Clustering

CS 444

Some material on these is slides borrowed from Andrew Moore's machine learning tutorials located at:

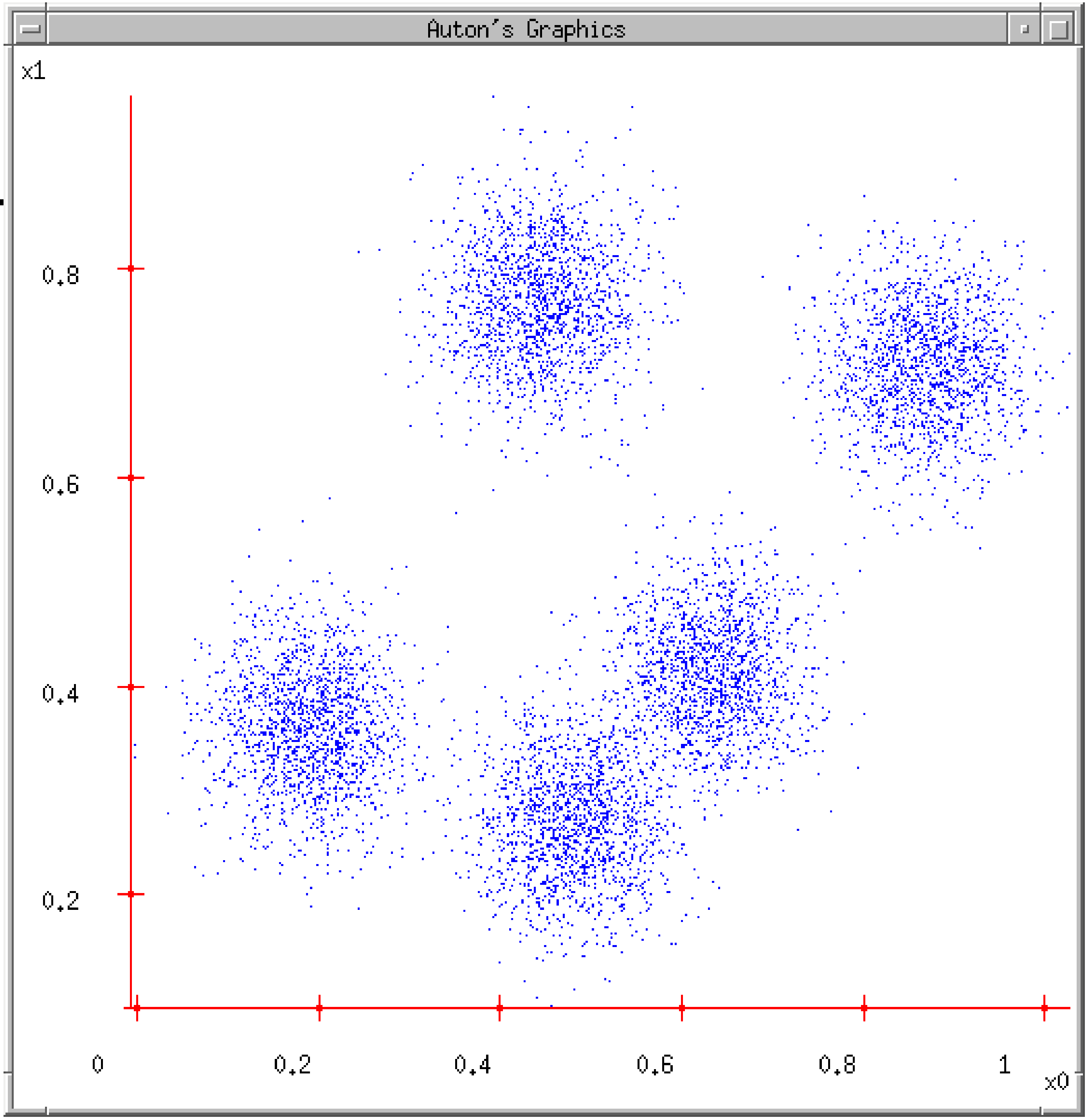
http://www.cs.cmu.edu/~awm/tutorials/

Clustering

- The problem of grouping unlabeled data on the basis of similarity.
- A key component of data mining – is there useful structure hidden in this data?
- Applications:
  - Image segmentation, document clustering, protein class discovery, compression

