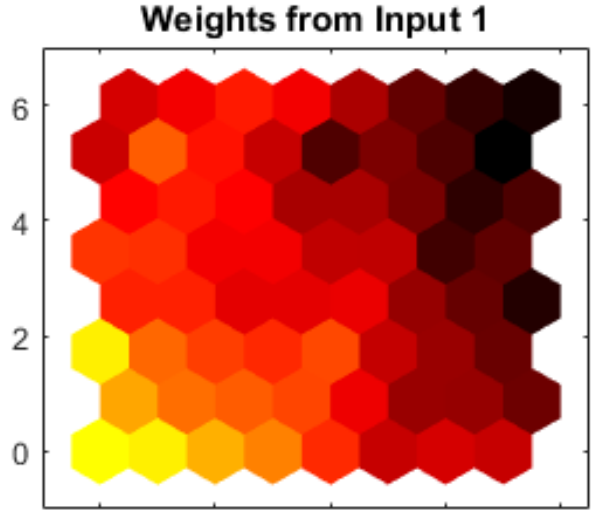


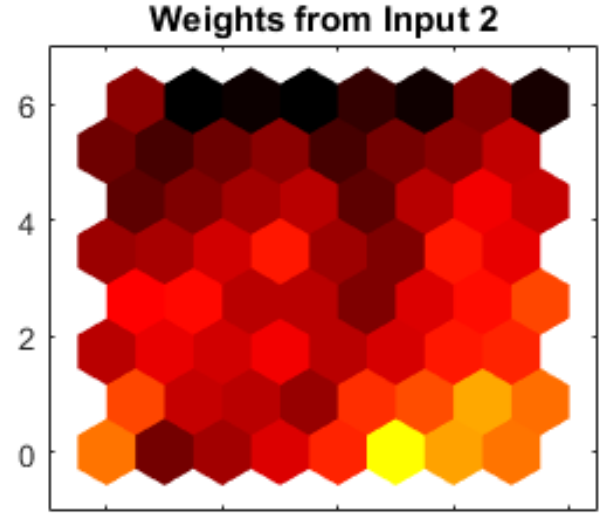


Source: technical Report on SOM Toolbox 2.0 for Matlab

SOM Example: Iris Flower Dataset

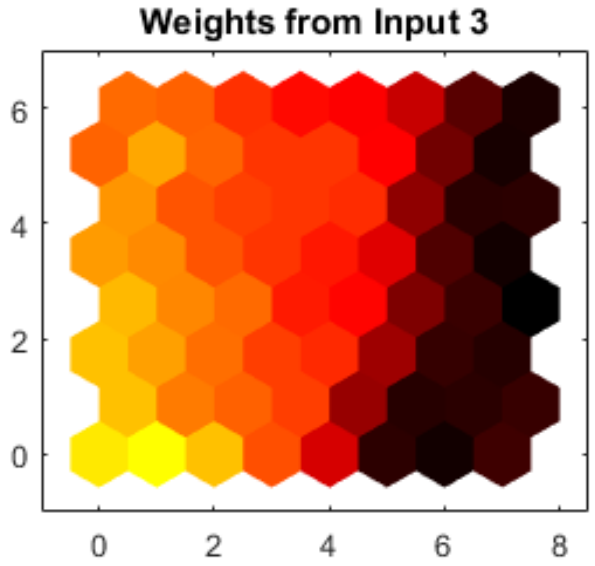
Sepal length

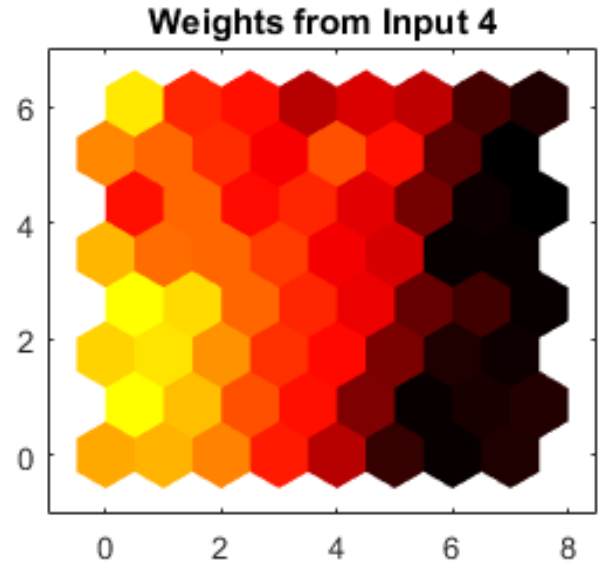




Sepal width

Pedal length





Pedal width

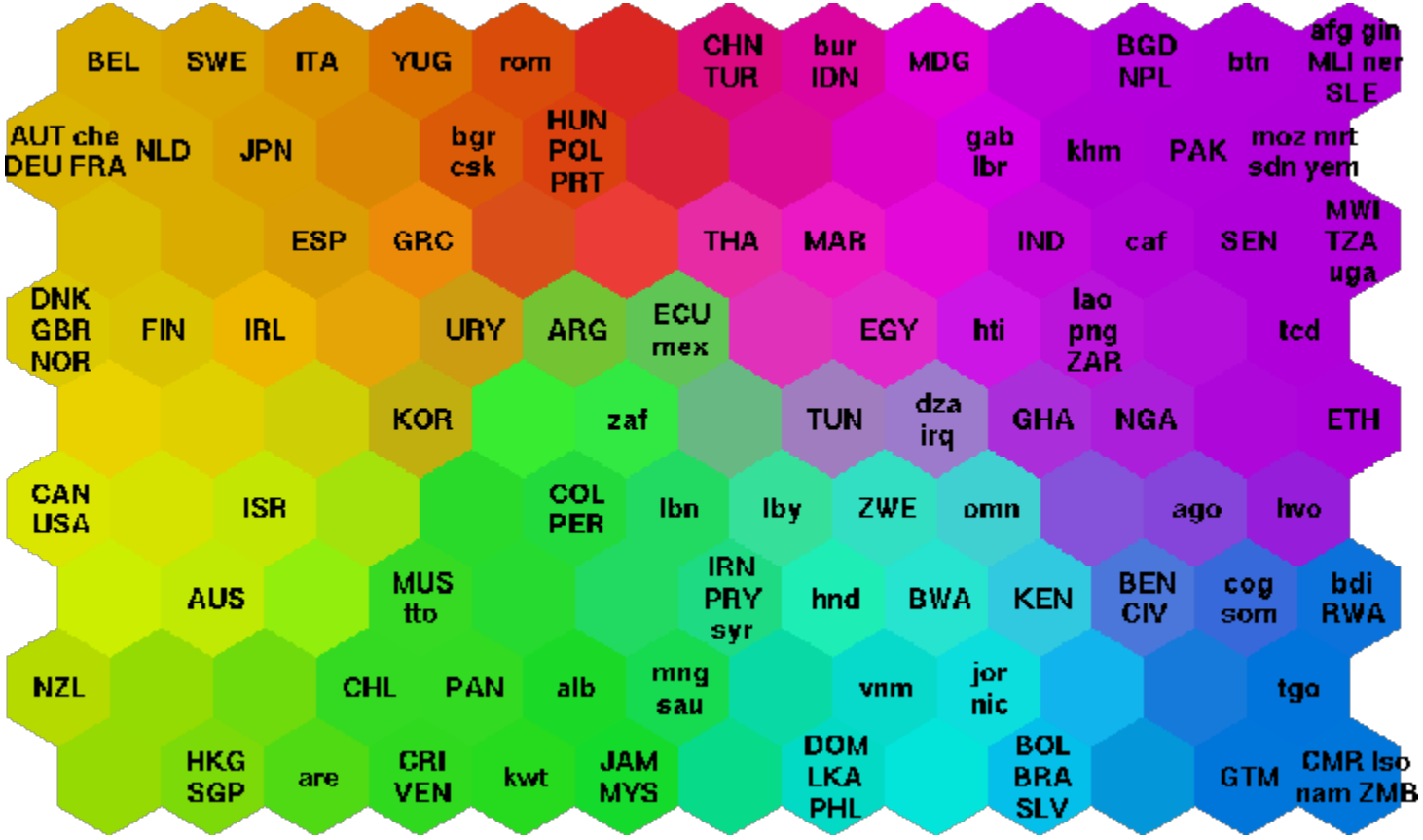
Source: technical Report on SOM Toolbox 2.0 for Matlab

SOM Example: Iris Flower Dataset

Method	SOM	K-means
Quadratic error	86.67	91.35
Std(Qerr)	0.33	25.76
ClassErr	9.22	15.23
Struct Err	0	18

F. Bacao, V. Lobo, and M. Painho. Self-organizing maps as substitutes for k-means clustering. In Computational Science - ICCS 2005, Pt. 3, Lecture Notes in Computer Science, pages 209–217, 2005.

