

CS 412

APRIL 21ST – UNSUPERVISED LEARNING

Administrivia

Project Meetings

- This week,
- I will extend them to next week as well
- You may sign up for more than one

extend through next week

Midterm exam

- Almost finished
- Graded copies back + solution video Thursday

Hw4 Due Thursday.
OH 5-7 pm Th

Hw5 Short assignment
Due next Thursday

How to choose a clustering algorithm

Clustering research has a long history. A vast collection of algorithms are available.

- We only introduced several main algorithms.

Choosing the “best” algorithm is a challenge.

- Every algorithm has limitations and works well with certain data distributions.
- It is very hard, if not impossible, to know what distribution the application data follow. The data may not fully follow any “ideal” structure or distribution required by the algorithms.

→ One also needs to decide how to standardize the data, to choose a suitable distance function and to select other parameter values.

also a problem for KNN
without labels there is no accuracy
distance? esp in n-dimensions.

Choose a clustering algorithm (cont ...)

Due to these complexities, the common practice is to

- run several algorithms using different distance functions and parameter settings, and
- then carefully analyze and compare the results.

The interpretation of the results must be based on insight into the meaning of the original data together with knowledge of the algorithms used.

Clustering is highly application dependent and to certain extent subjective (personal preferences).

Cluster Evaluation: hard problem

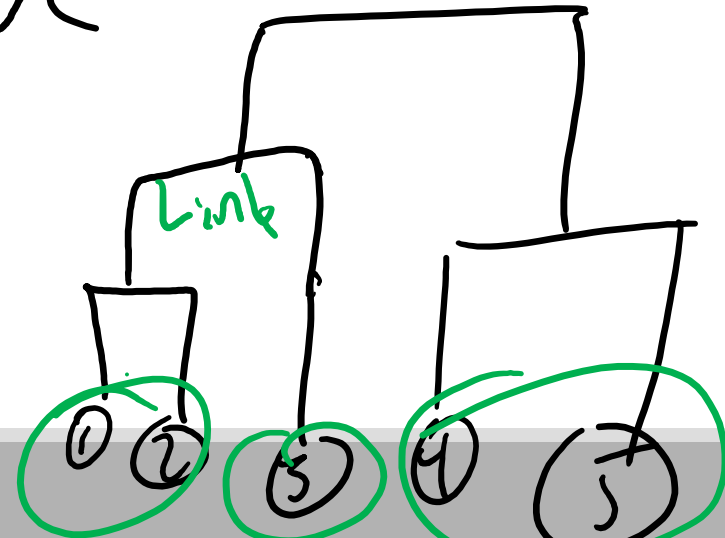
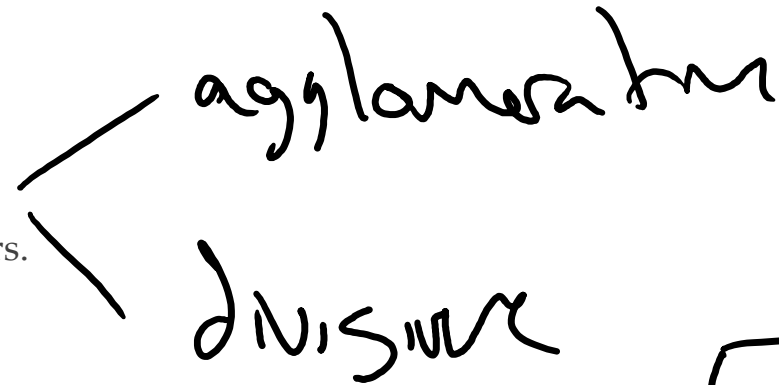
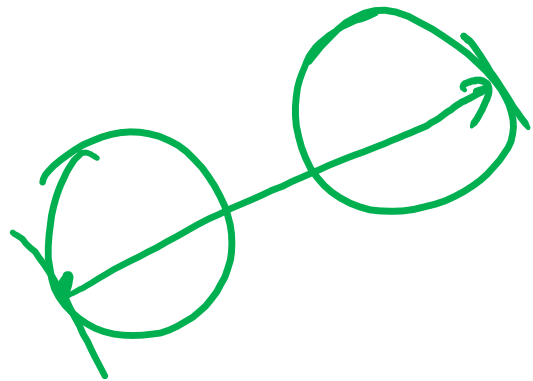
The quality of a clustering is very hard to evaluate because

- We do not know the correct clusters

there is no "ground truth"

Some methods are used:

- User inspection
 - Study centroids, and spreads
 - Rules from a decision tree.
 - For text documents, one can read some documents in clusters.



Cluster evaluation: ground truth

We use some labeled data (for classification)

Assumption: Each class is a cluster.

After clustering, a confusion matrix is constructed. From the matrix, we compute various measurements, entropy, purity, precision, recall and F-score.

- Let the classes in the data D be $C = (c_1, c_2, \dots, c_k)$. The clustering method produces k clusters, which divides D into k disjoint subsets, D_1, D_2, \dots, D_k .

Indirect evaluation

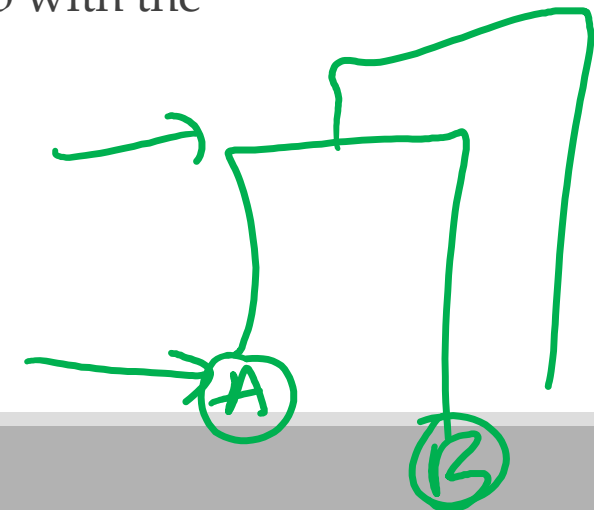
In some applications, clustering is not the primary task, but used to help perform another task.

We can use the performance on the primary task to compare clustering methods.

For instance, in an application, the primary task is to provide recommendations on book purchasing to online shoppers.

- If we can cluster books according to their features, we might be able to provide better recommendations.
- We can evaluate different clustering algorithms based on how well they help with the recommendation task.
- Here, we assume that the recommendation can be reliably evaluated.

Combine
unsupervised - cluster books
supervised - (recommen) dations



Aspects of clustering

A clustering algorithm

- Partitional clustering
- Hierarchical clustering

◦ *Distributional*

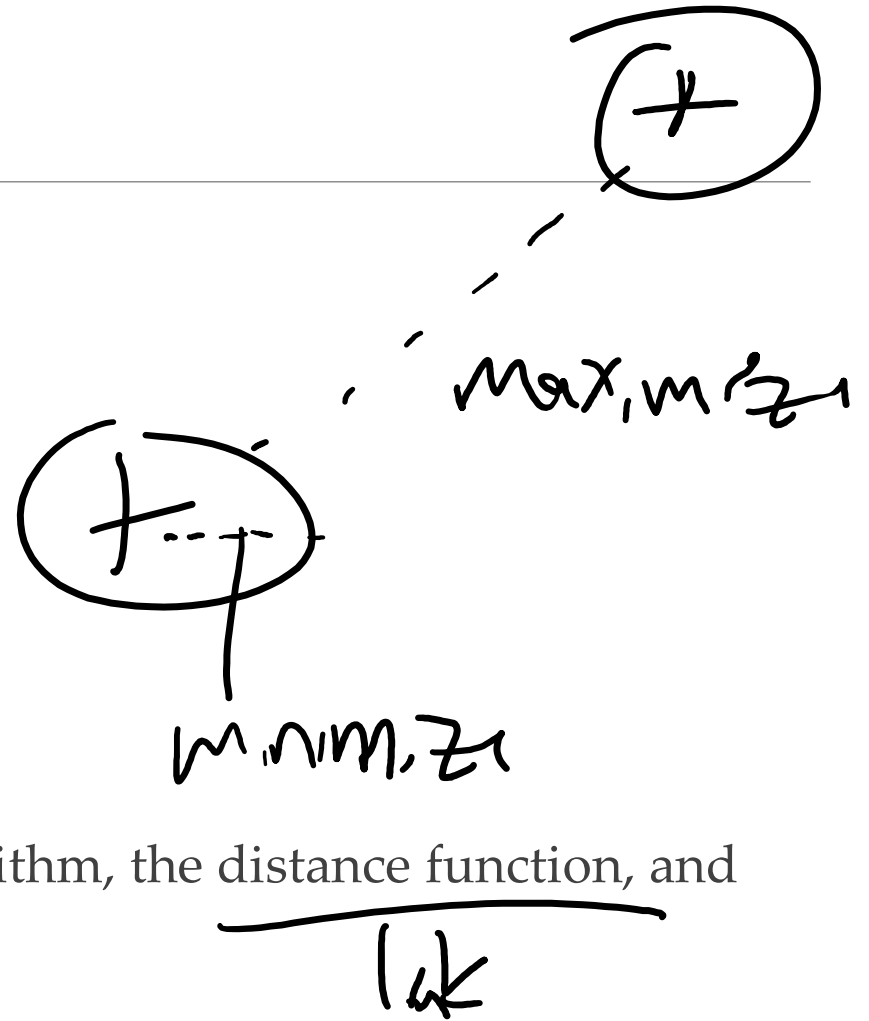
non-overlapping

A distance (similarity, or dissimilarity) function

Clustering quality

- Inter-clusters distance \Rightarrow maximized
- Intra-clusters distance \Rightarrow minimized

The quality of a clustering result depends on the algorithm, the distance function, and the application.



K-means algorithm

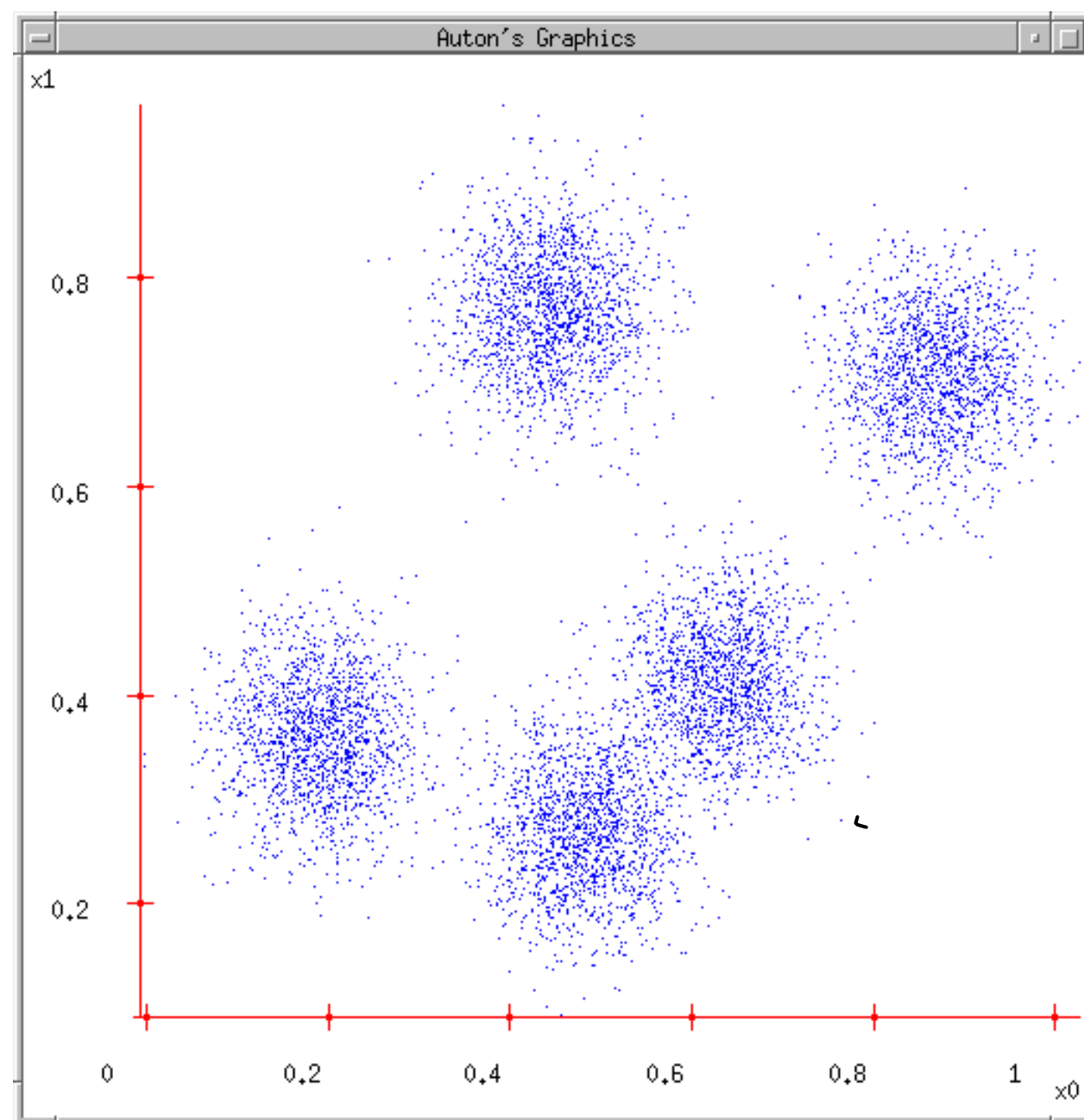
Given k , the k-means algorithm works as follows:

- Randomly choose k data points (seeds) to be the initial centroids, cluster centers
- Assign each data point to the closest centroid
- Re-compute the centroids using the current cluster memberships.
- If a convergence criterion is not met, go to 2).

Common ways to represent clusters

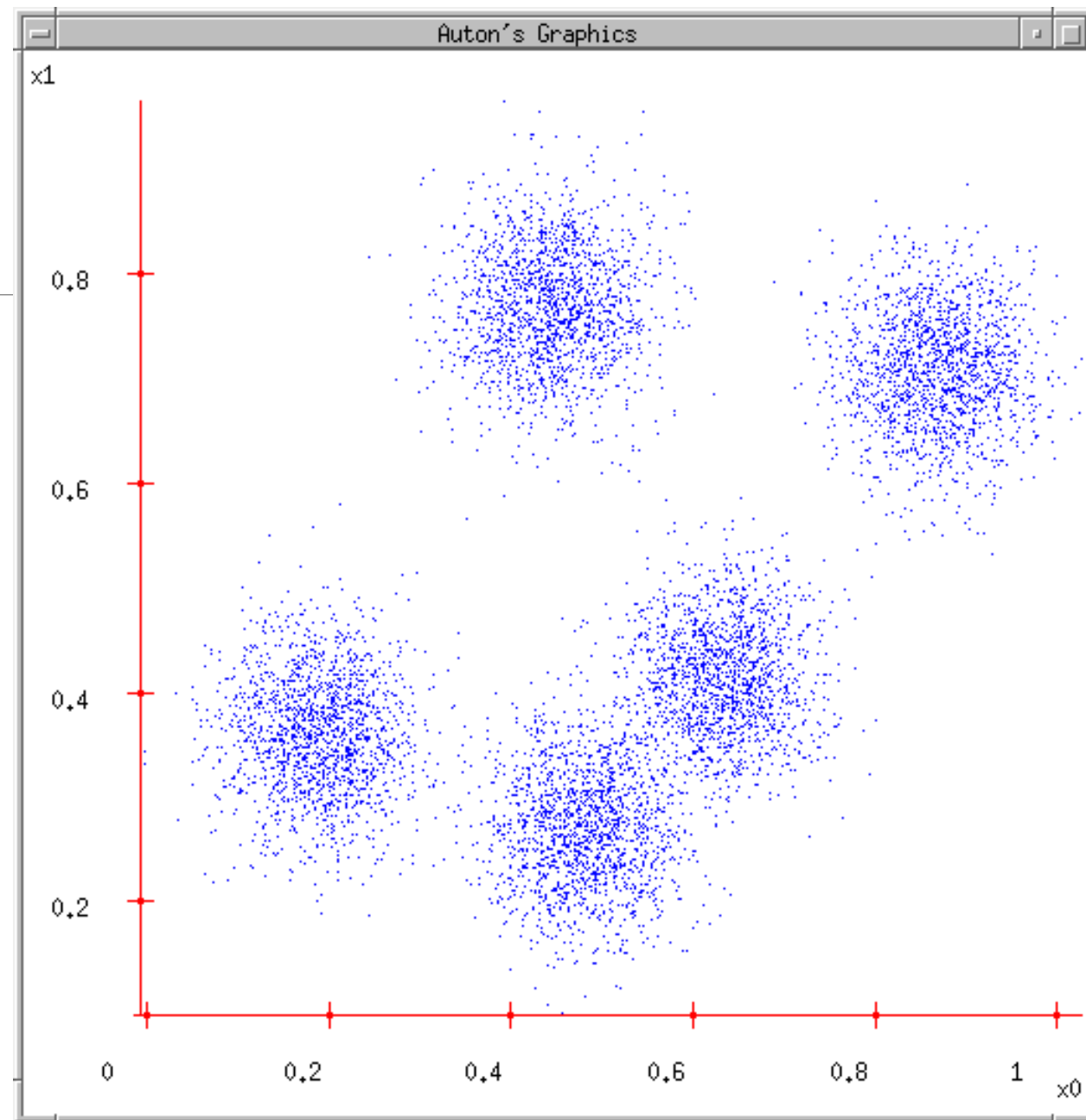
Use the centroid of each cluster to represent the cluster.

- compute the radius and
- standard deviation of the cluster to determine its spread in each dimension
- The centroid representation alone works well if the clusters are of the hyper-spherical shape.
- If clusters are elongated or are of other shapes, centroids are not sufficient