K-means

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

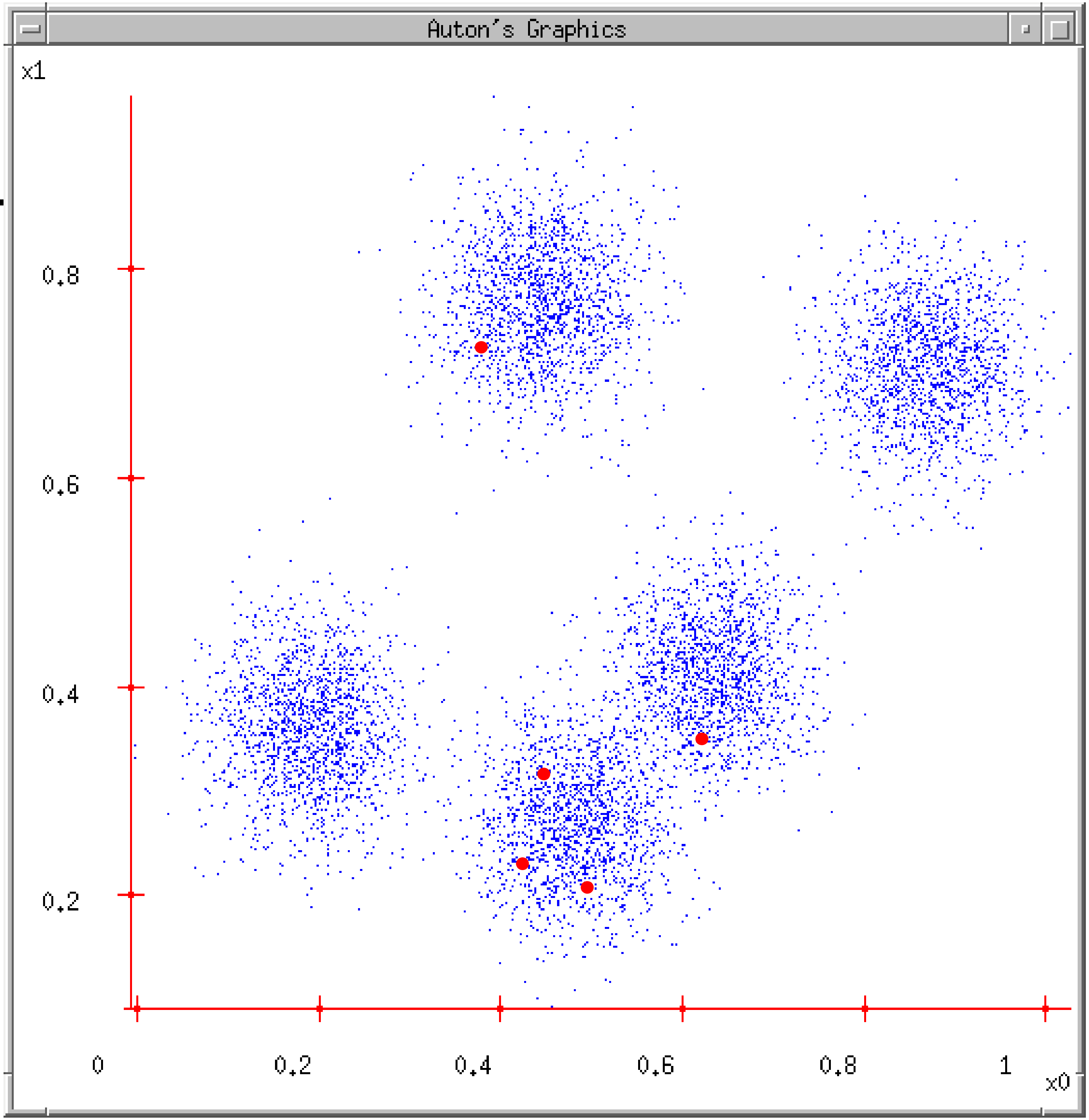


http://www.cs.cmu.edu/~awm/tutorials/

K-means

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

Randomly guess k cluster Center locations



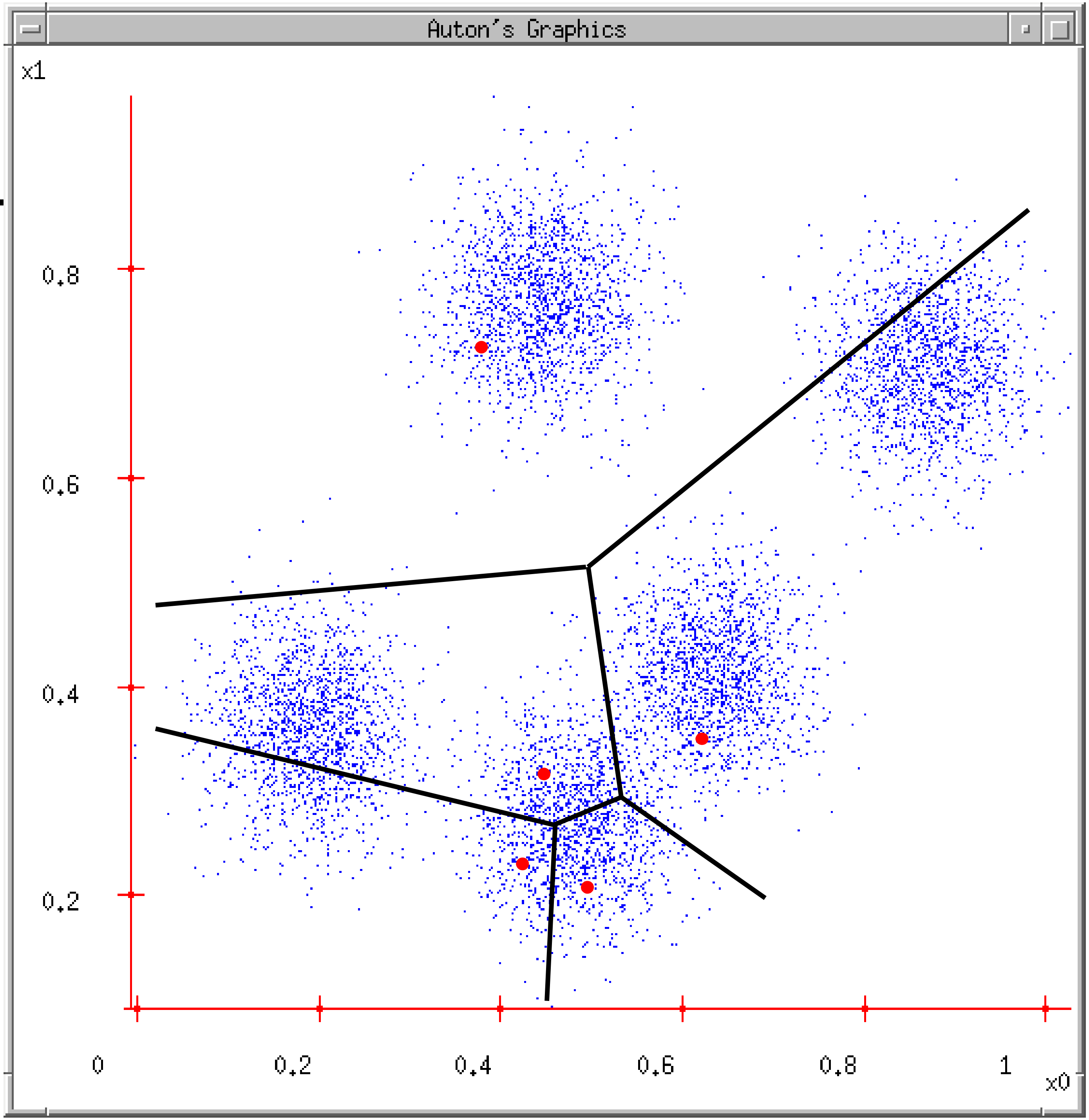
http://www.cs.cmu.edu/~awm/tutorials/

K-means

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

Randomly guess k cluster Center locations

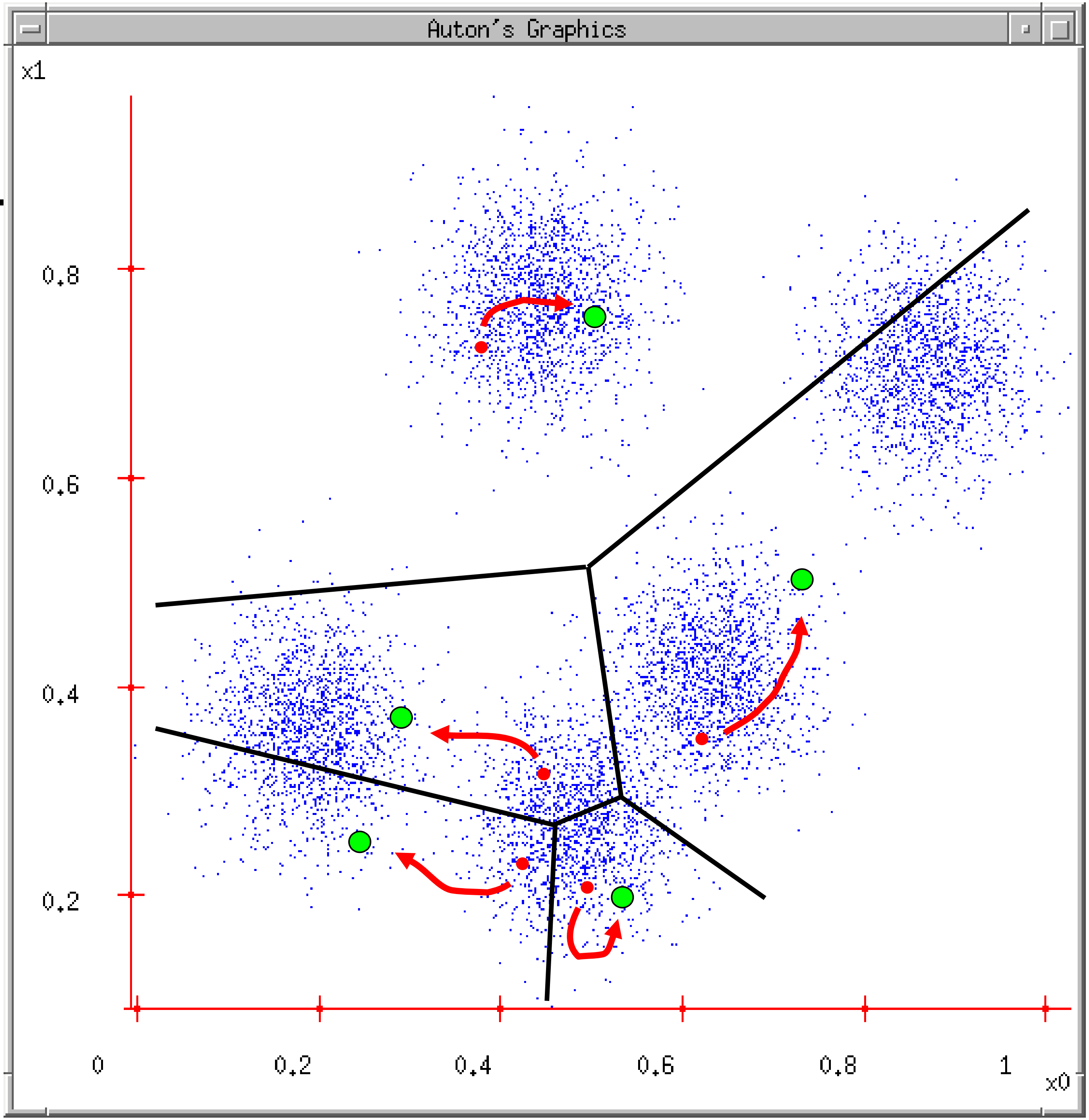
Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



http://www.cs.cmu.edu/~awm/tutorials/

K-means

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



http://www.cs.cmu.edu/~awm/tutorials/

K-means

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

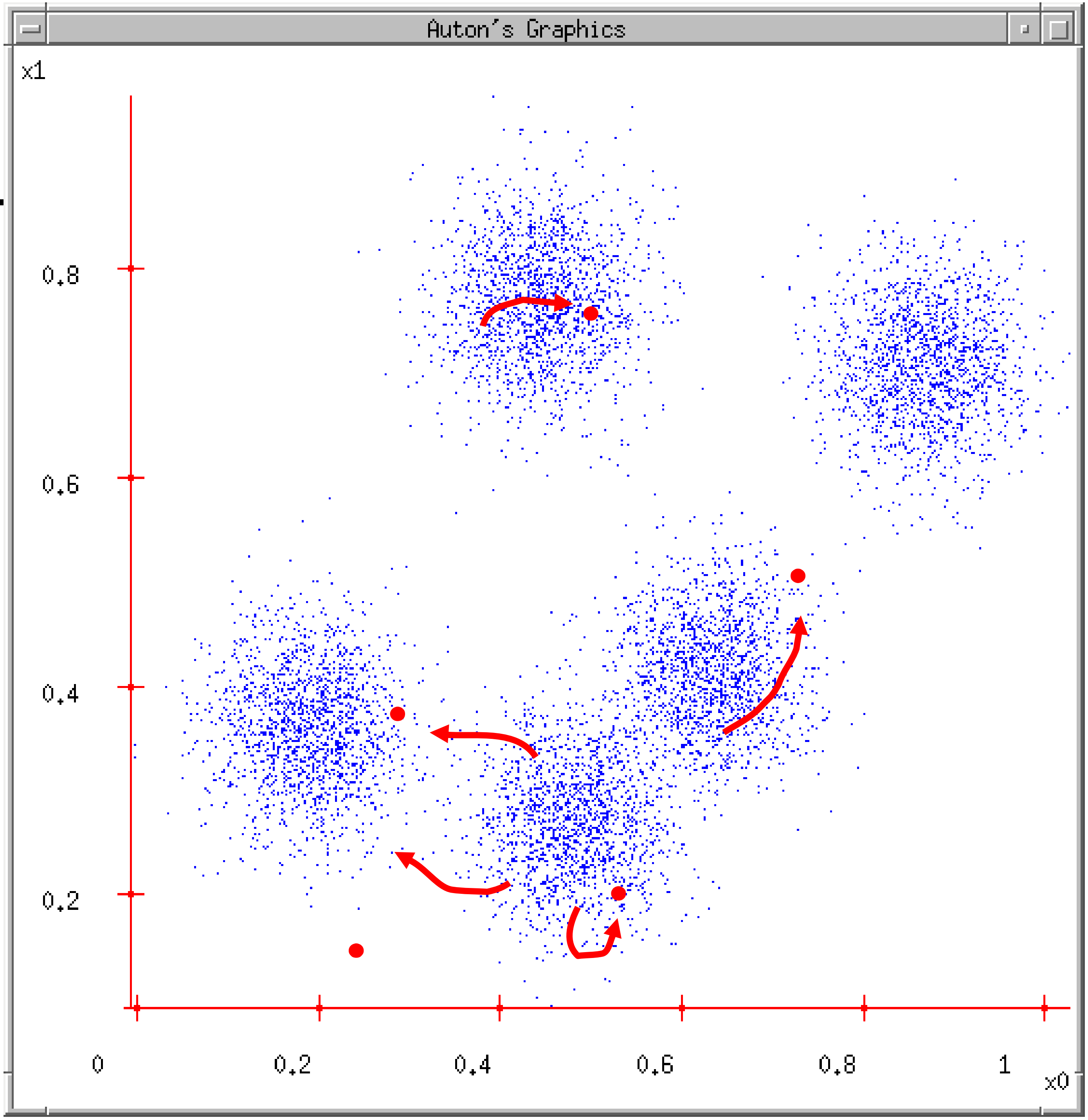
Randomly guess k cluster Center locations

Each datapoint finds out which Center it's closest to.

Each Center finds the centroid of the points it owns…

…and jumps there

…Repeat until terminated!