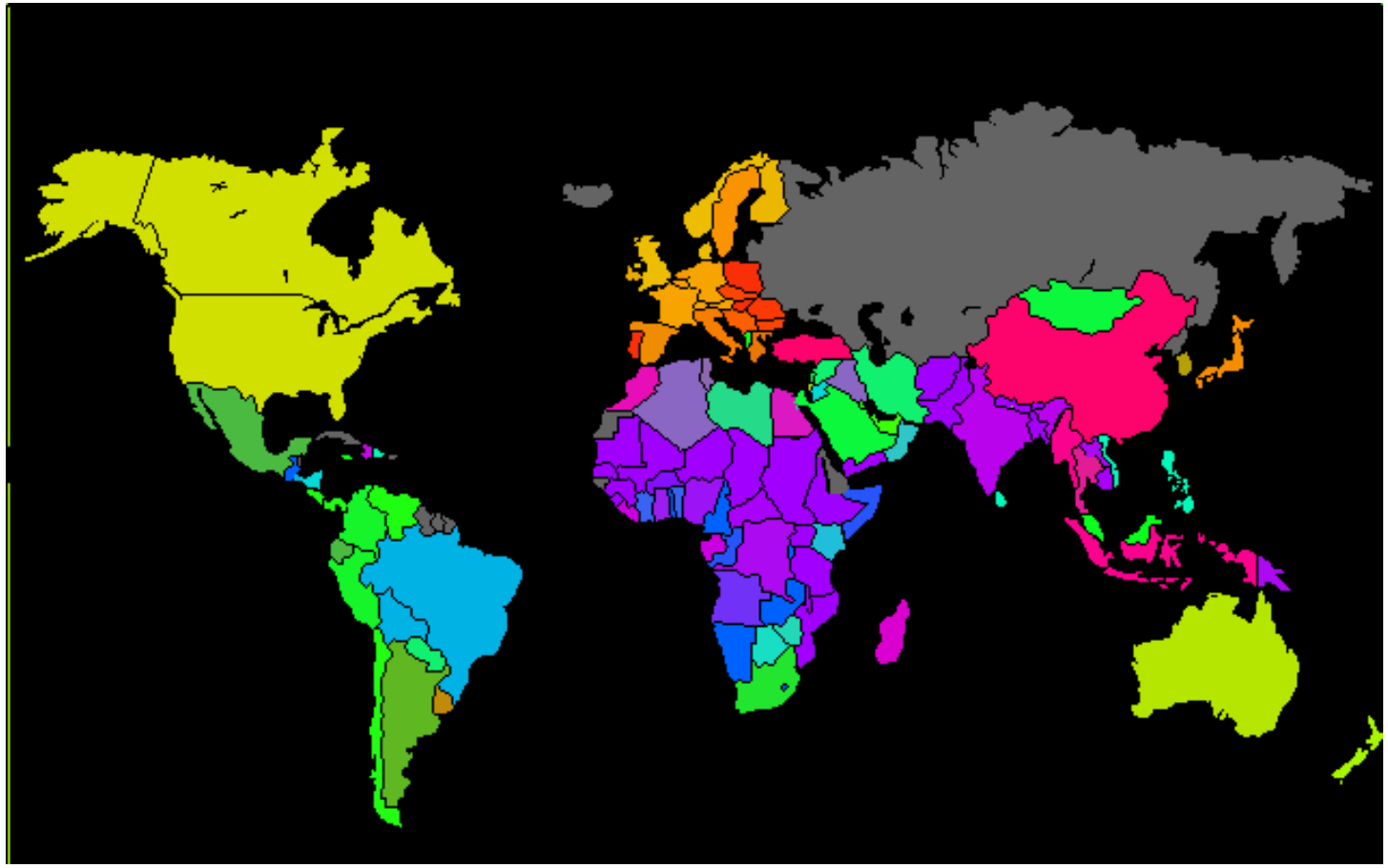
SOM – Visualization of **World Poverty Map**



Data: World Bank statistics of countries in 1992

Source: Neural Networks Research Centre, Helsinki University of Technology, Finland

SOM – Visualization of **World Poverty Map**



Data: World Bank statistics of countries in 1992

Source: Neural Networks Research Centre, Helsinki University of Technology, Finland