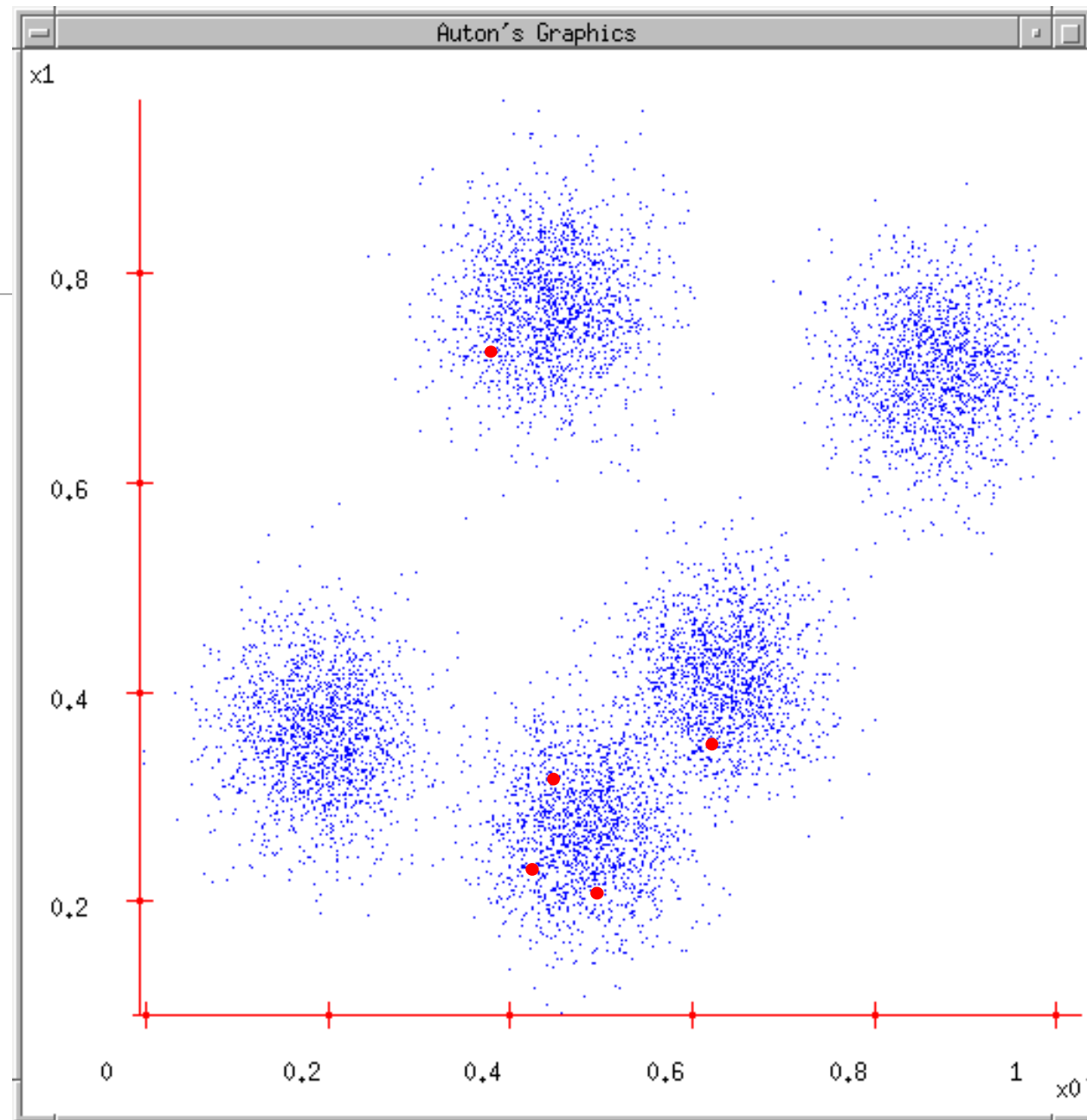
K-means

1. Ask user how many clusters they'd like.
(e.g. $k=5$)



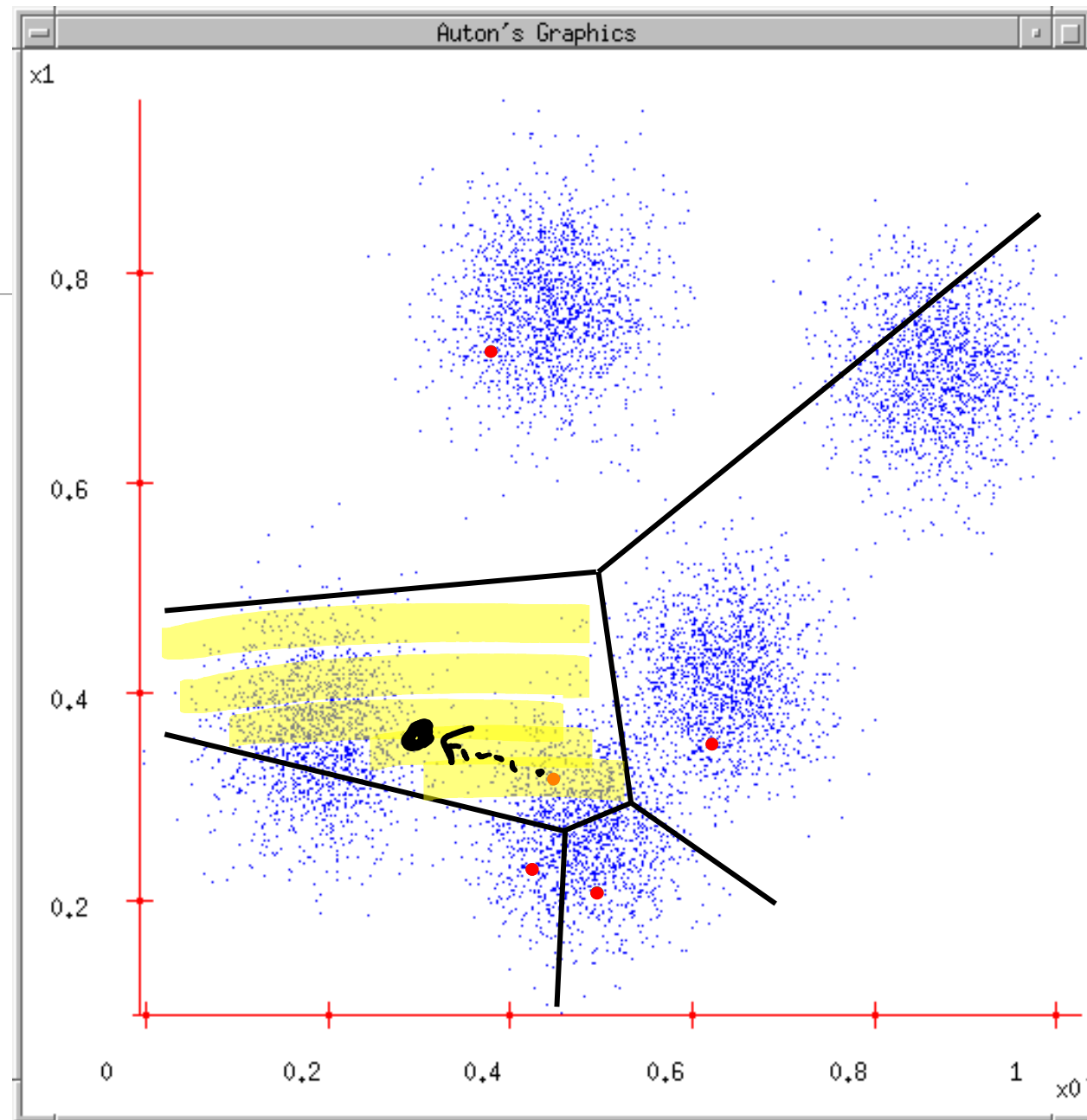
K-means

1. Ask user how many clusters they'd like.
(e.g. $k=5$)
2. Randomly guess k cluster Center locations



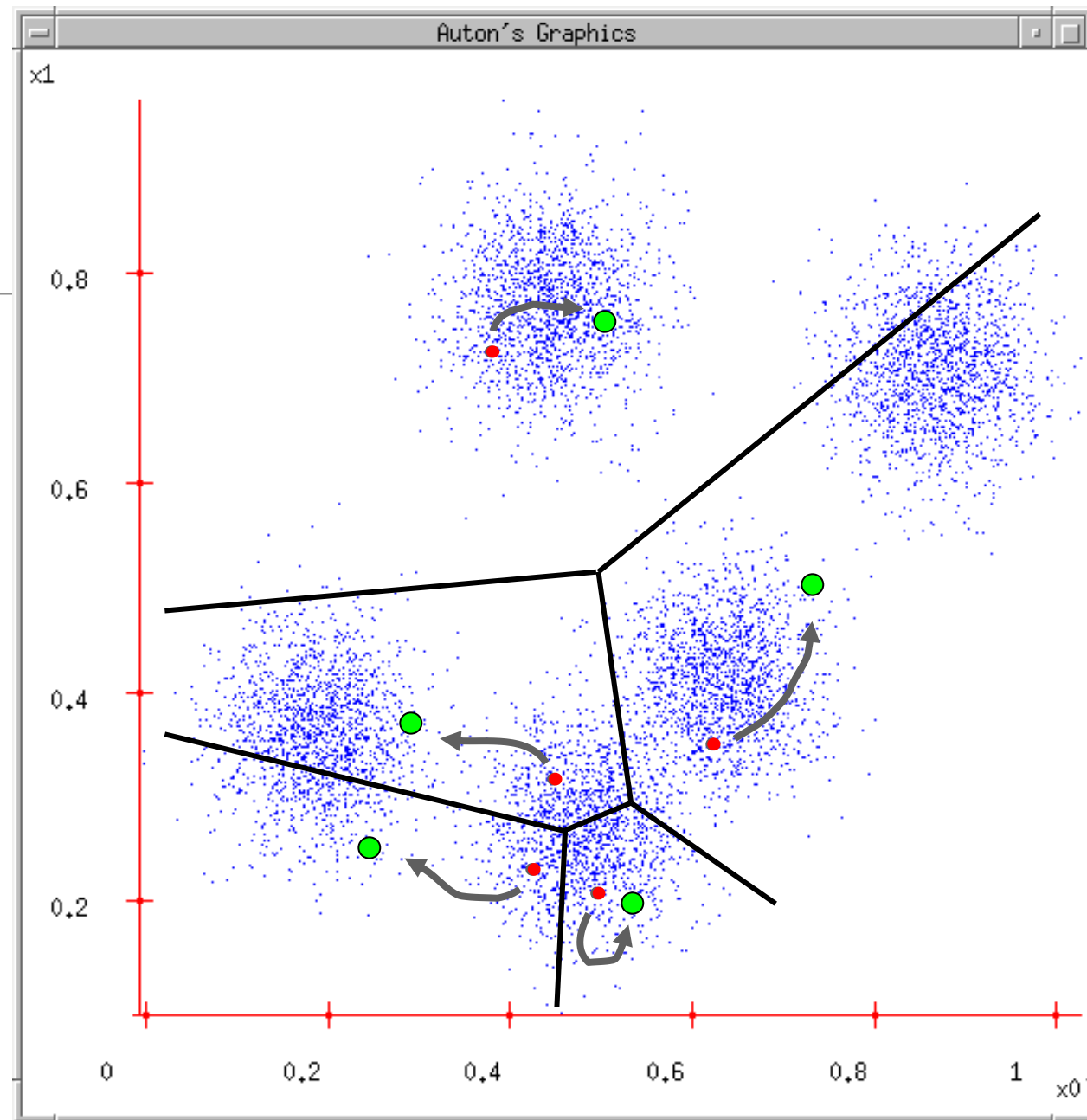
K-means

1. Ask user how many clusters they'd like.
(e.g. $k=5$)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



K-means

1. Ask user how many clusters they'd like.
(e.g. $k=5$)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns