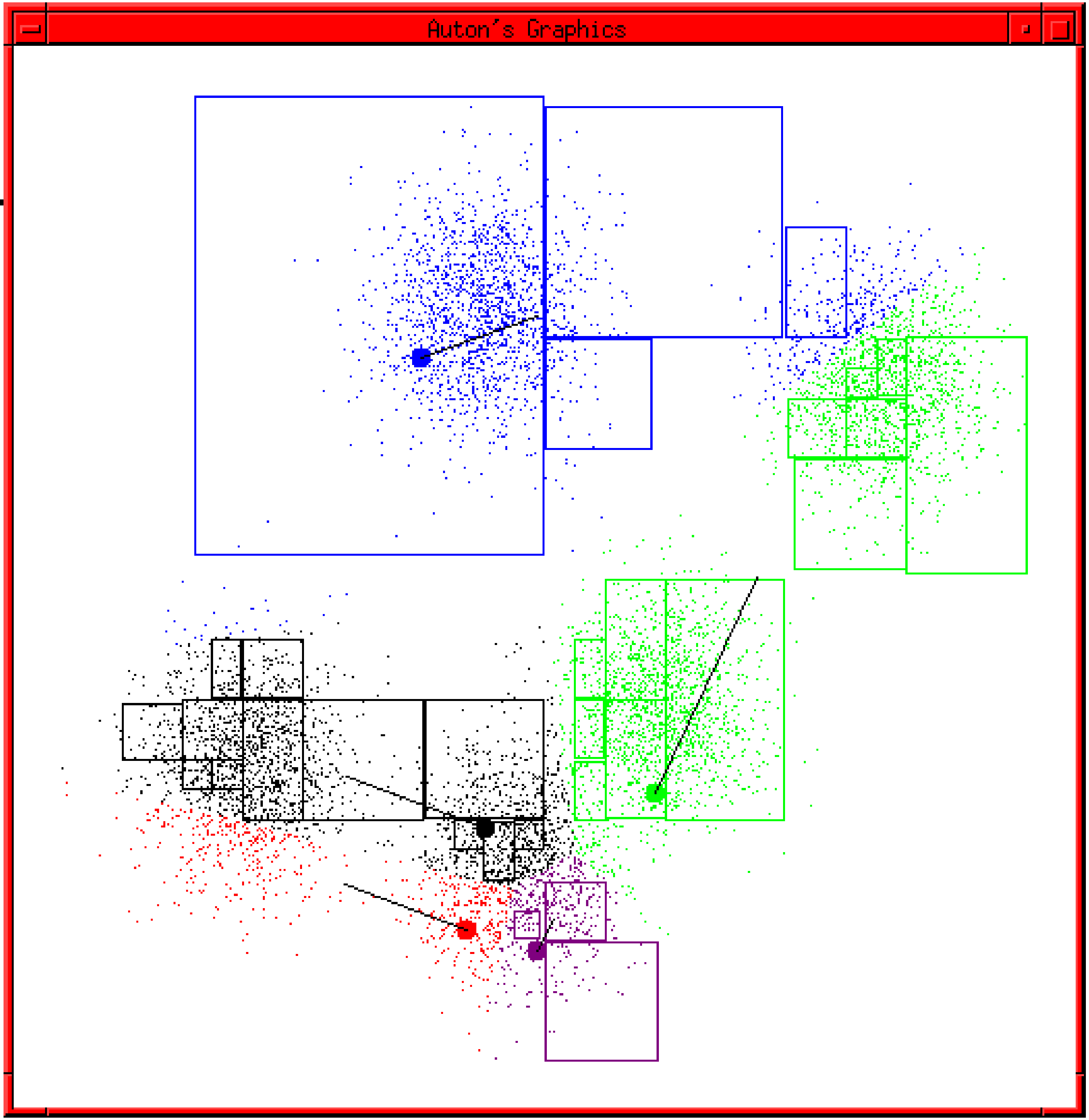
http://www.cs.cmu.edu/~awm/tutorials/

K-means Start

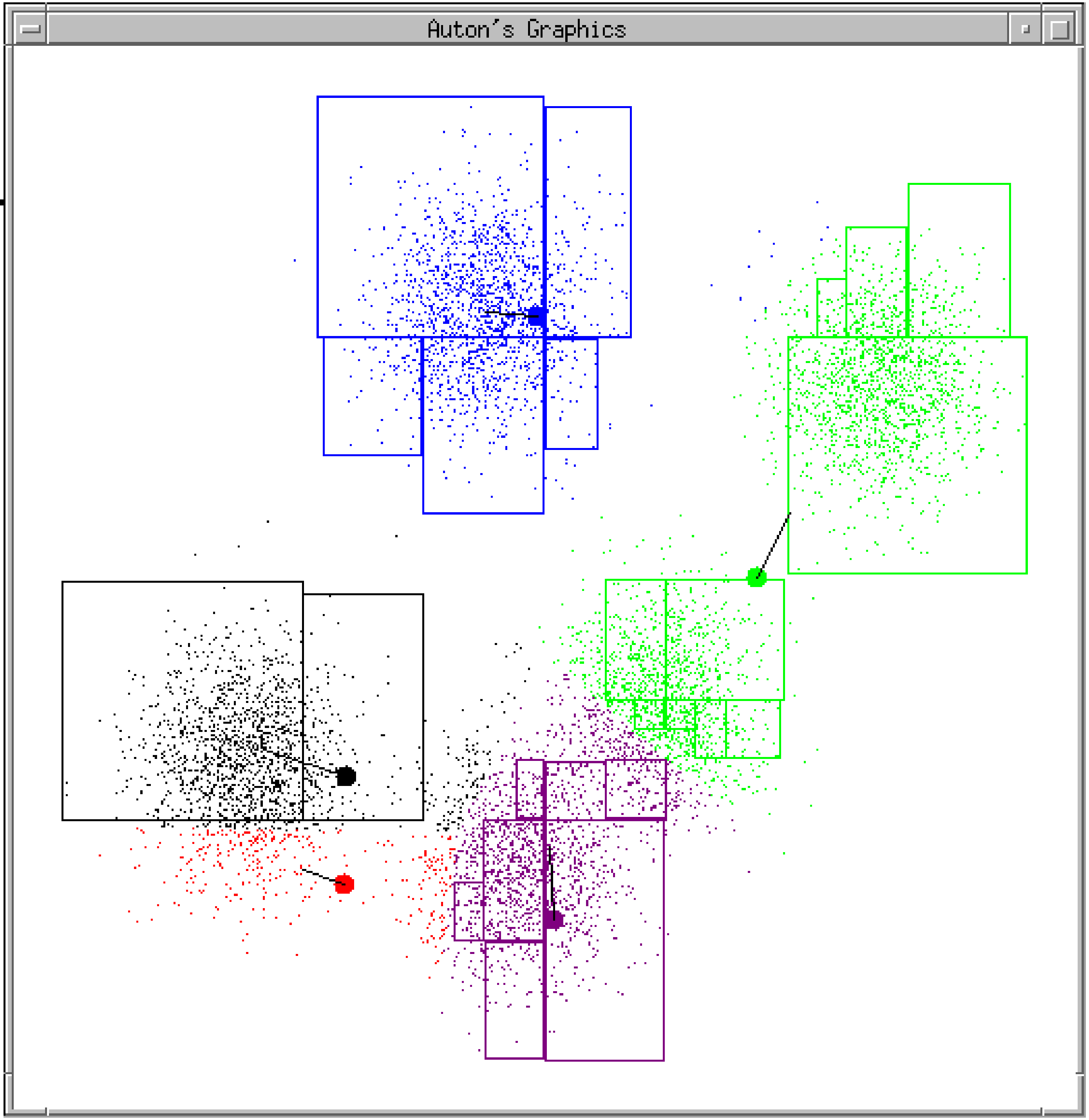
Advance apologies: in Black and White this example will deteriorate