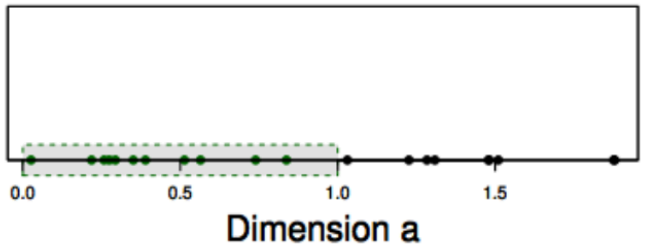
Clustering High-Dimensional Data

Clustering High-Dimensional Data

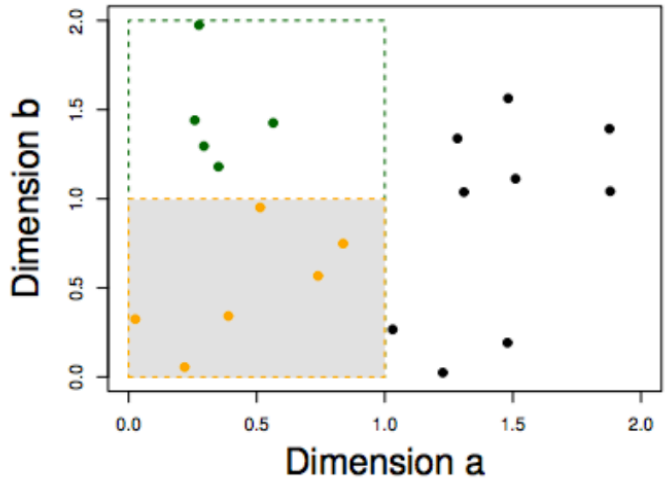
- Clustering high-dimensional data
  - Many applications: data mining, DNA micro-array dataMany
  - Challenges:
    - Distance measure becomes meaning less-due to equi-distance
    - Many irrelevant dimensions may mask clusters
    - Clusters may exist only in some subspaces
- Approaches
  - Subspace-Based Methods
  - Correlation-Based Methods
  - Bi-Clustering Methods

The Curse of Dimensionality

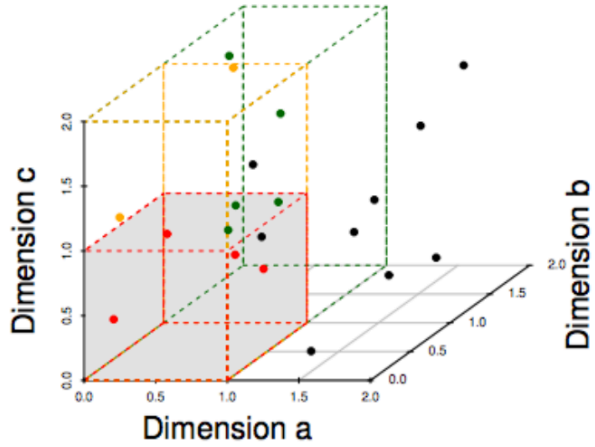
(graphs adapted from Parsons *Subspace Clustering for High Dimensional Data: A Review* KDD Explorations 2004)



(a) 11 Objects in One Unit Bin



(b) 6 Objects in One Unit Bin



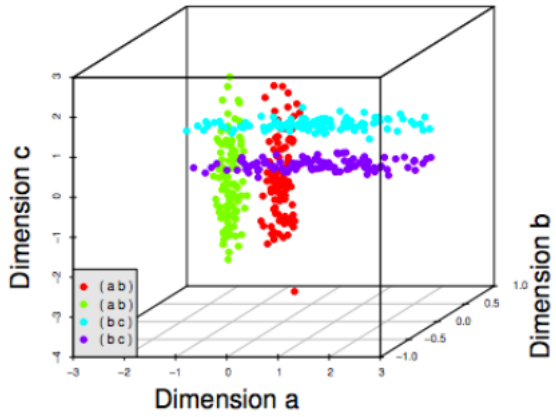
(c) 4 Objects in One Unit Bin

From one dimension to three dimensions, points go from packed to sparse. As the dimensions increase, points gradually become equi-distance, so distance measure becomes meaningless. (dataset: 20 points, 3 dimensions, random number between 0 and 2 in each dimension)

Subspace clustering

(graphs adapted from Parsons *Subspace Clustering for High Dimensional Data: A Review* KDD Explorations 2004)