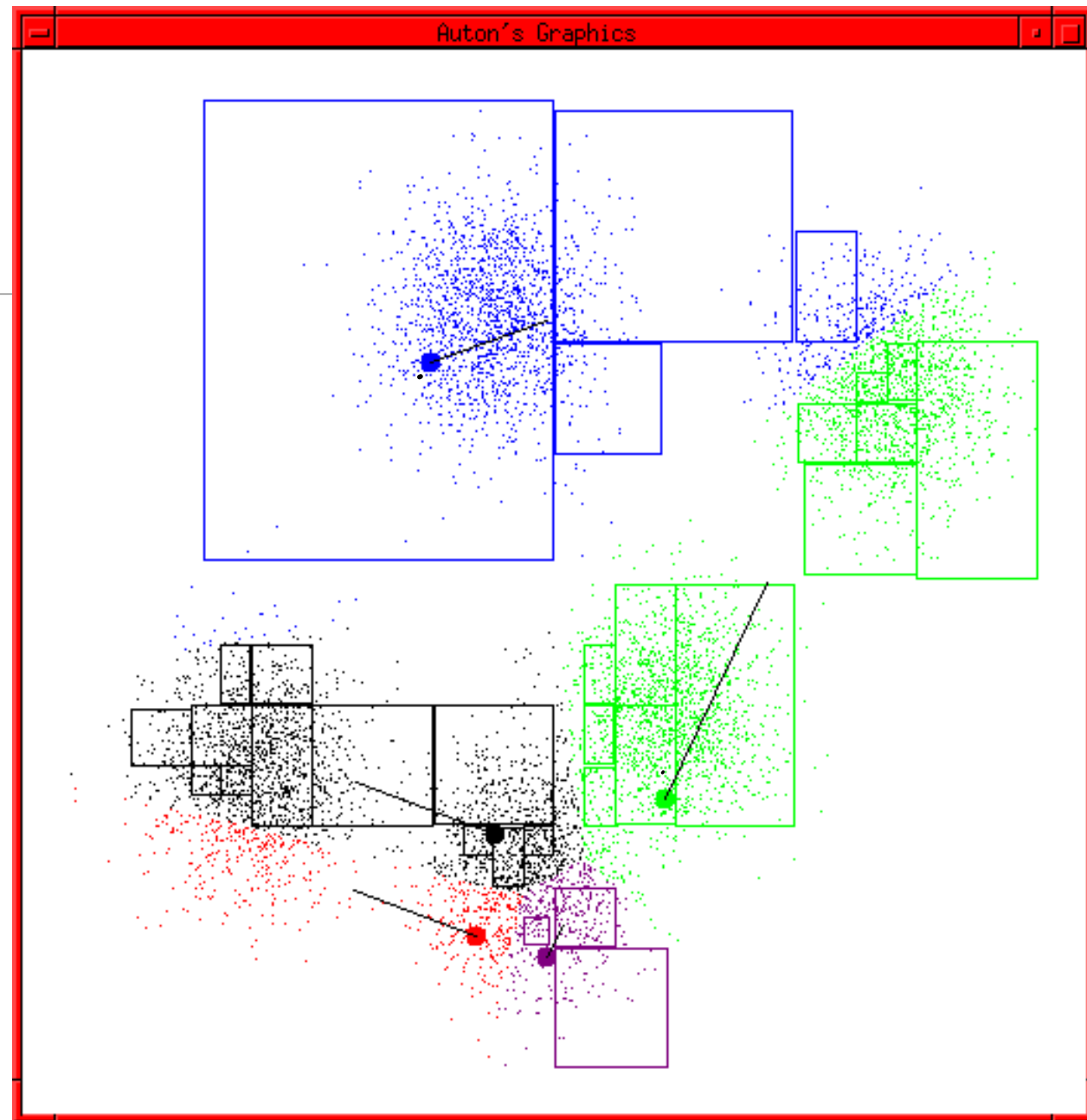


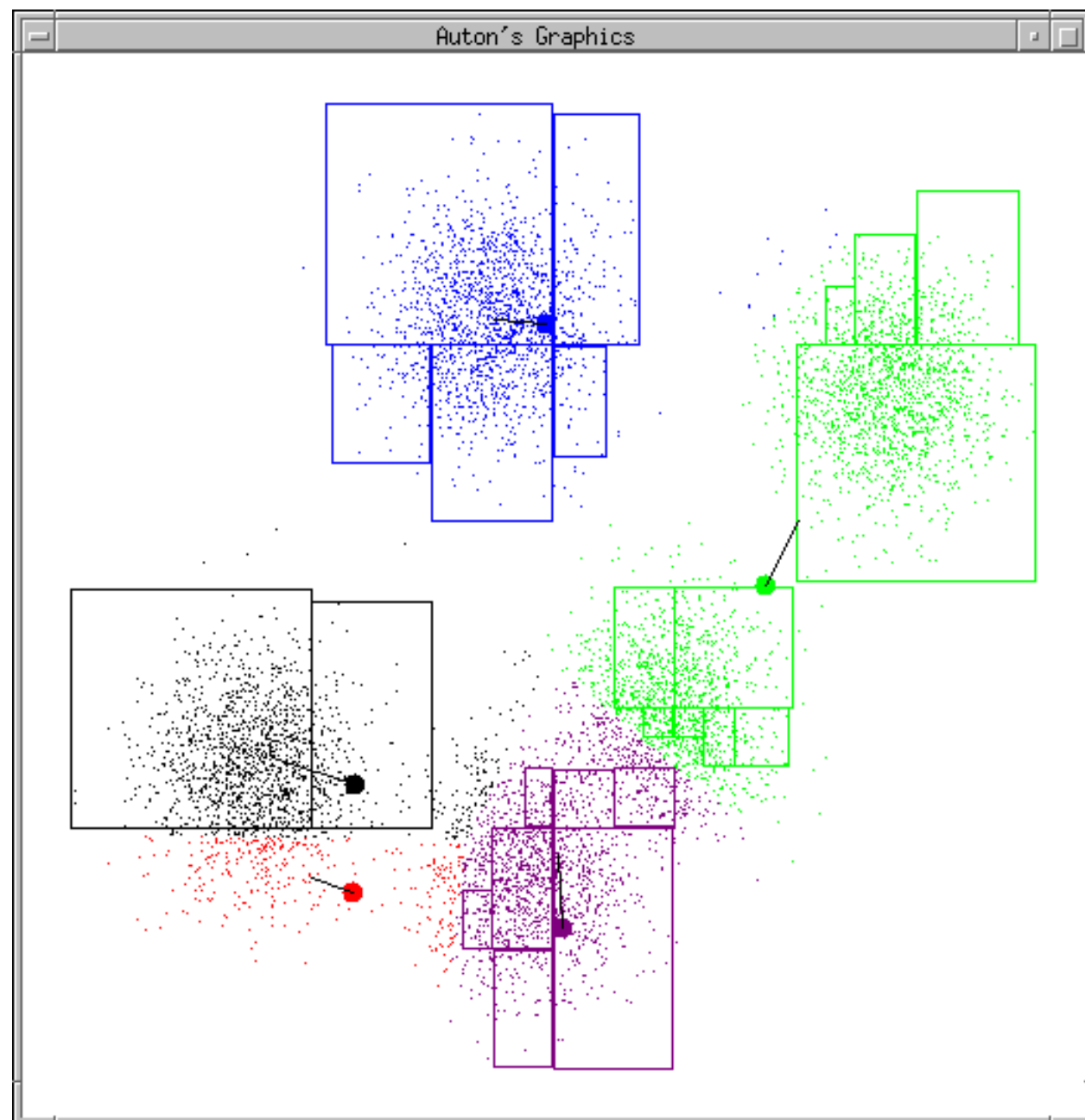
K-means Start

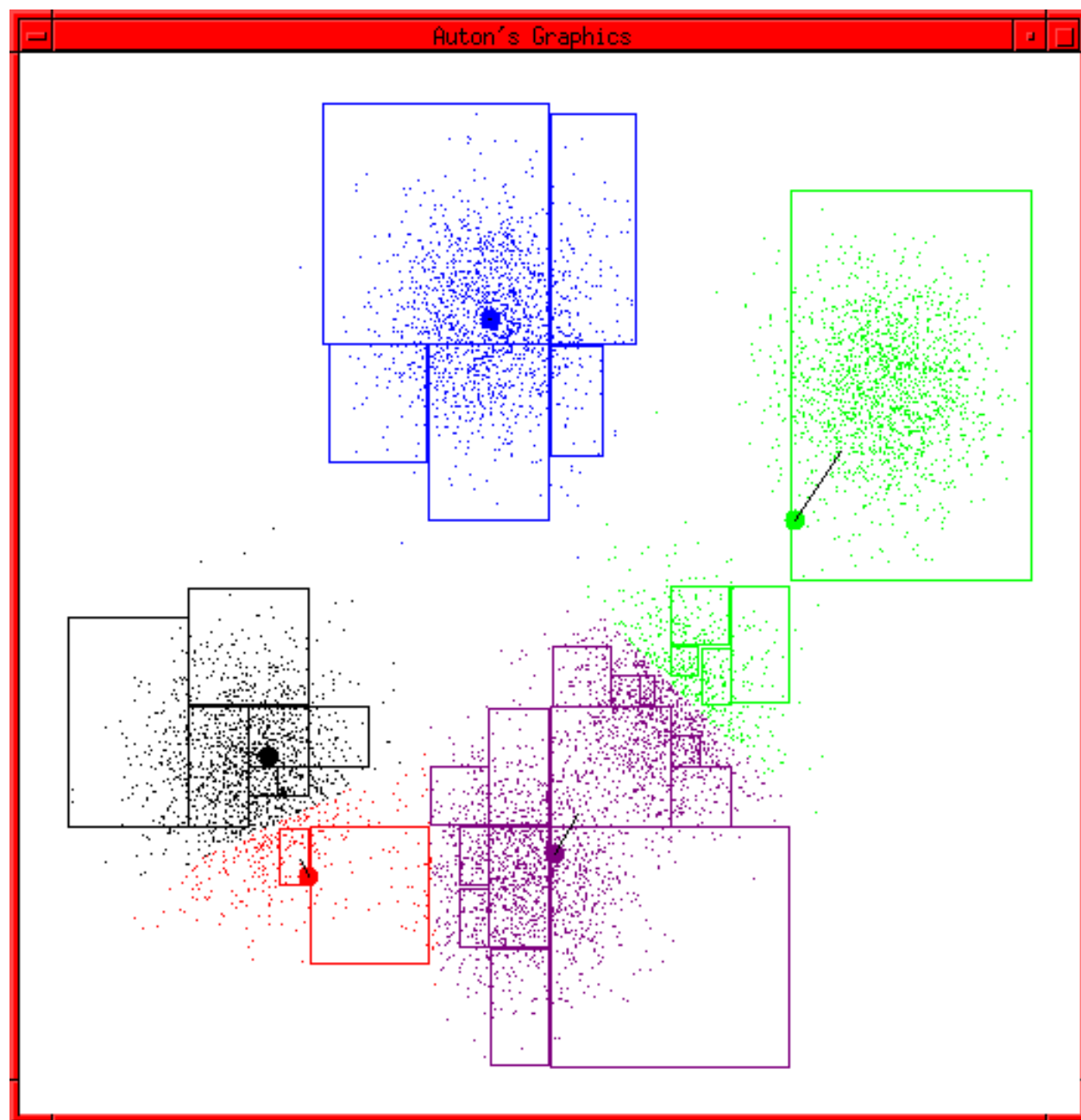
Advance apologies: in
Black and White this
example will deteriorate

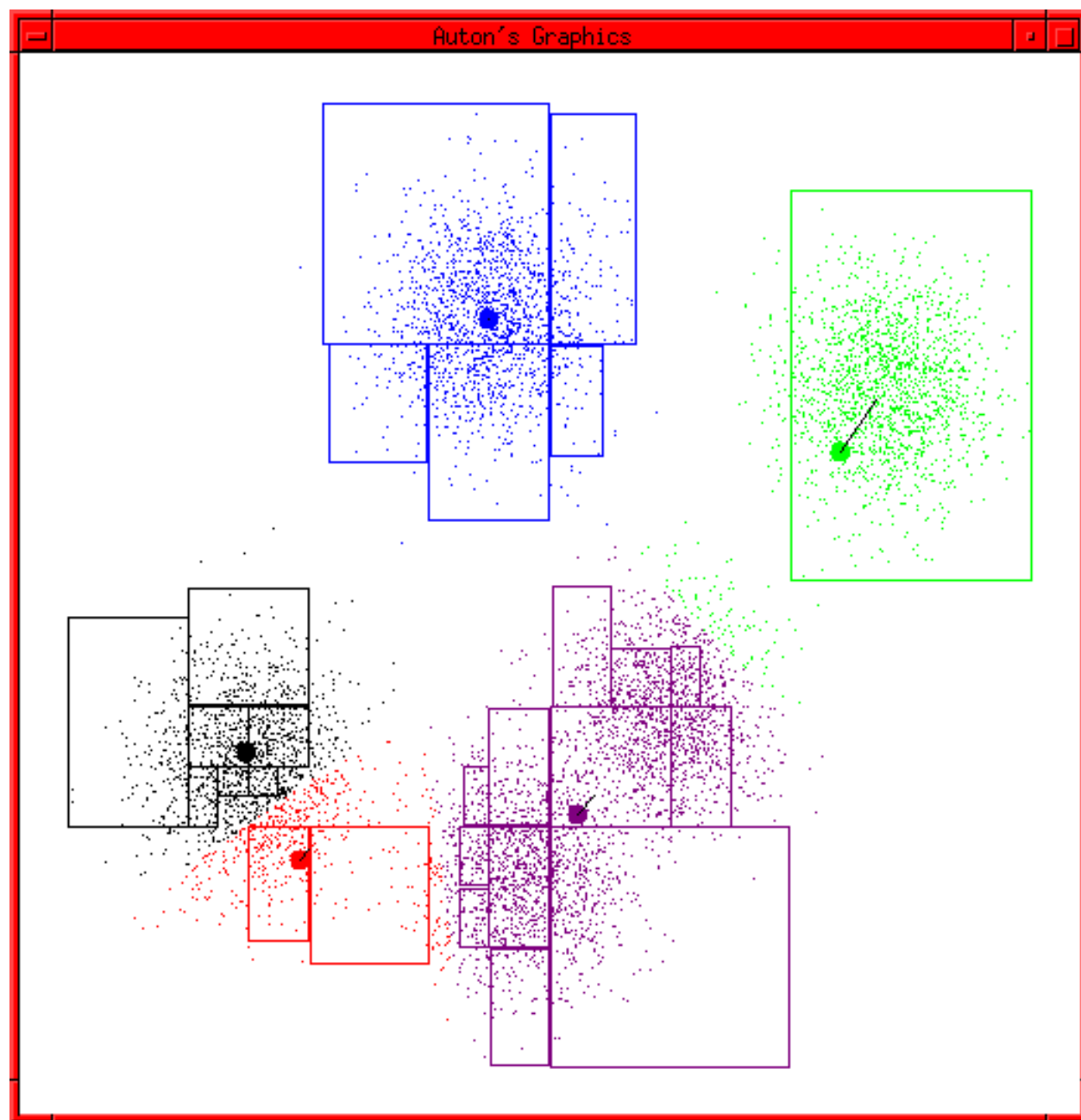
Example generated by
Dan Pelleg's super-duper
fast K-means system:

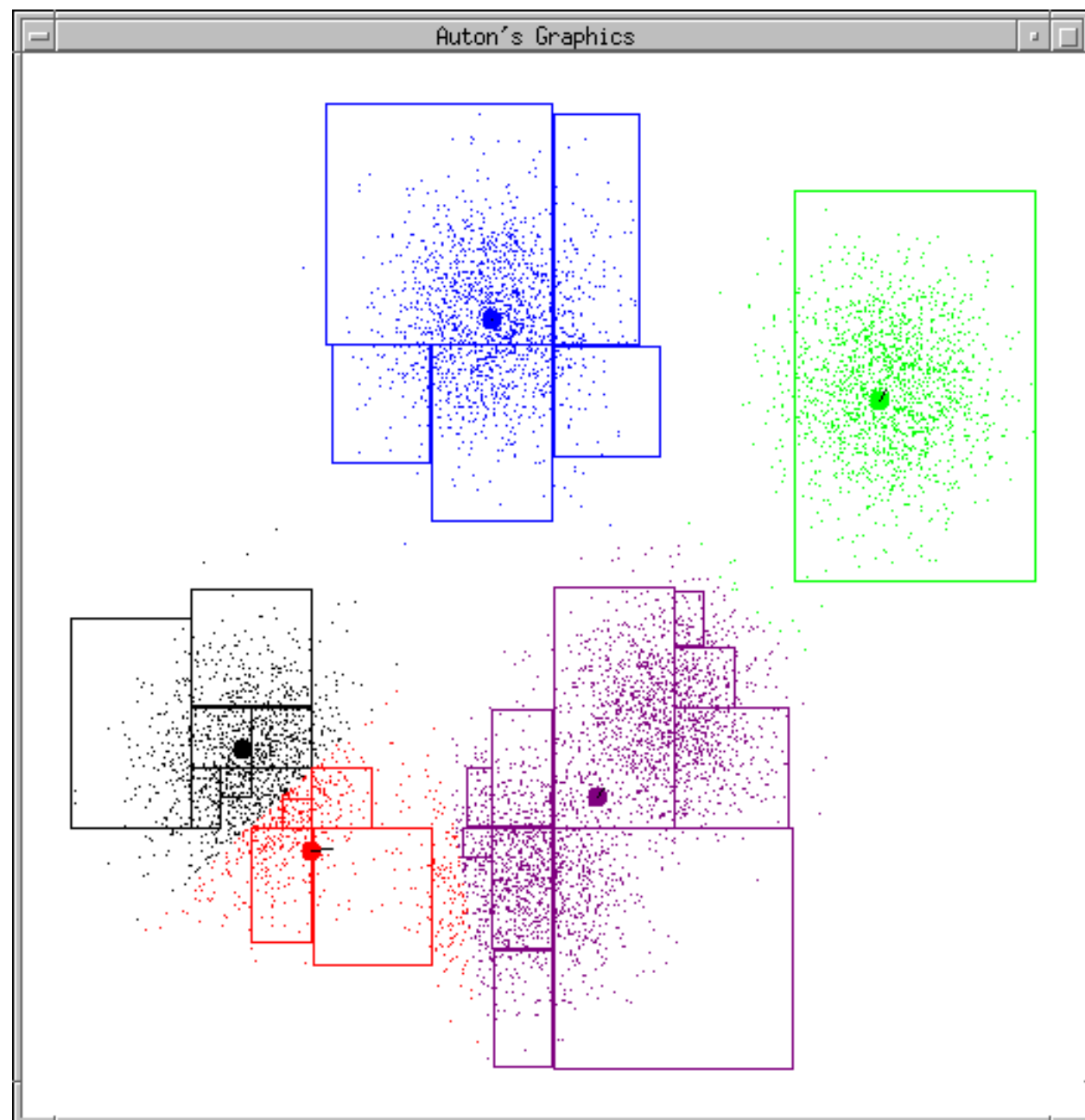
*Dan Pelleg and Andrew
Moore. Accelerating Exact
k-means Algorithms with
Geometric Reasoning.
Proc. Conference on
Knowledge Discovery in
Databases 1999,
(KDD99) (available on
www.autonlab.org/pap.html)*