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*)*

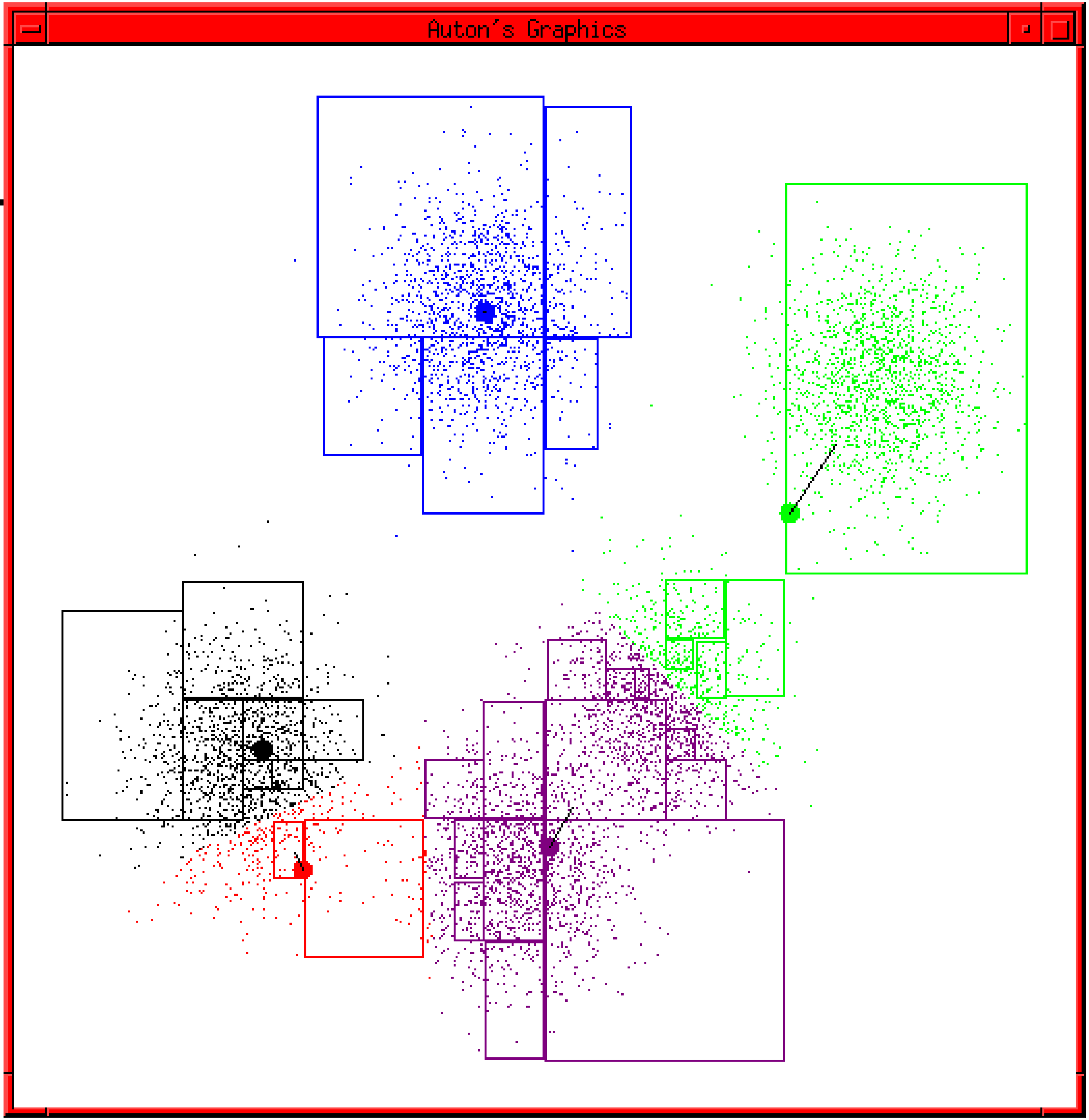


K-means continues



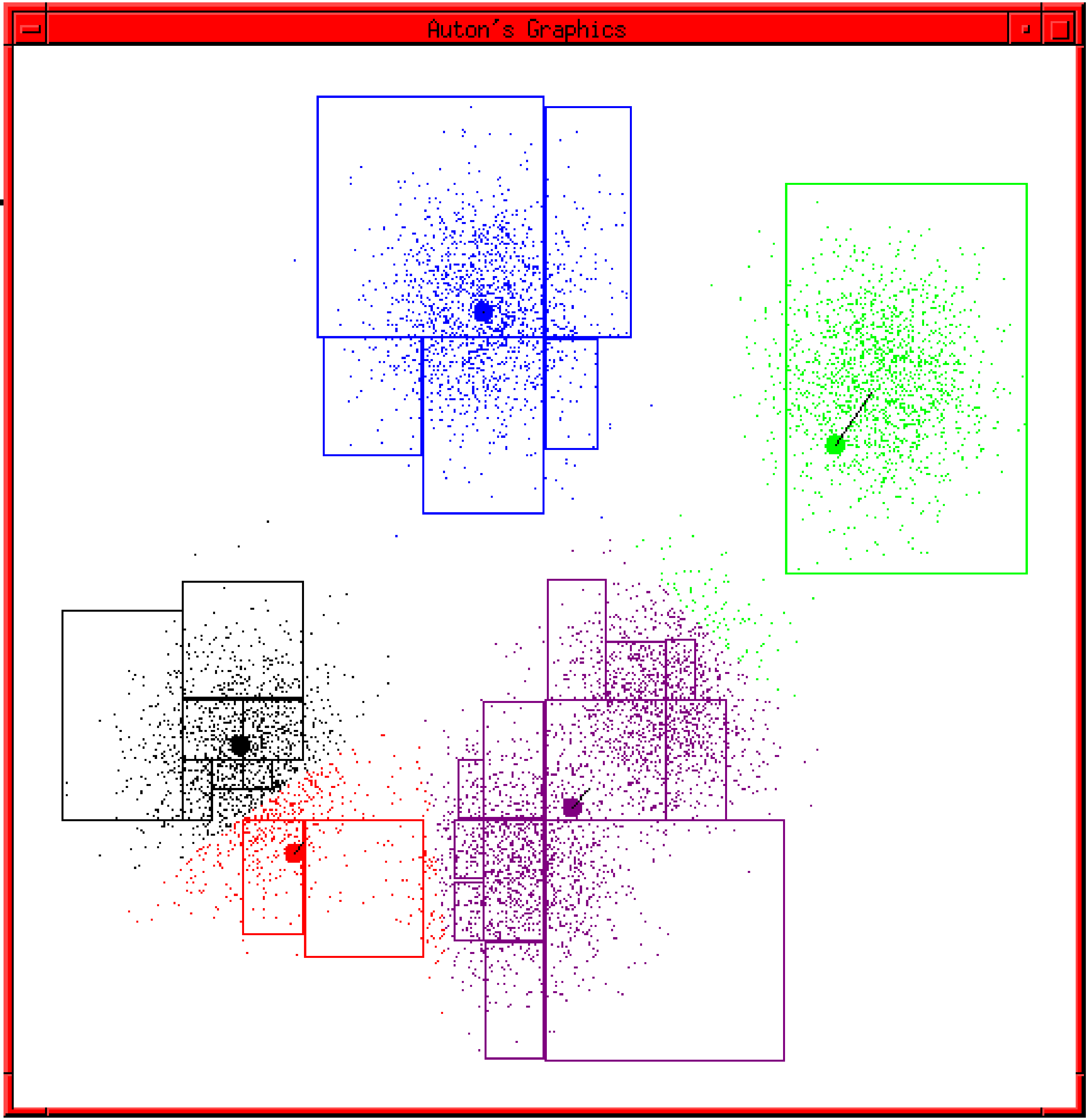
http://www.cs.cmu.edu/~awm/tutorials/

K-means continues…



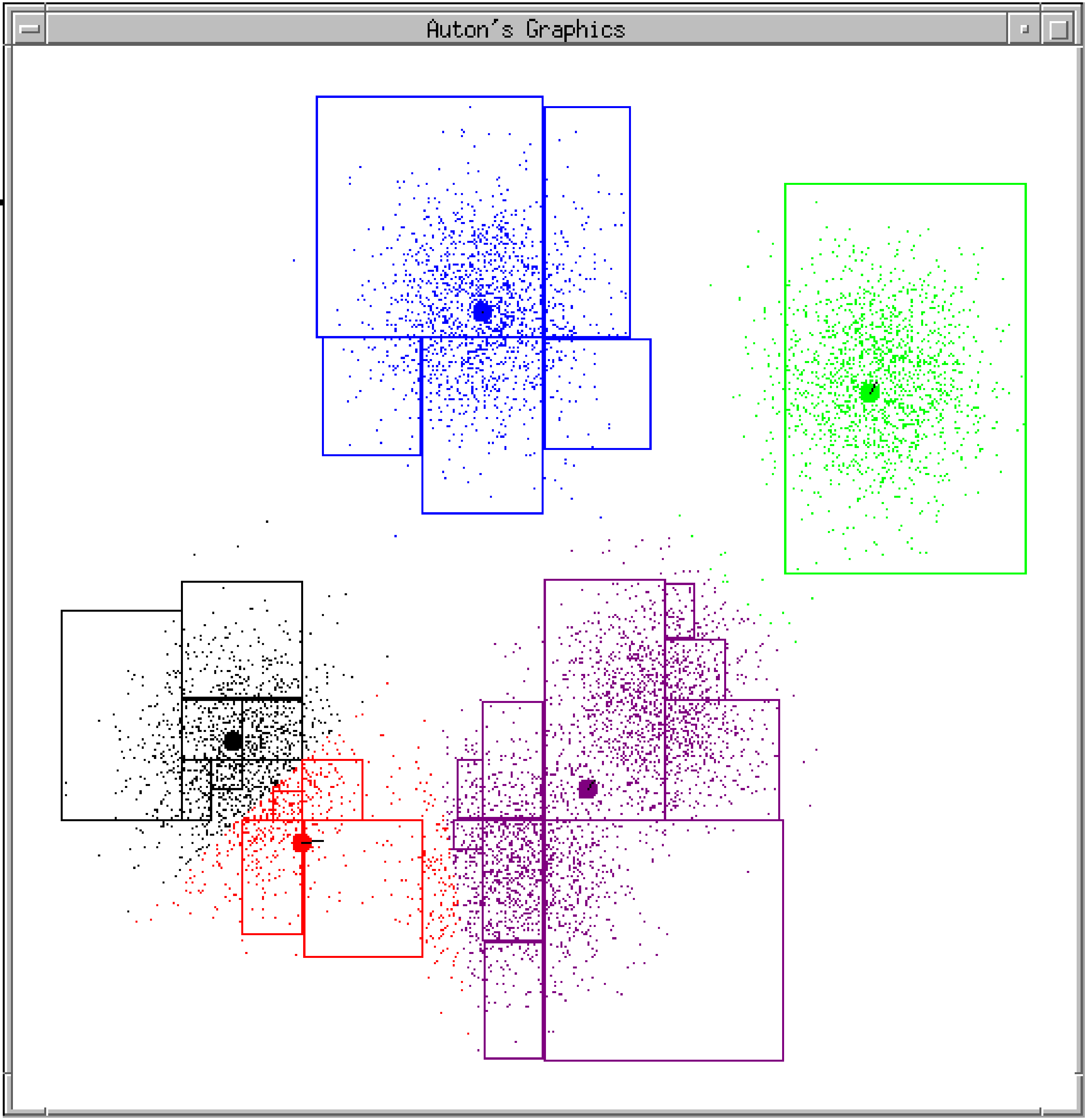
http://www.cs.cmu.edu/~awm/tutorials/

K-means continues…



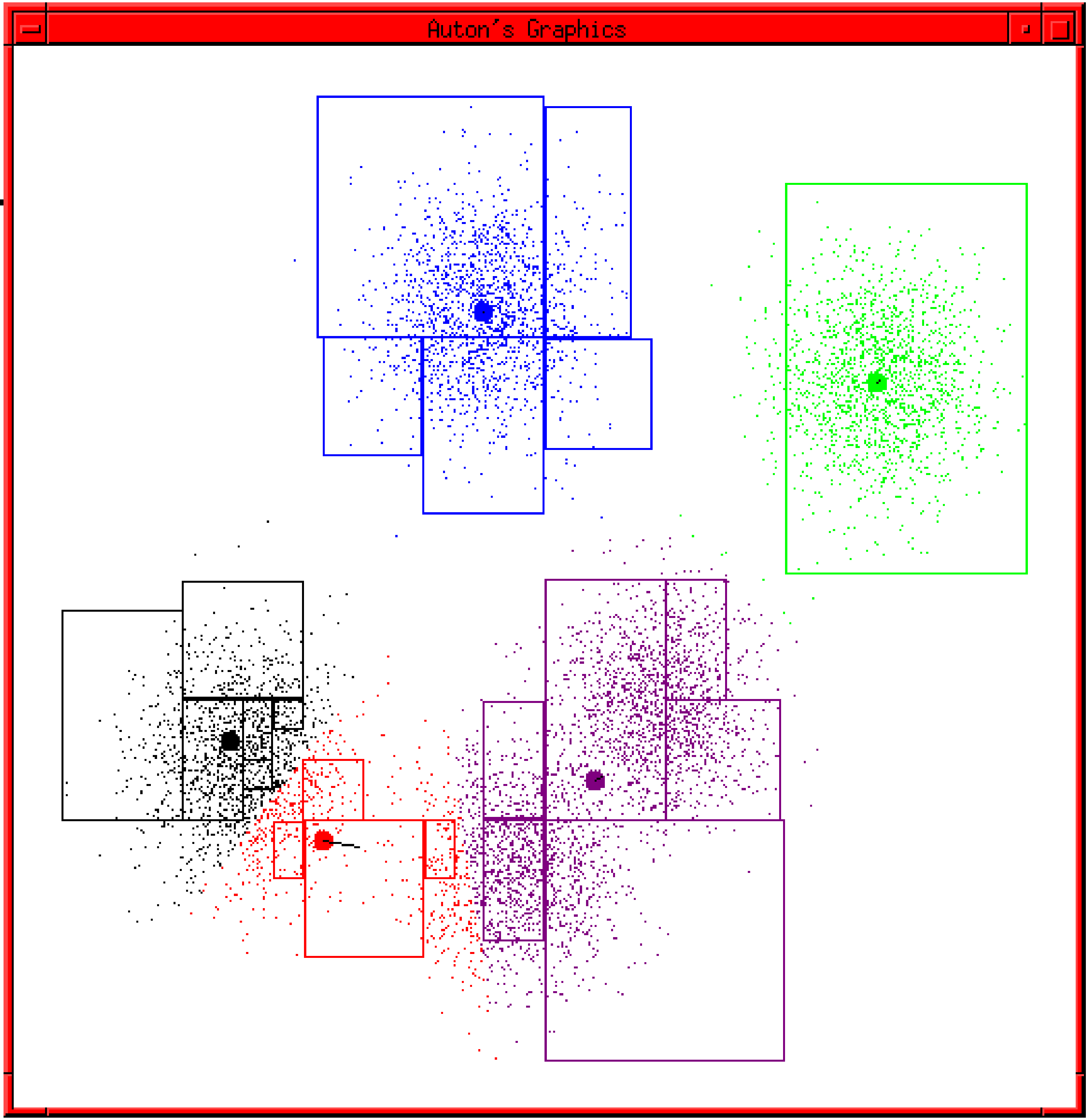
http://www.cs.cmu.edu/~awm/tutorials/

K-means continues...