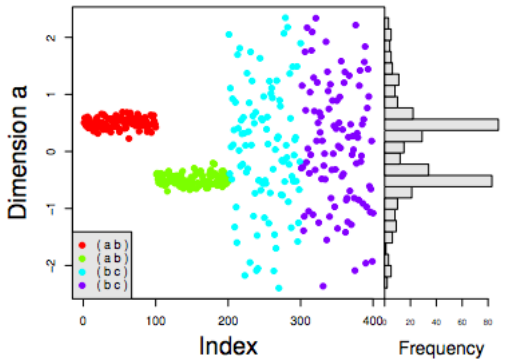
- Clusters may exist only in some subspaces
- Subspace-clustering: find clusters in all the subspaces



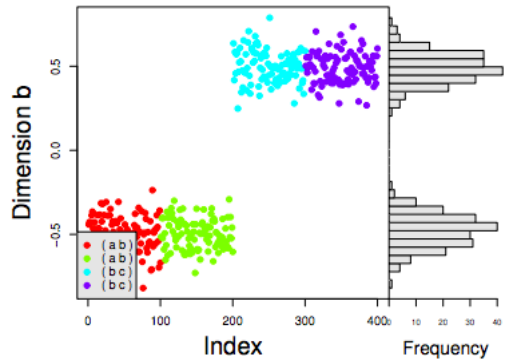
400 instances in 3 dimensions, divided into 4 clusters of 100 Instances. First two clusters exist in dimensions a and b. The data forms a normal distribution with means 0.5 and -0.5 in dimension a and 0.5 in dimension b. The second two clusters are in dimension b and c and were generated in the same manner.

Subspace clustering

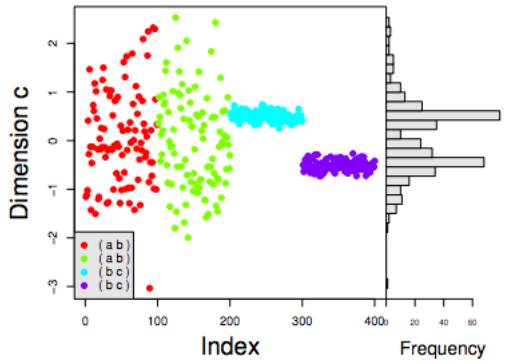
(graphs adapted from Parsons *Subspace Clustering for High Dimensional Data: A Review* KDD Explorations 2004)



(a) Dimension a



(b) Dimension b

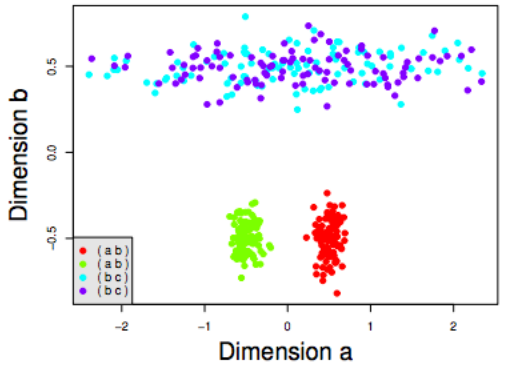


(c) Dimension c

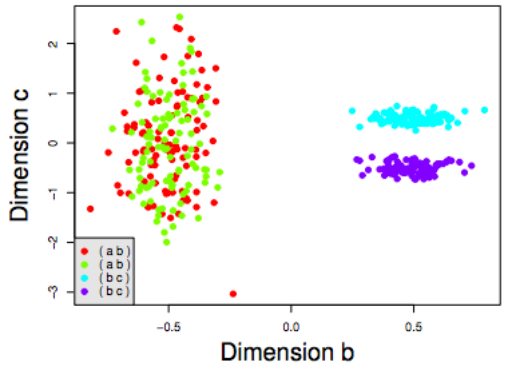
- The figures above shows the data projected in a single dimension (organized by index on the x-axis for ease of interpretation). We can see that none of these projections of the data are sufficient to fully separate the four clusters. Alternatively, if we only remove one dimension, we produce the graphs in next page.

Subspace clustering

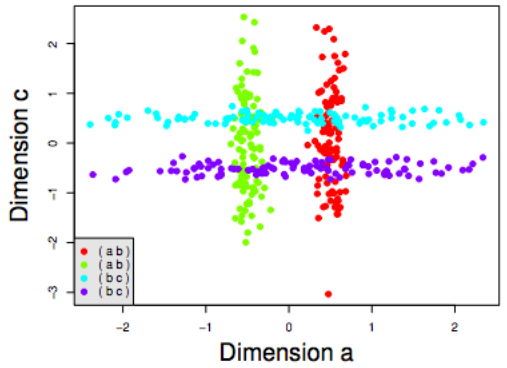
(graphs adapted from Parsons *Subspace Clustering for High Dimensional Data: A Review* KDD Explorations 2004)



(a) Dims a & b



(b) Dims b & c



(c) Dims a & c

- The first two clusters (red and green) are easily separated from each other and the rest of the data when viewed in dimensions a and b. This is because those clusters were created in dimensions a and b and removing dimension c removes the noise from those two clusters. The other two clusters (blue and purple)completely overlap in this view since they were created in dimensions b and c and removing c made them indistinguishable from one another. Thus, the key to finding each of the clusters in this dataset is to look in the appropriate subspaces.