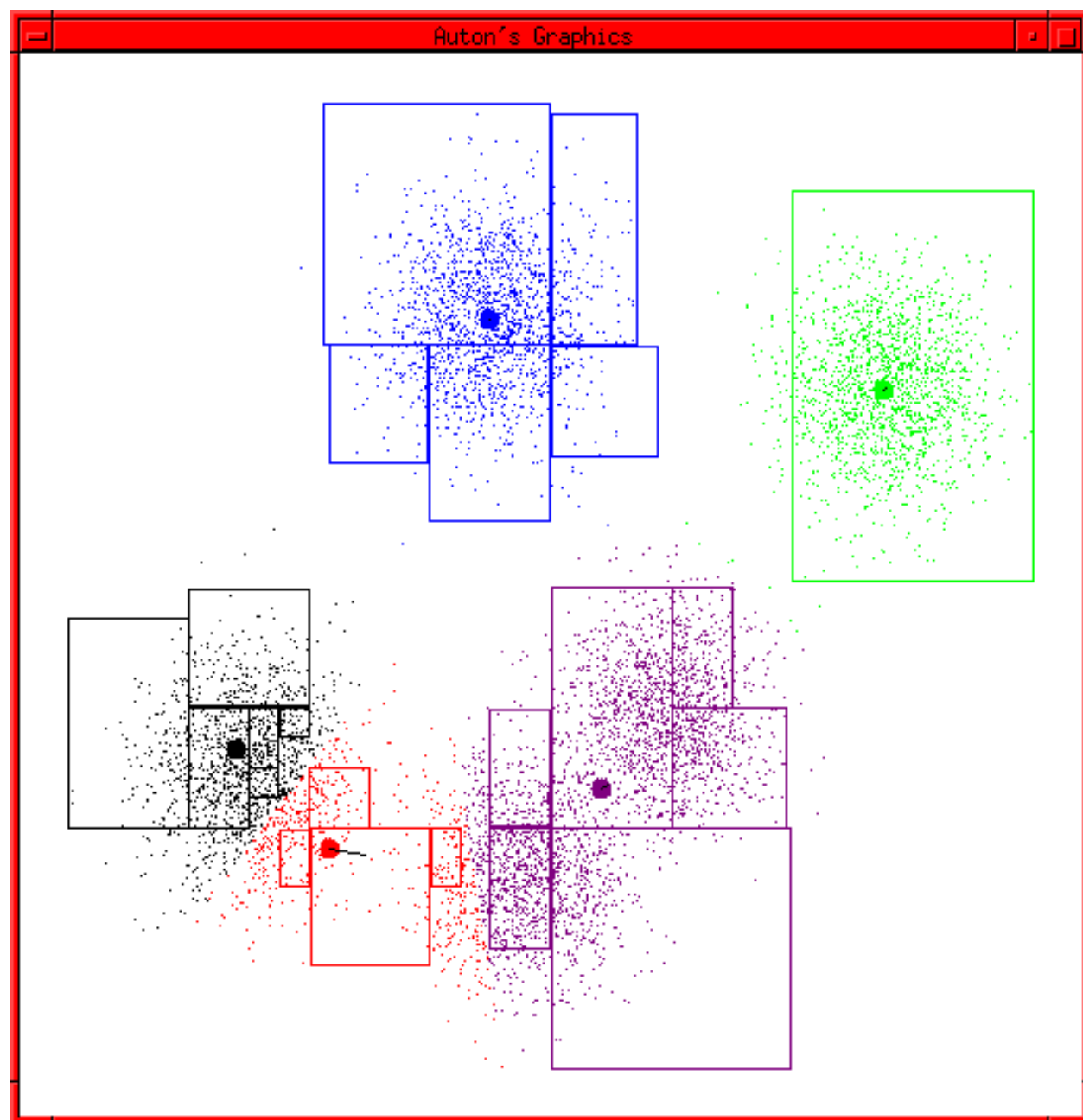


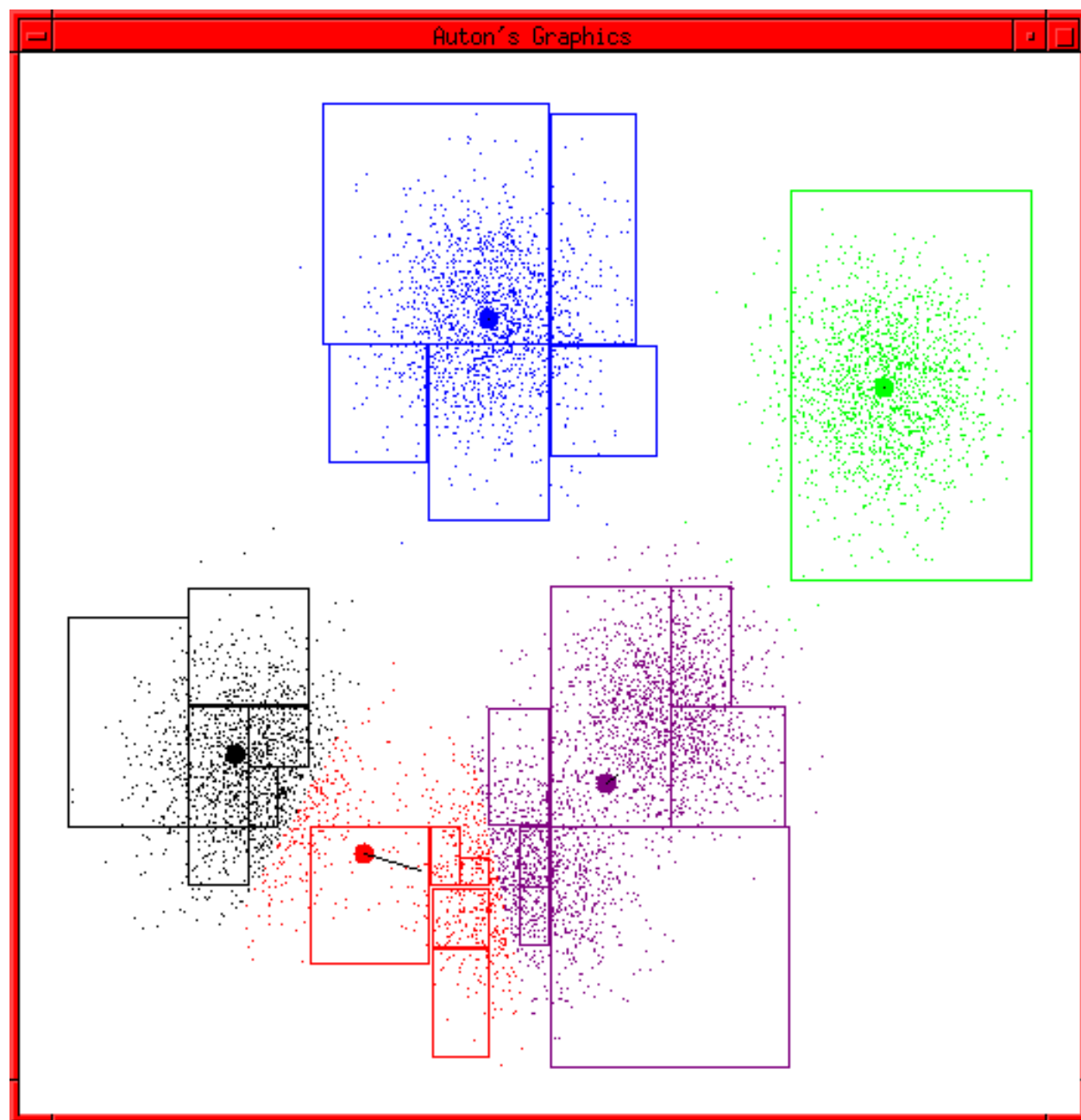


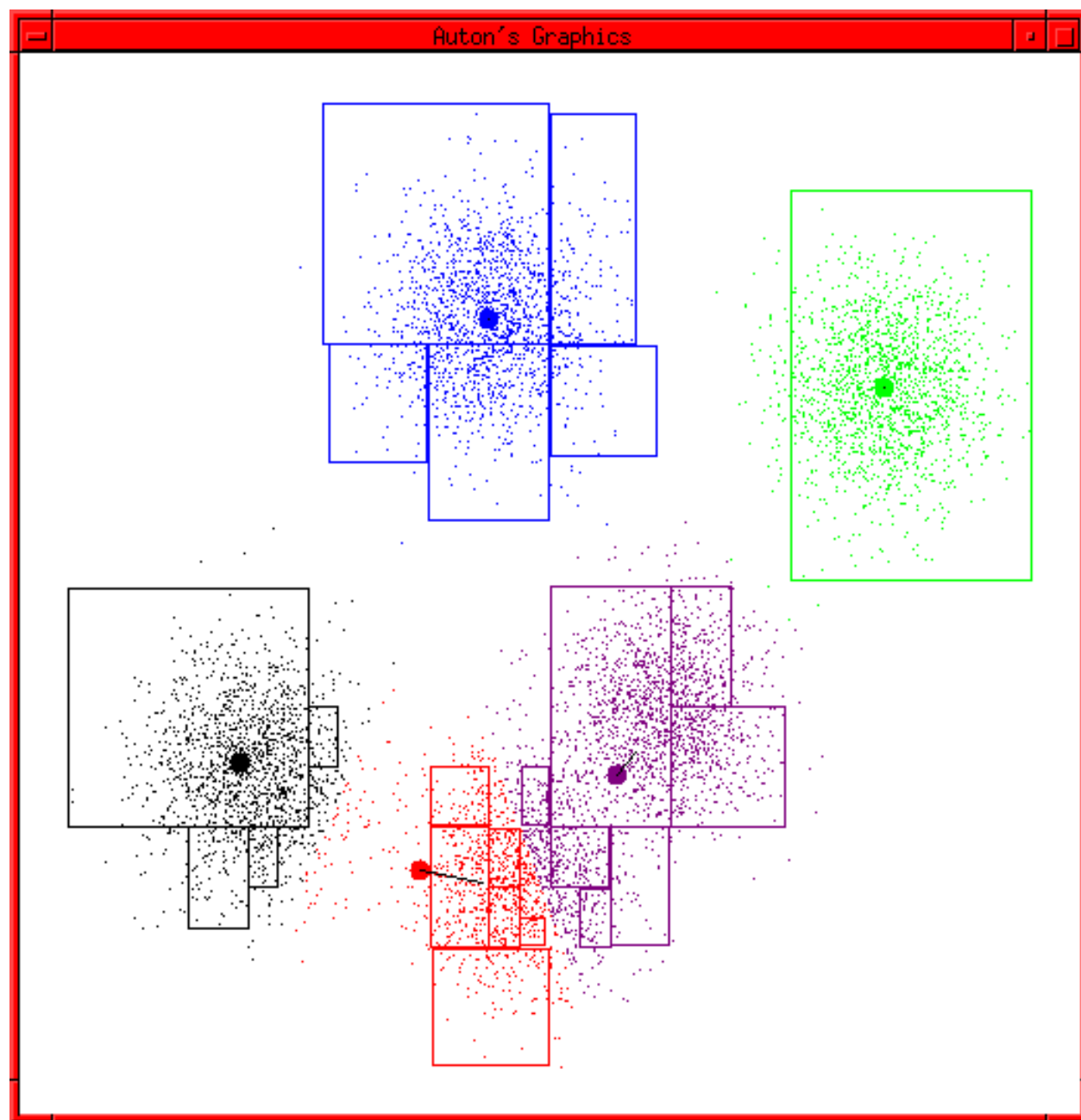


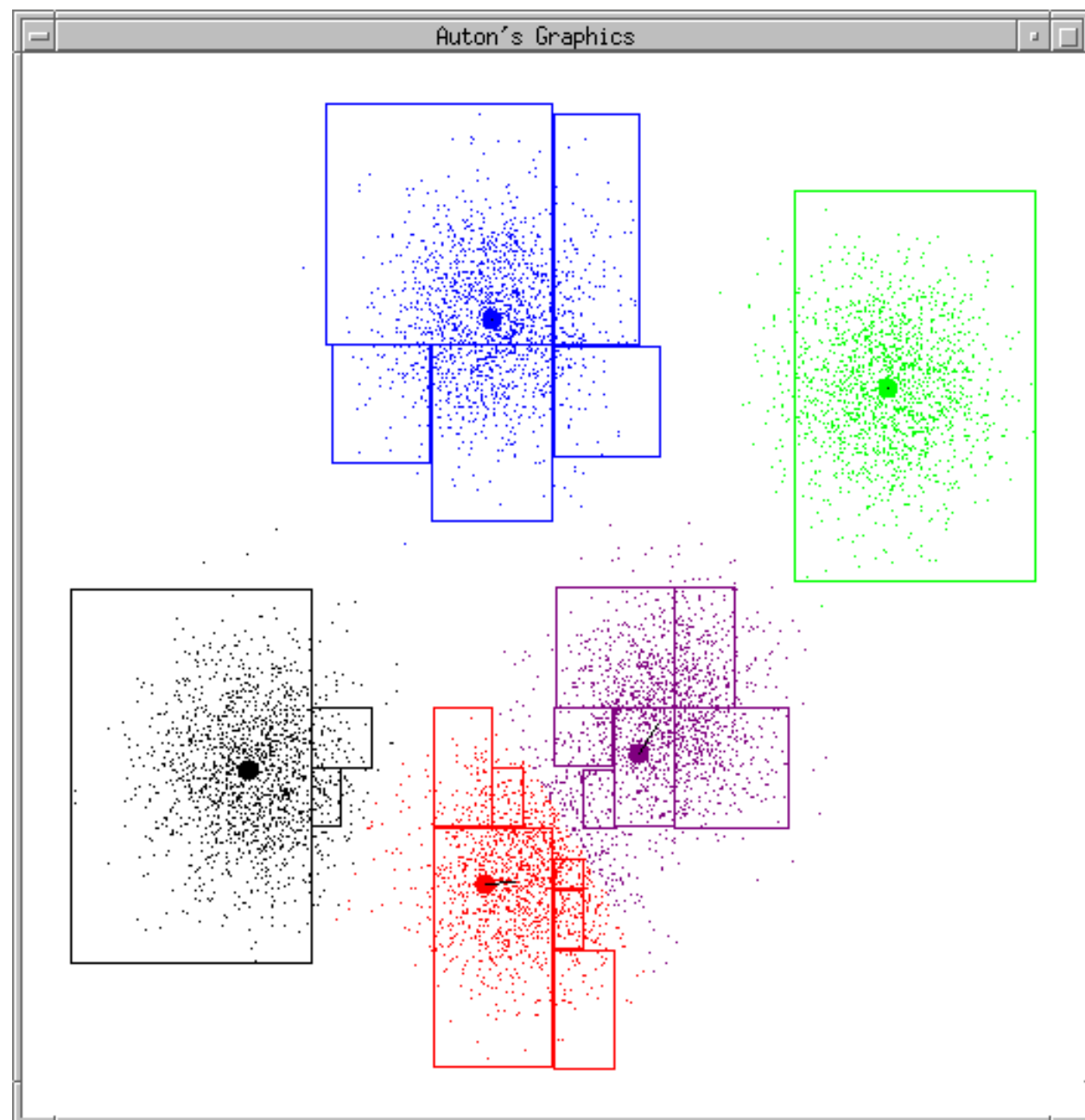






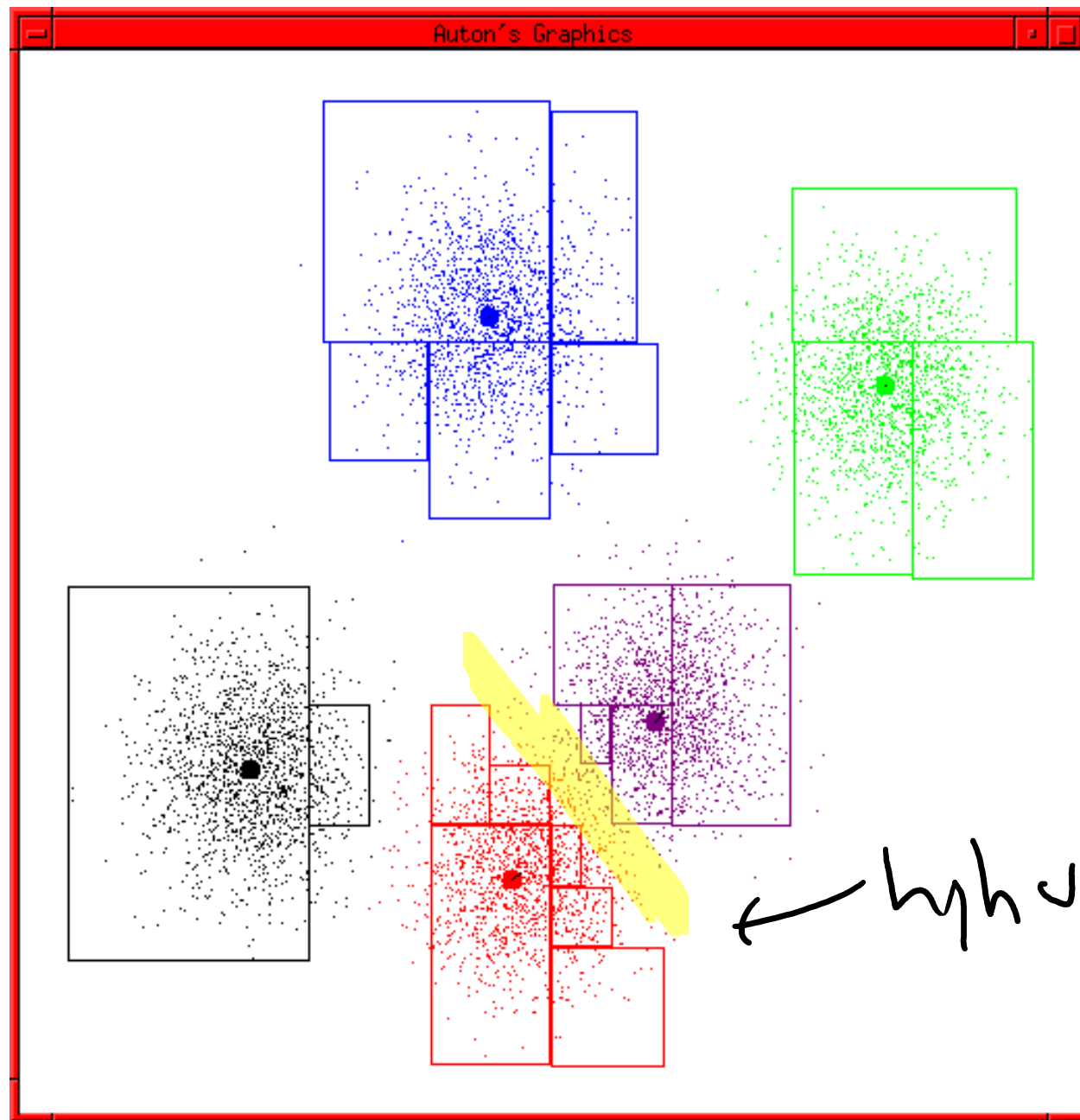






What
Do these
boxes represent?

Boxes are
regions of
high certainty



high uncertainty

Choosing k


Defined by the application, e.g., color quantization

Plot data (Projection to low dimension) and check for clusters