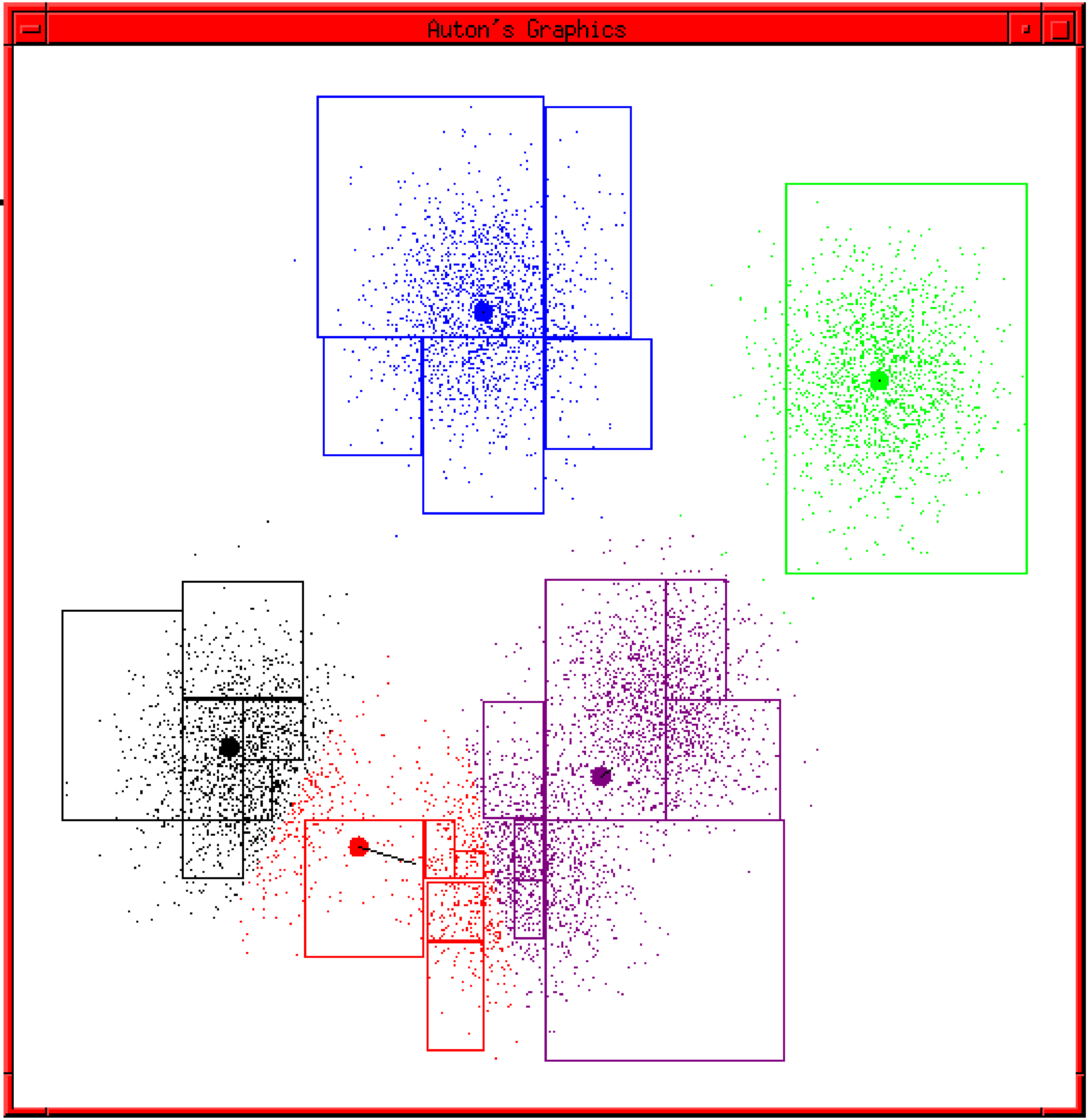
http://www.cs.cmu.edu/~awm/tutorials/

K-means continues…



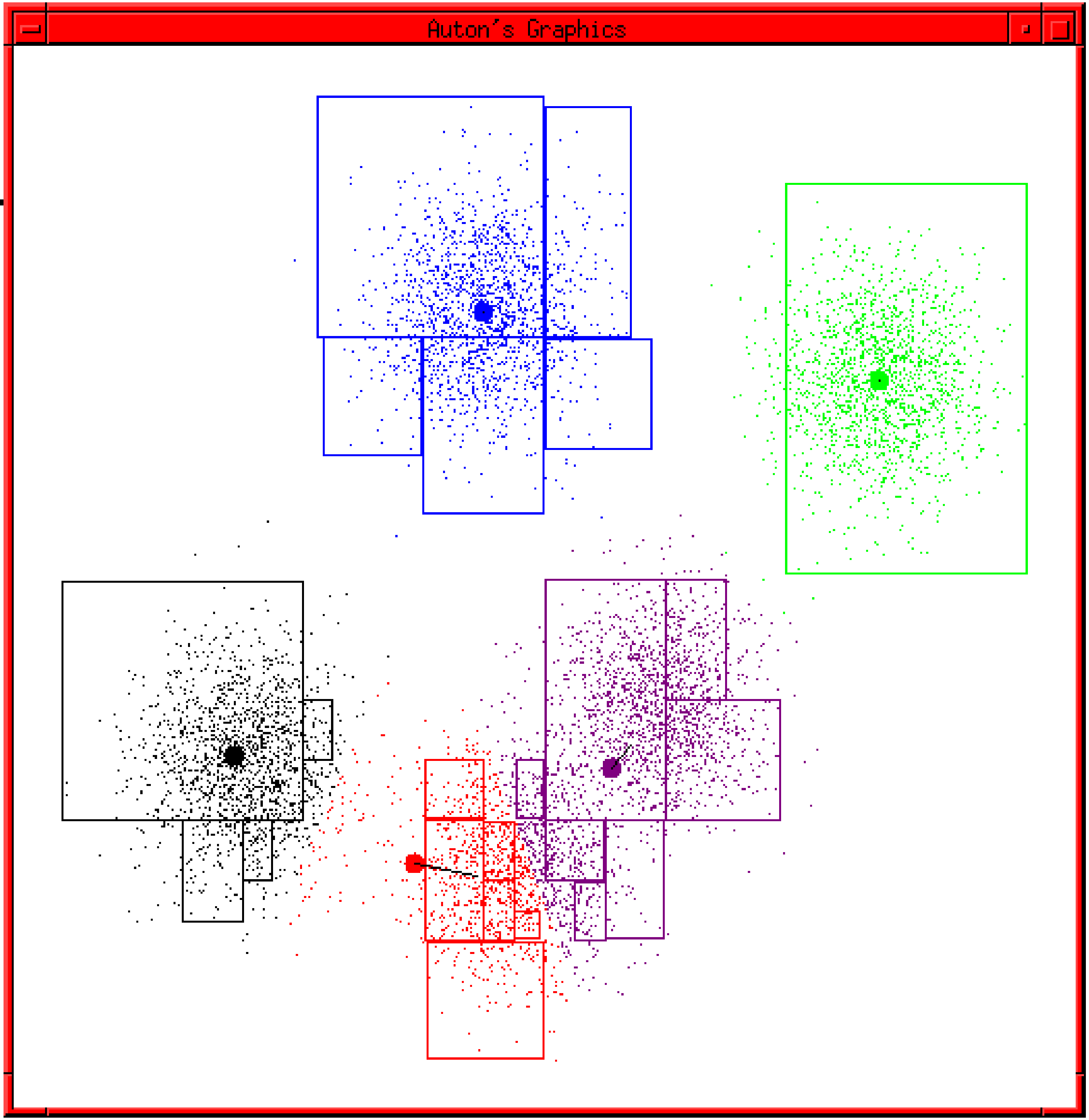
http://www.cs.cmu.edu/~awm/tutorials/

K-means continues…



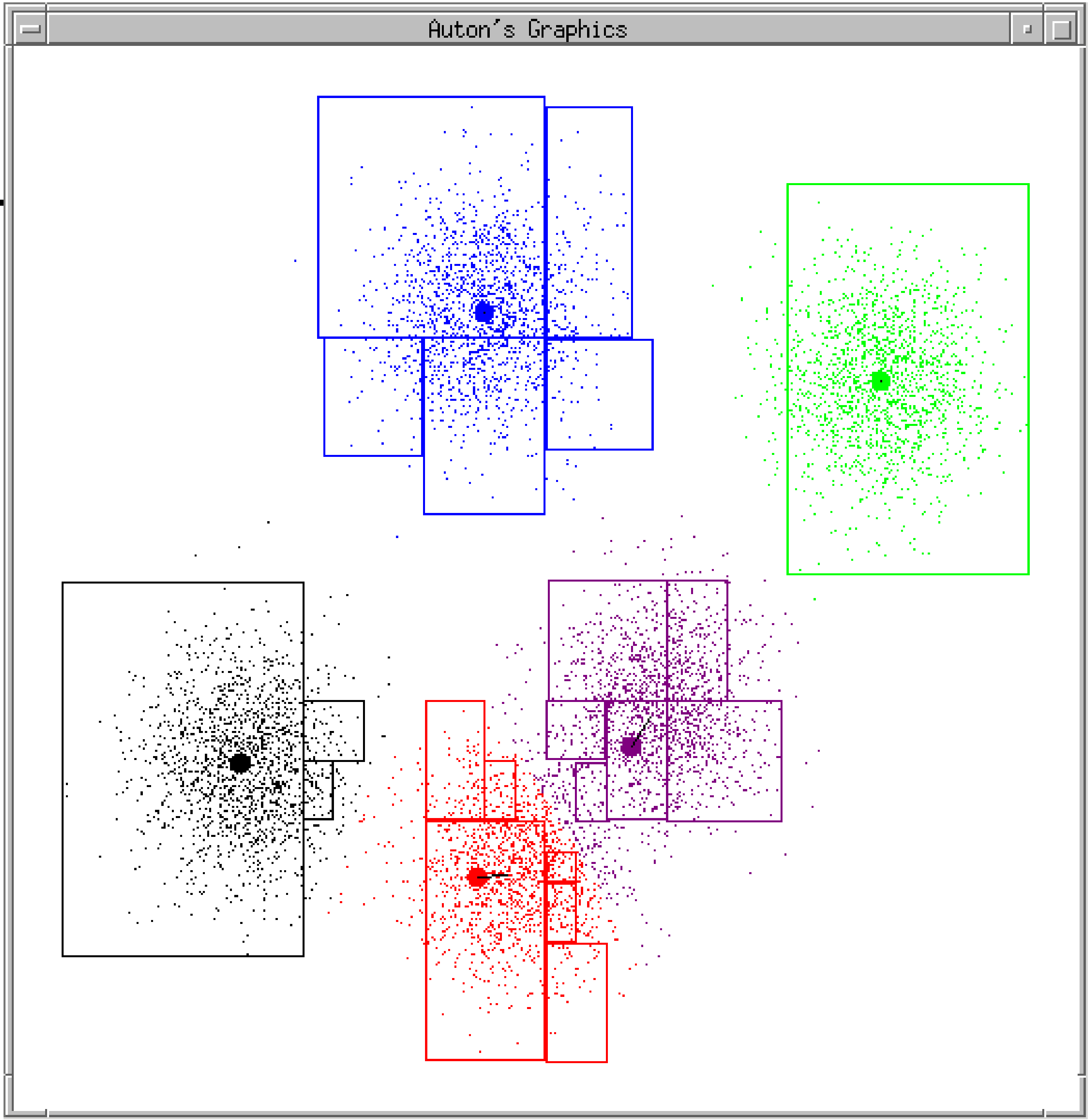
http://www.cs.cmu.edu/~awm/tutorials/

K-means continues…



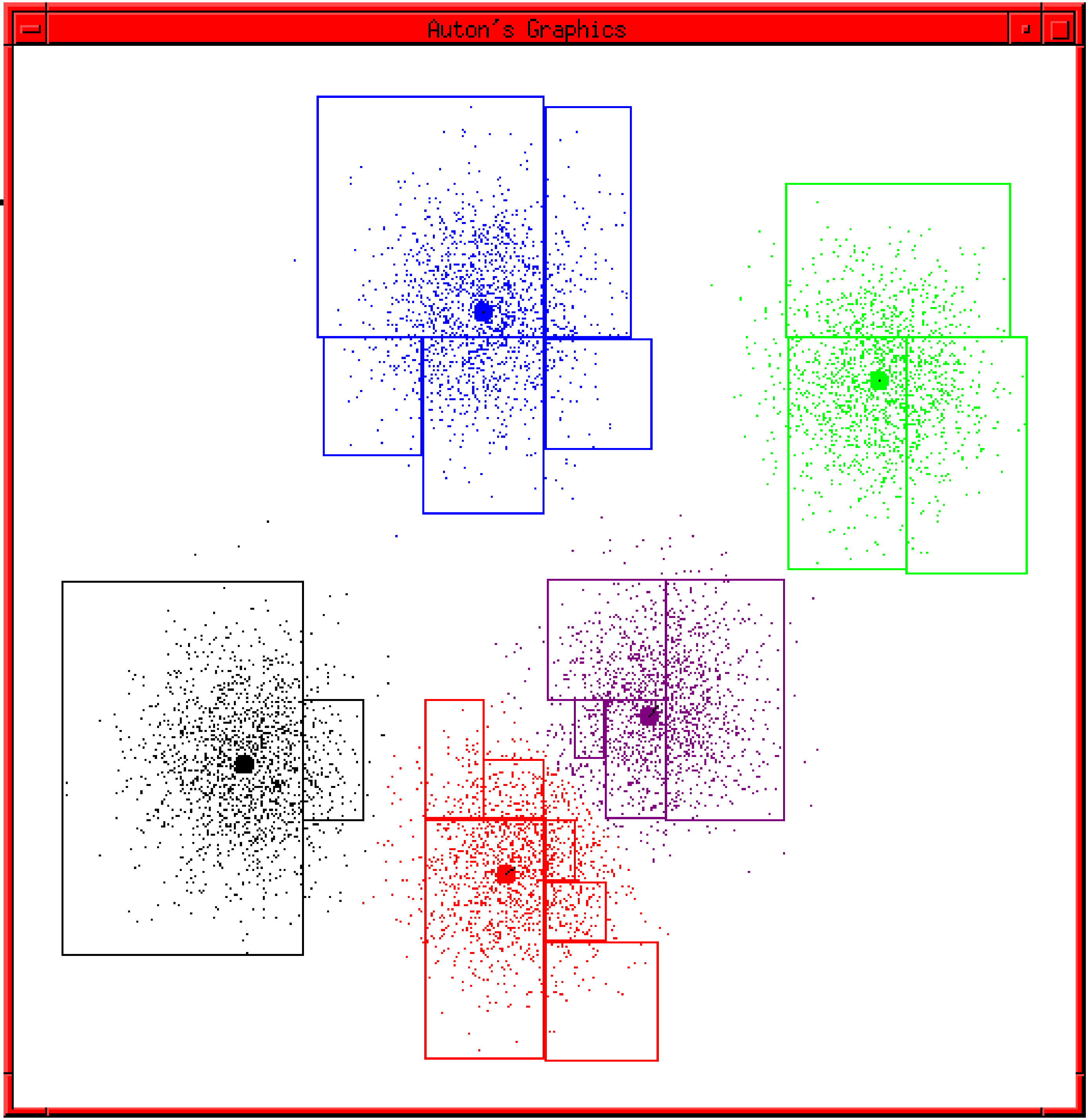
http://www.cs.cmu.edu/~awm/tutorials/

K-means continues…



http://www.cs.cmu.edu/~awm/tutorials/

K-means terminates



http://www.cs.cmu.edu/~awm/tutorials/

K-Means

- This can be seen as attempting to minimize the total squared difference between the data points and their clusters.
- It is guaranteed to converge.
- It is not guaranteed to reach a global minima.
- Commonly used because:
    - It is easy to code
    - It is efficient

Parameterized Probability Distributions

- Parameterized probability distribution:

$$P(X) = P(X|\theta)$$

- θ - The parameters for the distribution.
- Trivial discrete example: X is a Boolean random variable θ indicates the probability that it will be true.

$\theta = .6$	$p(X = TRUE \mid \theta = .6) = .6$ $p(X = FALSE \mid \theta = .6) = .4$
$\theta = .1$	$p(X = TRUE \mid \theta = .1) = .1$ $p(X = FALSE \mid \theta = .1) = .9$