Subspace Clustering Methods

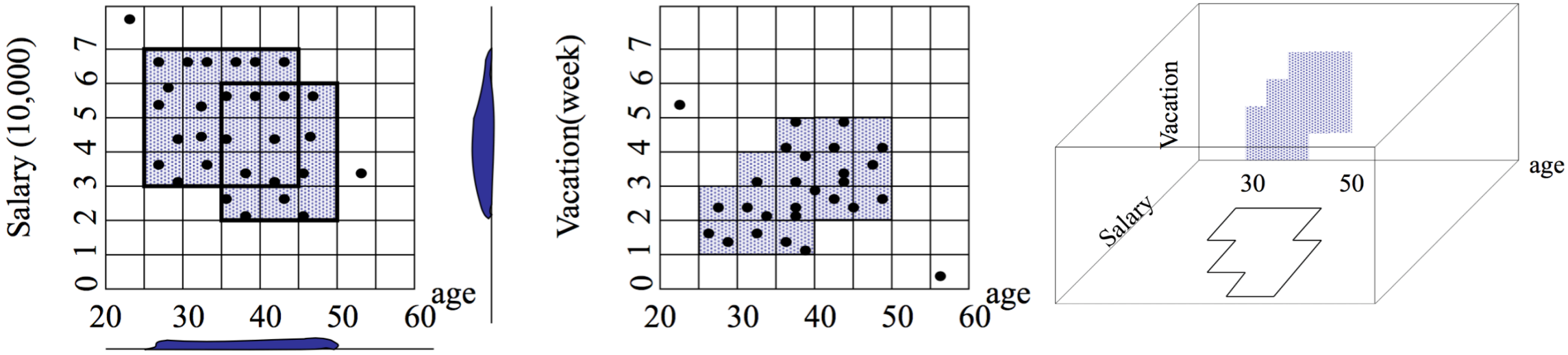
- Bottom-Up Subspace Search Methods
  - CLIQUE
  - ENCLUS
  - MAFIA
  - CBF
- Iterative Top-Down Subspace Search Methods
  - PROCLUS
  - ORCLUS
  - FINDIT
  - δ-Clusters

CLIQUE

- CLIQUE(Clustering In QUEst)(Agrawal, Gehrke, Gunopulos, Raghavan: SIGMOD'98)
- CLIQUE is a **density-based** and **grid-based** subspace clustering algorithm
  - **Grid-based**: It discretizes the data space through a grid and estimates the density by counting the number of points in a grid cell
  - **Density-based**: A cluster is a maximal set of connected dense units in a subspace(dense unit: total data points in the unit exceeds the input model parameter)
  - **Subspace clustering**: A subspace cluster is a set of neighboring dense cells in an arbitrary subspace. It also discovers some minimal descriptions of the clusters

Example of CLIQUE:

graph from Jyoti Yadav *Subspace Clustering using CLIQUE: An Exploratory Study* 2014 pg4



Start at 1-D space and discretize numerical intervals in each axis into grid

Find dense regions(clusters) in each subspace and generate their minimal descriptions

Use the dense regions to find promising candidates in 2-D space based on the Apriori principle

Major Steps of the CLIQUE Algorithm

- Identify subspaces that contain clusters
  - Partition the data space and find the number of points that lie inside each cell of the partition
  - Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters
  - Determine dense units in all subspaces of interests
  - Determine connected dense units in all subspaces of interests
- Generate minimal descriptions for the clusters
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determine minimal cover for each cluster

Strengths and Weaknesses of CLIQUE

- Strengths
  - Automatically finds subspaces of the highest dimensionality as long as high density clusters exist in those subspaces
  - Insensitive to the order of records in input and does not presume some canonical data distribution
  - Scales linearly with the size of input and has good scalability as the number of dimensions in the data increases
- Weaknesses
  - As in all grid-based clustering approaches, the quality of the results crucially depends on the appropriate choice of the number and width of the partitions and grid cells

Thanks!

Group: #14

Title: Cluster Analysis