Incremental (leader-cluster) algorithm: Add one at a time until “elbow” (reconstruction error/log likelihood/intergroup distances)

Manually check for meaning

check multiple
values of k



intra
cluster
inter
cluster



Issues (cont'd)

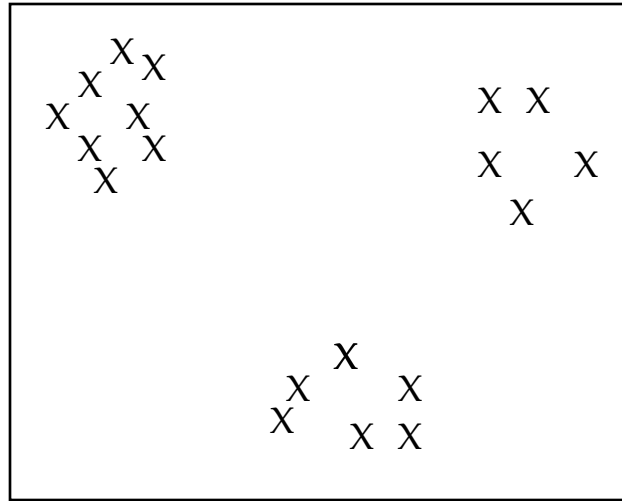
How to determine k ?

One can try different values for k until the smallest k such that increasing k does not much decrease the average ~~points~~ distance of points to their centroids.