Fitting a Distribution to Data

- Assume we have a set of data points x_1 to x_N.
- The goal is to find a distribution that fits that data. I.e. that could have generated the data.
- One possibility:
  - Maximum likelihood estimate (MLE) find the parameters that maximize the probability of the data:

$$\hat{\theta} = \underset{\theta}{argmax}\, P(x_1, x_2, \ldots, x_N | \theta)$$

ML Learning

- We will assume that x_1 to x_N are **iid** – independent and identically distributed.
- So we can rewrite our problem like this (factorization):

$$\hat{\theta} = \underset{\theta}{argmax} \prod_{i=1}^{N} P(x_i|\theta)$$

- Then we can apply our favorite log trick giving us log likelihood:

$$\hat{\theta} = \underset{\theta}{argmax} \sum_{i=1}^{N} \log(P(x_i|\theta)) = \underset{\theta}{argmax}\, LL$$

Maximizing Log Likelihood

- Just another instance of function maximization.
- One approach, set the partial derivatives to 0 and solve:

$$\frac{\partial LL}{\partial \theta_1} = 0$$

$$\frac{\partial LL}{\partial \theta_2} = 0$$

$$\cdots$$

$$\frac{\partial LL}{\partial \theta_K} = 0$$

- If you can't solve it, gradient descent, or your favorite search algorithm.

Silly Example

- Parameterized coin: Theta – probability of heads:
- ***d*** -- vector of toss data, *h* number of heads, *t* number of tails.

$$P(\mathbf{d}|\theta) = \prod_{i=1}^{N} P(d_i|\theta) = \theta^h (1-\theta)^t$$

$$L(\mathbf{d}|\theta) = log(P(\mathbf{d}|\theta)) = h\, log\,\theta + t\, log(1-\theta)$$

$$\frac{\partial L}{\partial \theta} = \frac{h}{\theta} - \frac{t}{1-\theta} = 0 \quad \rightarrow \quad \theta = \frac{h}{h+t}$$

Remember: $\frac{d}{dx} log(x) = 1/x$

Example borrowed from Russel & Norvig

EM

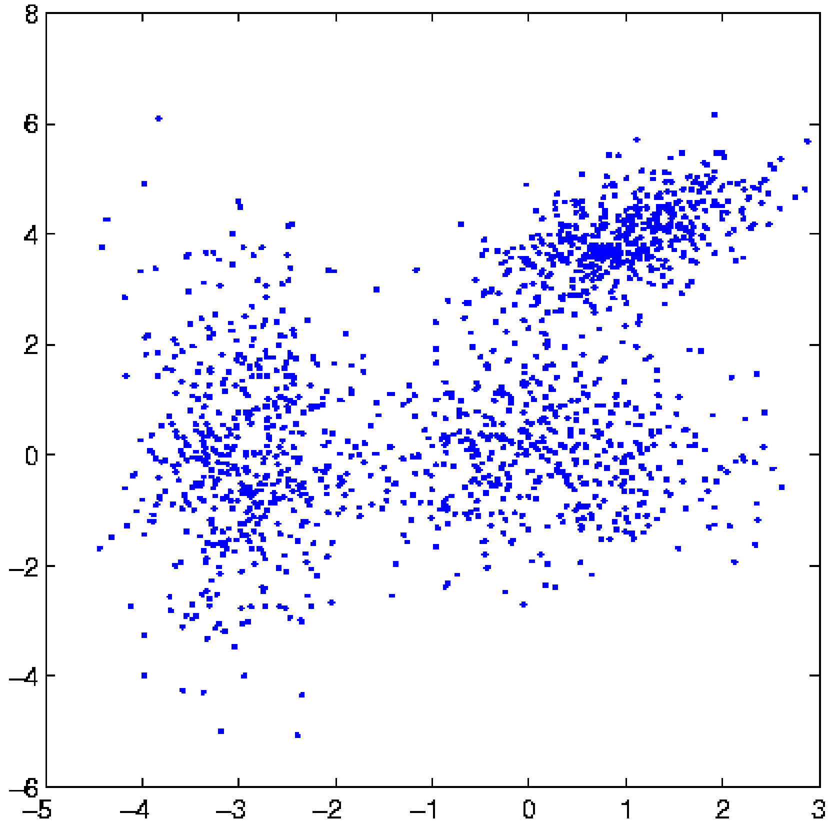
- A general approach to maximum likelihood learning in cases of missing data.
- E.g. clustering... Each data point REALLY comes from some cluster, but that data is missing.

EM

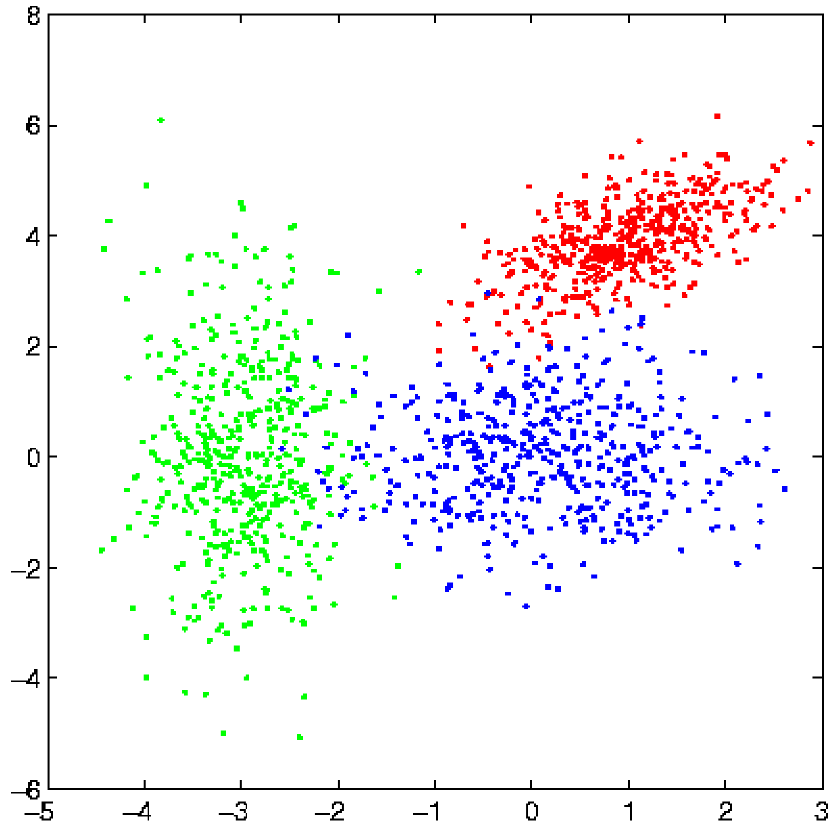
- Hidden variables are Z, observed variables are X.
- Guess an assignment to our parameters: $\hat{\theta}$.
- Expectation-Step:
  - Compute the expected value of our hidden variables E[Z].
- Maximization-Step
  - Pretend that E[Z] is the true value of Z and use maximum likelihood to calculate a new $\hat{\theta}$

Gaussian Mixture Example

We have this:



Life would be easier if we had this:



EM for GMM

- E-Step ($p_{i,j}$ is the probability that point i was generated by mixture component j)

$$p_{i,j} = \frac{p(x_i | \mu_j, \Sigma_j) \pi_j}{\sum_{k=1}^{K} p(x_i | \mu_k, \Sigma_k) \pi_k}$$

EM for GMM

- M Step: Update the parameters:

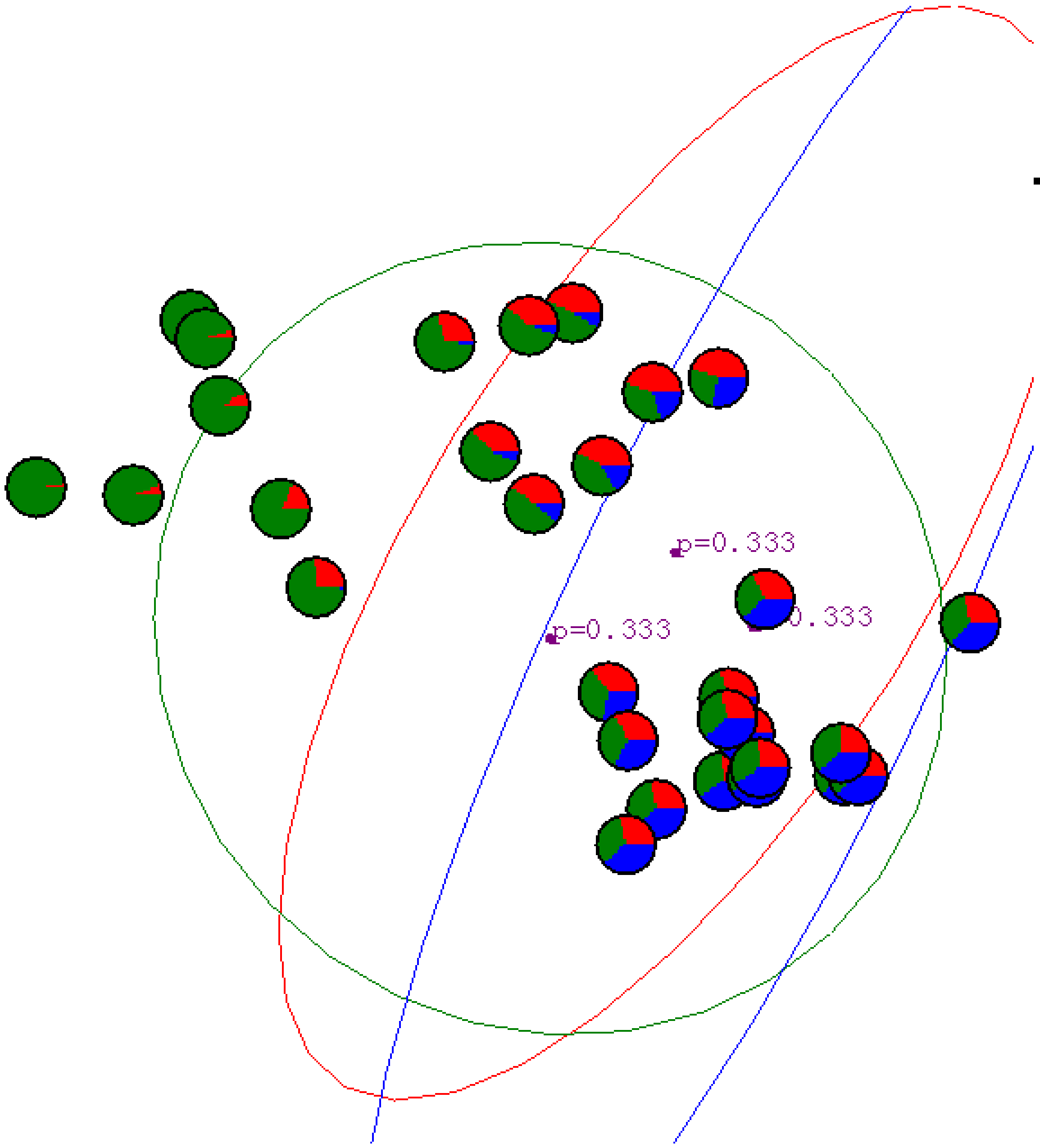
$$\mu_j = \frac{\sum_{i=1}^{N} p_{i,j} x_i}{\sum_{i=1}^{N} p_{i,j}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{N} p_{i,j} (\boldsymbol{x_i} - \mu_j)(\boldsymbol{x_i} - \mu_j)^T}{\sum_{i=1}^{N} p_{i,j}}$$