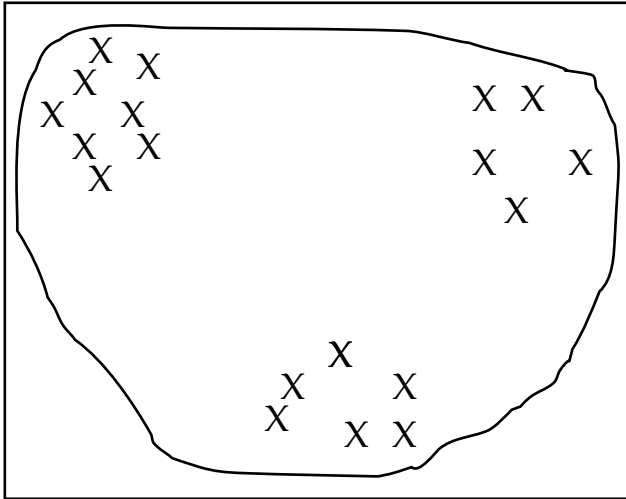
distance

← define our metric

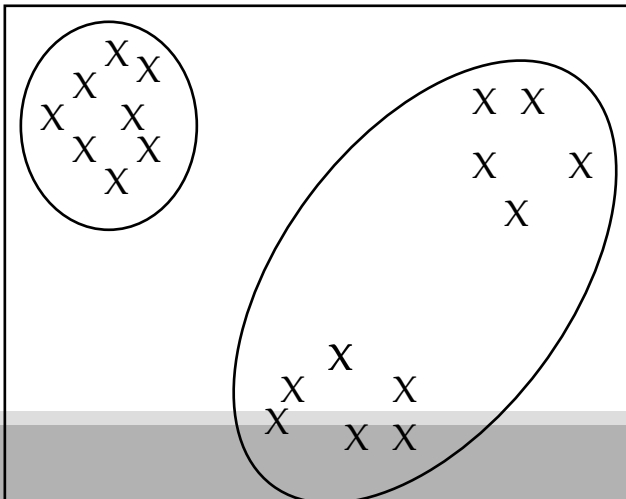
k



Determining k

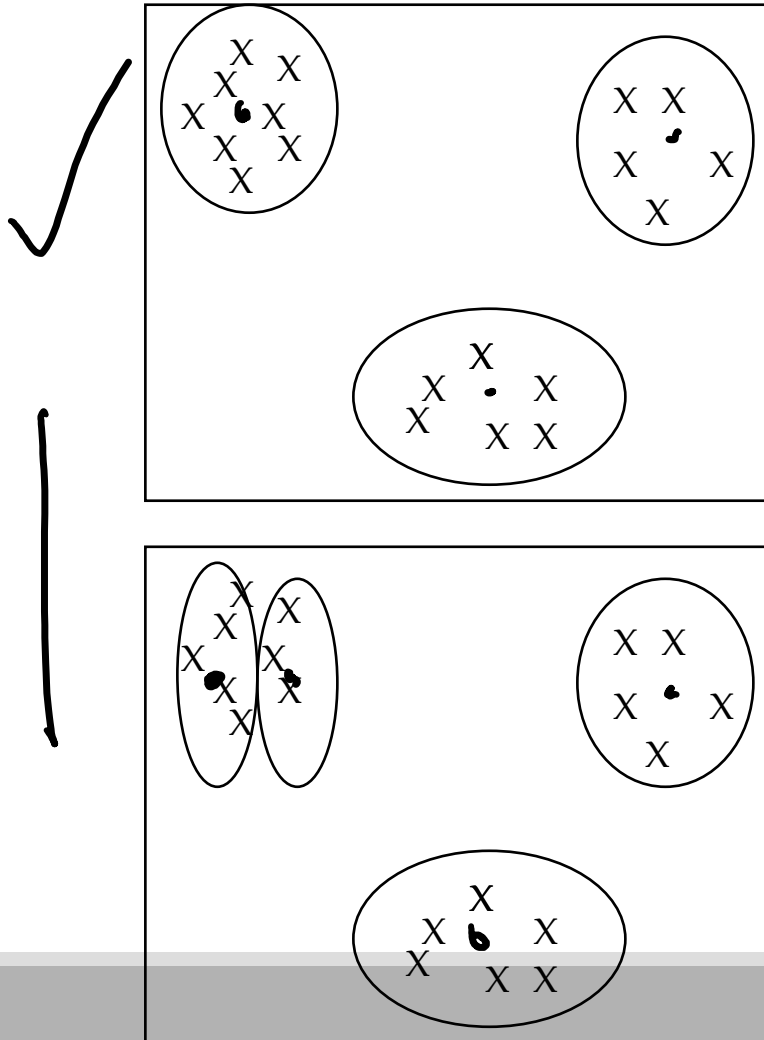


When $k = 1$, all the points are in one cluster, and the average distance to the centroid will be high.



When $k = 2$, one of the clusters will be by itself and the other two will be forced into one cluster. The average distance of points to the centroid will shrink considerably.

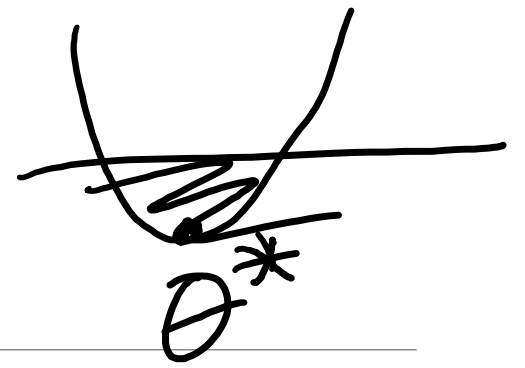
Determining k (cont'd)



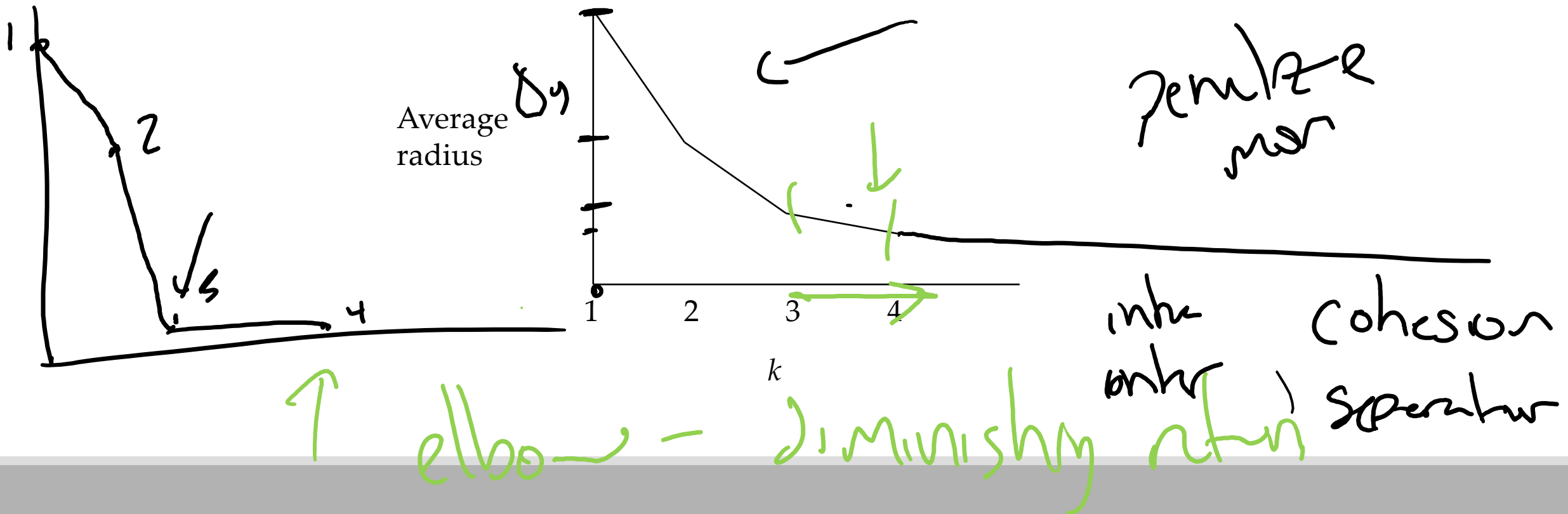
When $k = 3$, each of the apparent clusters should be a cluster by itself, and the average distance from the points to their centroids shrinks again.

When $k = 4$, then one of the true clusters will be artificially partitioned into two nearby clusters. The average distance to the centroids will drop a bit, but not much.

Determining k (cont'd)



This failure to drop further suggests that $k = 3$ is right. This conclusion can be made even if the data is in so many dimensions that we cannot visualize the clusters.



Holes in data space

All the clustering algorithms only group data.

Clusters only represent one aspect of the knowledge in the data.

Another aspect that we have not studied is the holes.

- A hole is a region in the data space that contains no or few data points. Reasons:
 - insufficient data in certain areas, and/or
 - certain attribute-value combinations are not possible or seldom occur.



Holes are useful too

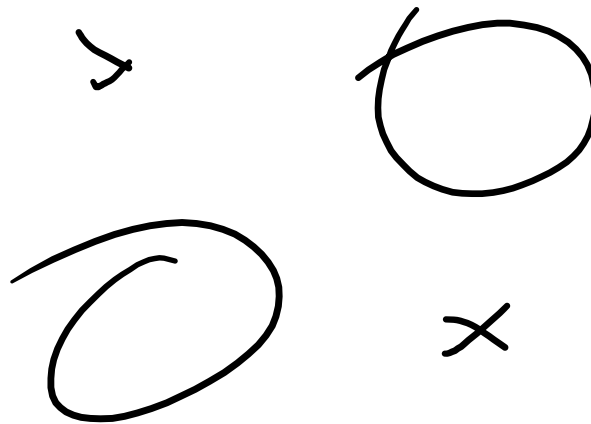
Although clusters are important, holes in the space can be quite useful too.

For example, in a disease database

- we may find that certain symptoms and/or test values do not occur together, or
- when a certain medicine is used, some test values never go beyond certain ranges. ←

Discovery of such information can be important in medical domains because

- it could mean the discovery of a cure to a disease or some biological laws.



Data regions and empty regions

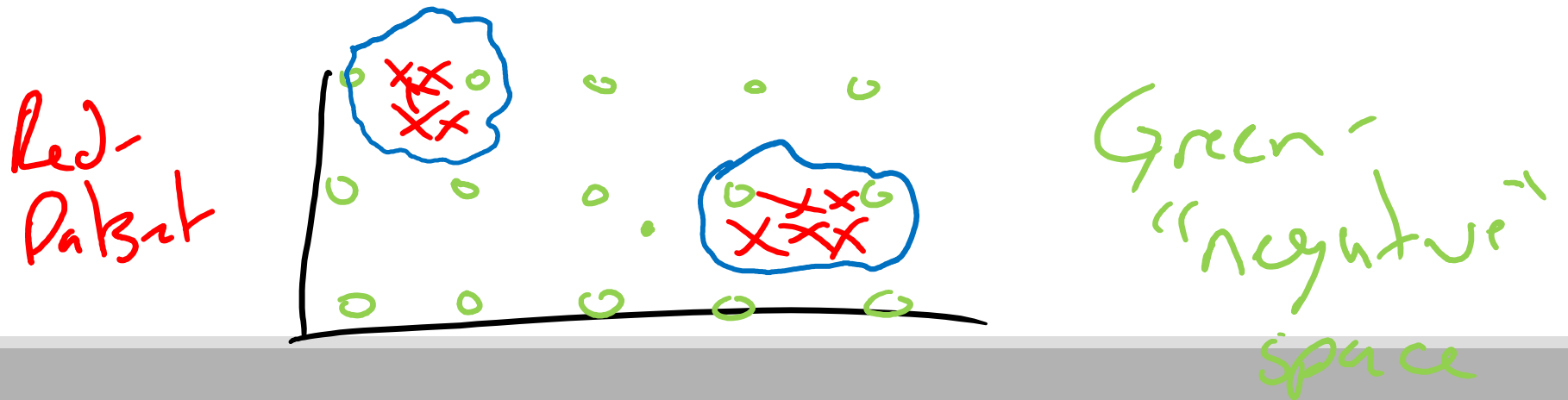
Given a data space, separate

- data regions (clusters) and
- empty regions (holes, with few or no data points).

Use a supervised learning technique, i.e., decision tree induction, to separate the two types of regions.

Due to the use of a supervised learning method for an unsupervised learning task,

- an interesting connection is made between the two types of learning paradigms.



Supervised learning for unsupervised learning

Decision tree algorithm is not directly applicable.

- it needs at least two classes of data.
- A clustering data set has no class label for each data point.

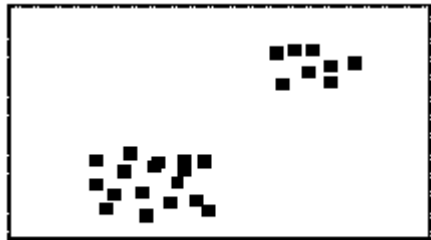
we label it as
Positive

The problem can be dealt with by a simple idea.

- Regard each point in the data set to have a class label Y.
- Assume that the data space is uniformly distributed with another type of points, called non-existing points. We give them the class, N. ←

With the N points added, the problem of partitioning the data space into data and empty regions becomes a supervised classification problem.

An example



(A): The original data space



(B). Partitioning with added
 N points

~~pos~~ pos
neg

A decision tree method is used for partitioning in (B).

Can it done without adding N points?

Yes.

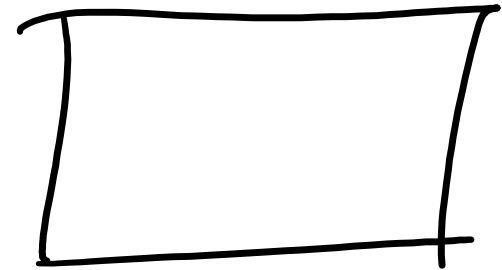
Physically adding N points increases the size of the data and thus the running time.

More importantly: it is unlikely that we can have points truly uniformly distributed in a high dimensional space as we would need an exponential number of points.

Fortunately, no need to physically add any N points.

- We can compute them when needed

(we 'just' treat it as a distribution)



Characteristics of the approach

It provides representations of the resulting data and empty regions in terms of hyper-rectangles, or rules.

uncertainty

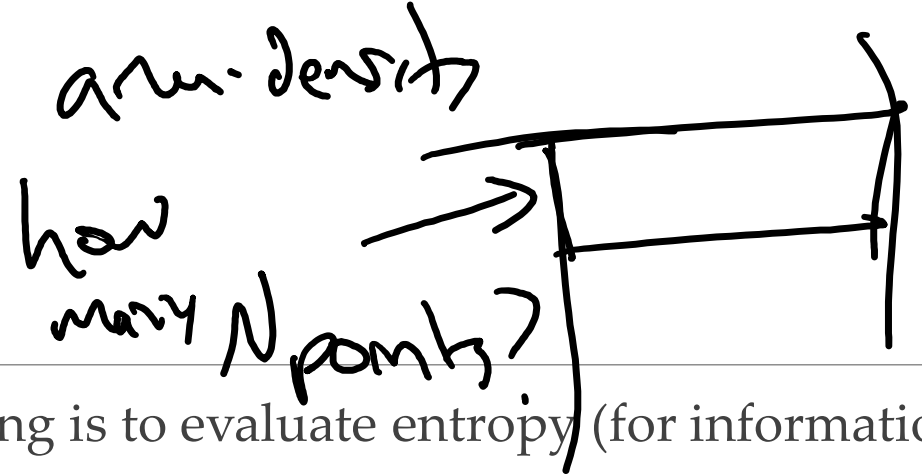
It detects outliers automatically. Outliers are data points in an empty region.

It may not use all attributes in the data just as in a normal decision tree for supervised learning.

- It can automatically determine what attributes are useful. Subspace clustering ...

Drawback: data regions of irregular shapes are hard to handle since decision tree learning only generates hyper-rectangles (formed by axis-parallel hyper-planes), which are rules.

Building the Tree



The main computation in decision tree building is to evaluate entropy (for information gain):

$$\text{entropy}(D) = - \sum_{j=1}^{|C|} \text{Pr}(c_j) \log_2 \text{Pr}(c_j)$$

only need the count

Can it be evaluated without adding N points? Yes.

$\text{Pr}(c_j)$ is the probability of class c_j in data set D , and $|C|$ is the number of classes, Y and N (2 classes).

- To compute $\text{Pr}(c_j)$, we only need the number of Y (data) points and the number of N (non-existing) points.
- We already have Y (or data) points, and we can compute the number of N points on the fly. Simple: as we assume that the N points are uniformly distributed in the space.

An example

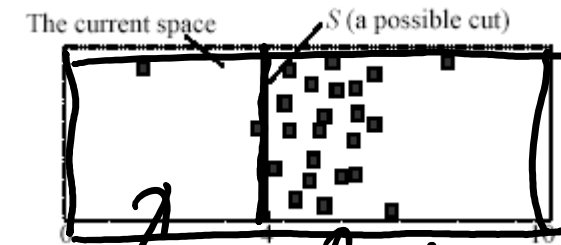
The space has 25 data (Y) points and 25 N points. Assume the system is evaluating a possible cut S.

- # N points on the left of S is $25 * 4/10 = 10$. The number of Y points is 3.
- Likewise, # N points on the right of S is 15 ($= 25 - 10$). The number of Y points is 22.

With these numbers, entropy can be computed.

overfitting
need to prune
in order to
terminate

40 60



$$\sum_{pos} = \sum N_y = 18$$

$$P=2 \quad P=16$$

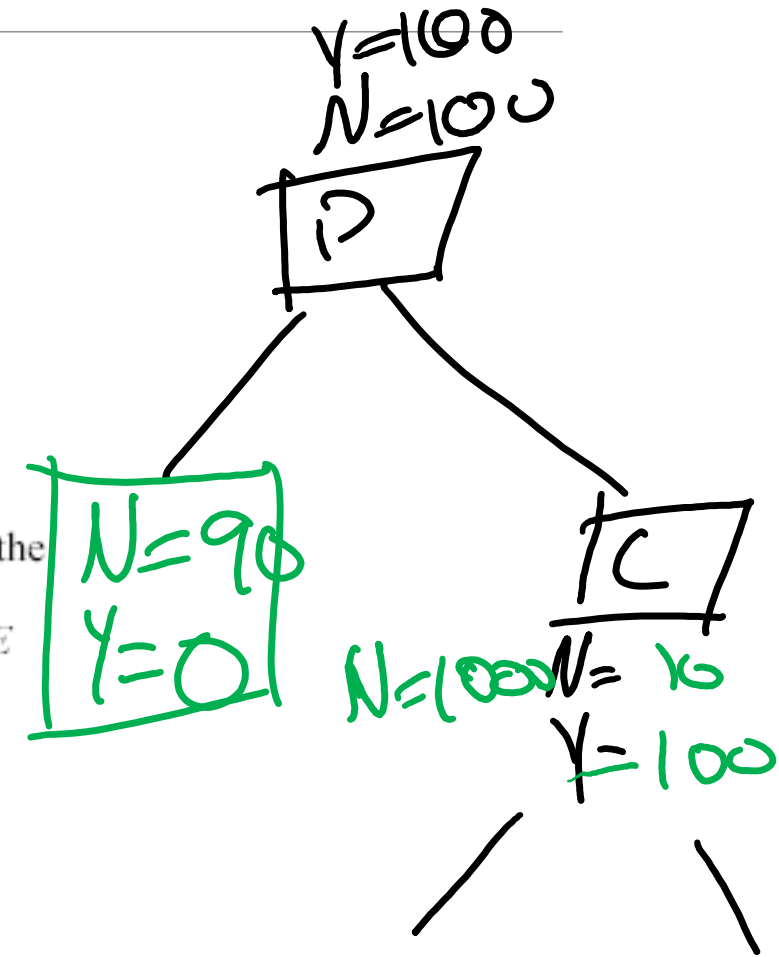
$$N_L = 10 \cdot 0.4 \quad N_R = 18 - 0.6$$

How many N points to add?

We add a different number of N points at each different node.

- The number of N points for the current node E is determined by the following rule (note that at the root node, the number of inherited N points is 0):

- 1 **If** the number of N points inherited from the parent node of E is less than the number of Y points in E **then**
- 2 the number of N points for E is increased to the number of Y points in E
- 3 **else** the number of inherited N points is used for E



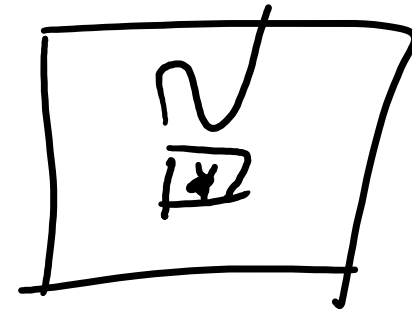
How many N points to add? (cont...)

Basically, for a Y node (which has more data points), we increase N points so that

$$\#Y = \#N$$

The number of N points is not reduced if the current node is an N node (an N node has more N points than Y points).

- A reduction may cause outlier Y points to form Y nodes (a Y node has an equal number of Y points as N points or more).
- Then data regions and empty regions may not be separated well.



Building the decision tree

Using the above ideas, a decision tree can be built to separate data regions and empty regions.

The actual method is more sophisticated as a few other tricky issues need to be handled in

- tree building and
- tree pruning.

else all positive points will have
a cluster
more pruned = more outliers

Summary

Clustering is has along history and still active

- There are a huge number of clustering algorithms
- More are still coming every year.

We only introduced several main algorithms. There are many others, e.g.,

- density based algorithm, sub-space clustering, scale-up methods, neural networks based methods, fuzzy clustering, co-clustering, etc.

Clustering is hard to evaluate, but very useful in practice. This partially explains why there are still a large number of clustering algorithms being devised every year.

Clustering is highly application dependent and to some extent subjective.

Semi-supervised learning

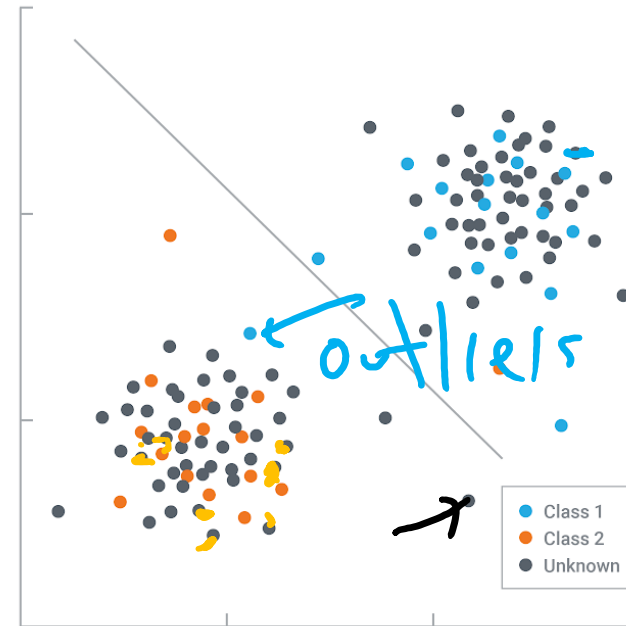
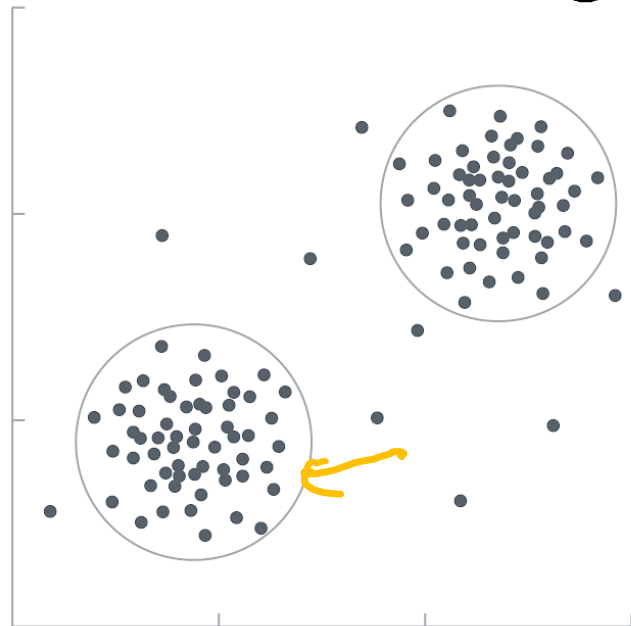
E_{CV} only
calculates
on labeled
points

any clustering alg semi

UNSUPERVISED

SUPERVISED

no
label



Some
(label).

outliers will
have big impact

form clusters. then class. f.
the cluster

Resources: Journals

Journal of Machine Learning Research www.jmlr.org

Machine Learning

IEEE Transactions on Neural Networks

IEEE Transactions on Pattern Analysis and Machine Intelligence

Annals of Statistics

Journal of the American Statistical Association

...

Resources: Conferences

International Conference on Machine Learning (ICML)

European Conference on Machine Learning (ECML)

Neural Information Processing Systems (NIPS)

Computational Learning

International Joint Conference on Artificial Intelligence (IJCAI)

ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)

IEEE Int. Conf. on Data Mining (ICDM)