$$\pi_j = \frac{1}{N} \sum_{i=1}^{N} p_{i,j}$$

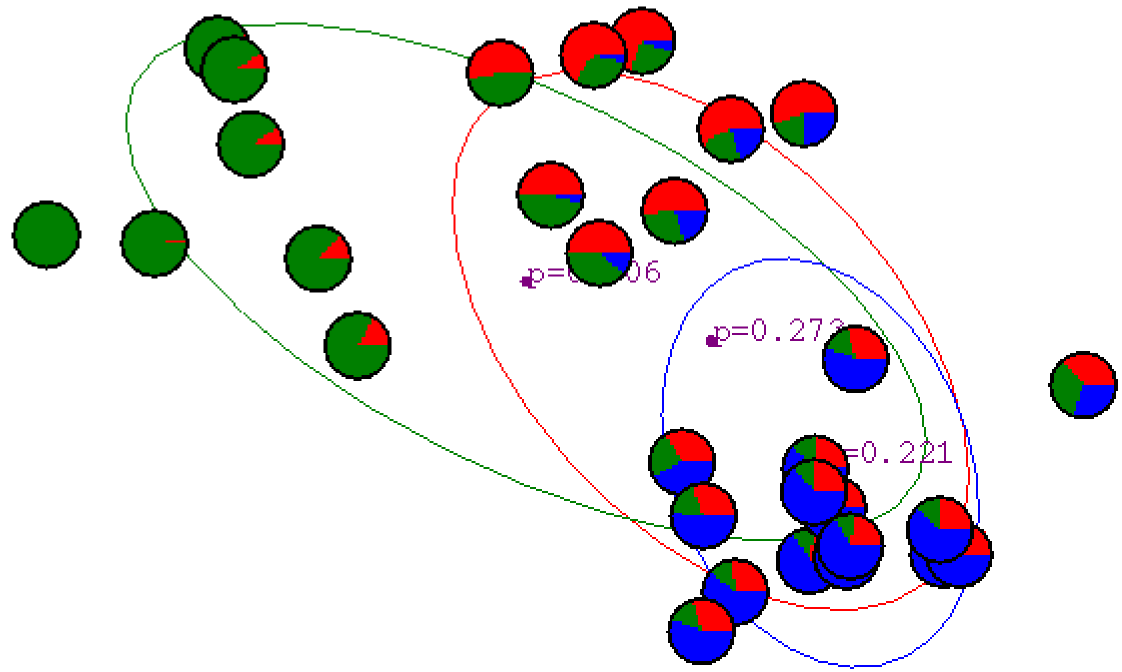
Gaussian Mixture Example: Start

Advance apologies: in Black and White this example will be incomprehensible



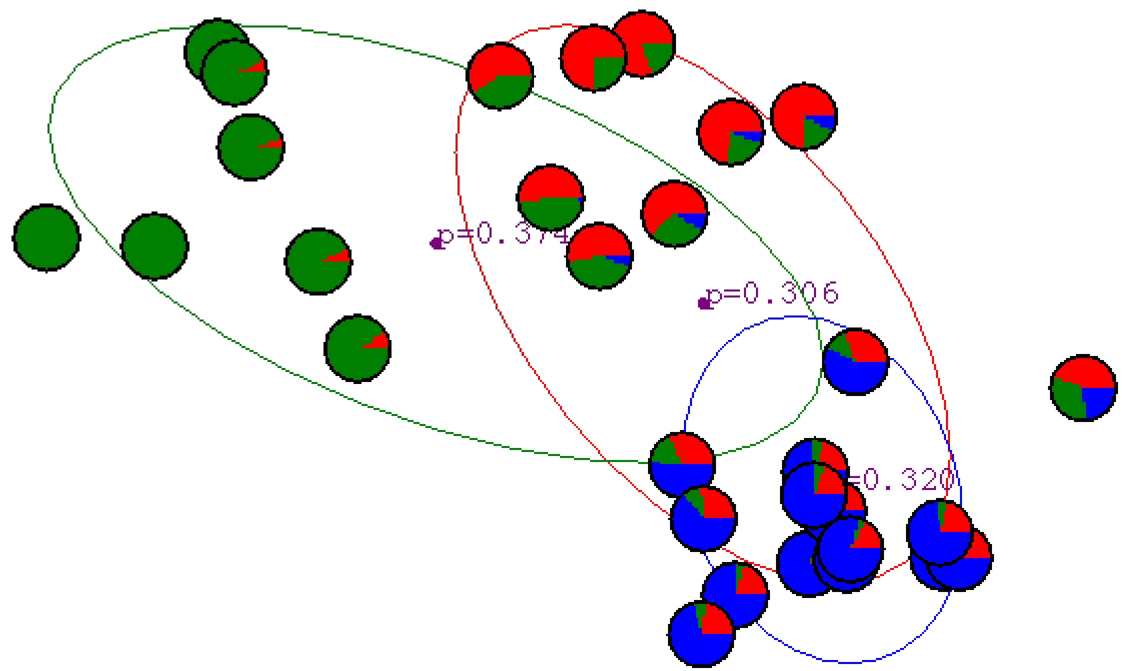
http://www.cs.cmu.edu/~awm/tutorials/

After first iteration



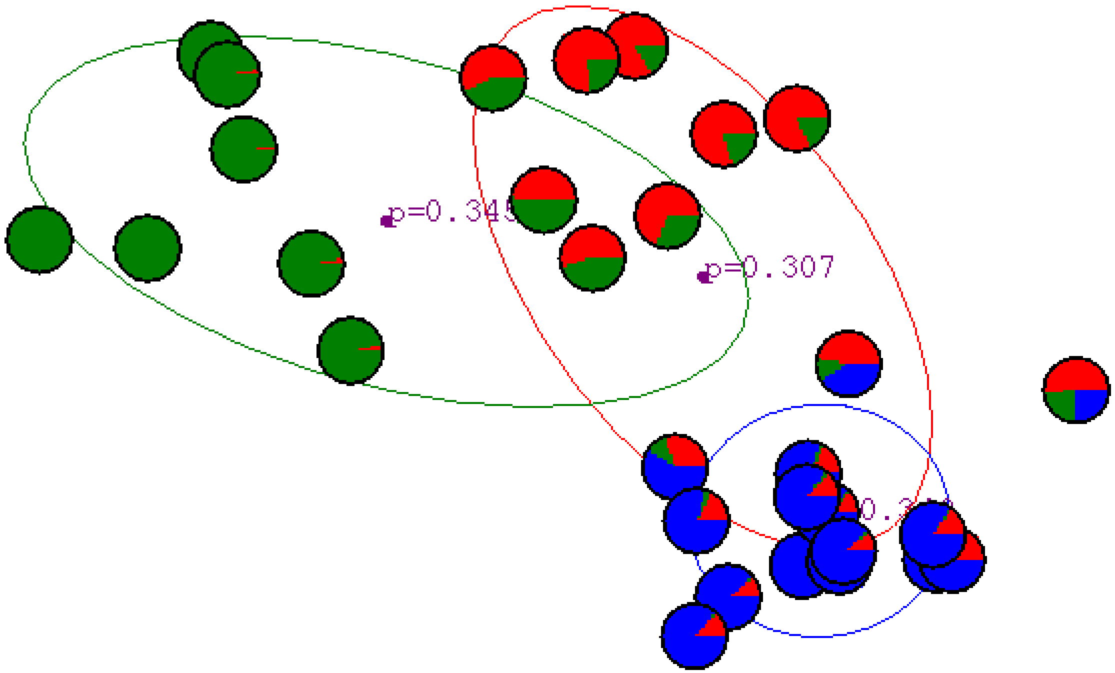
http://www.cs.cmu.edu/~awm/tutorials/

After 2nd iteration



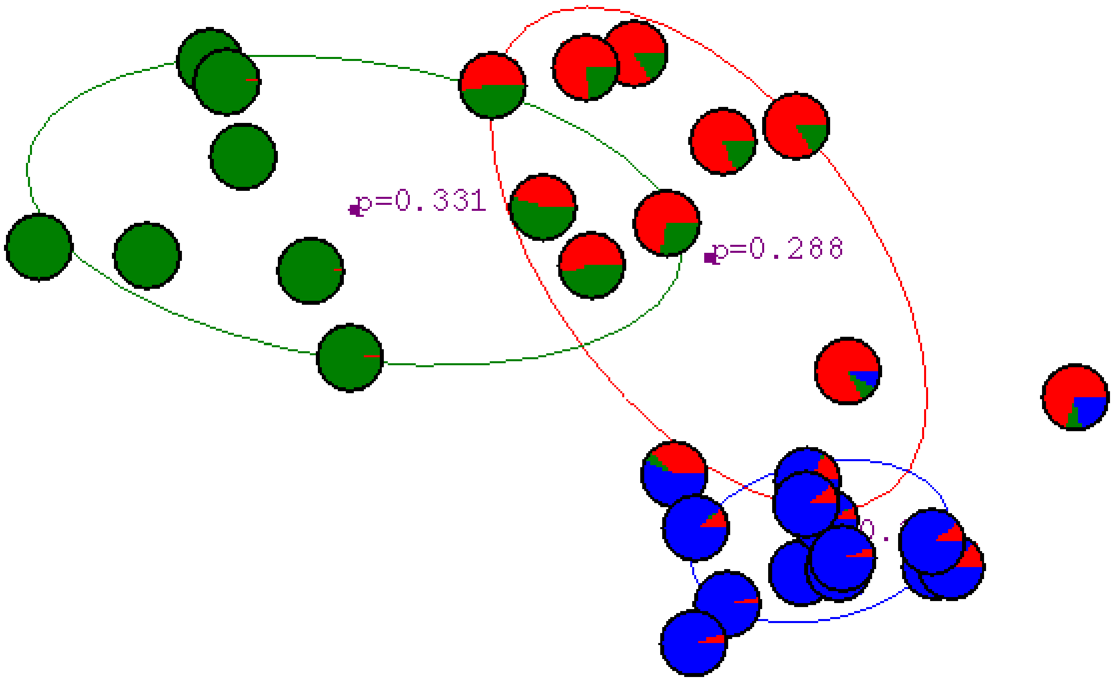
http://www.cs.cmu.edu/~awm/tutorials/

After 3rd iteration



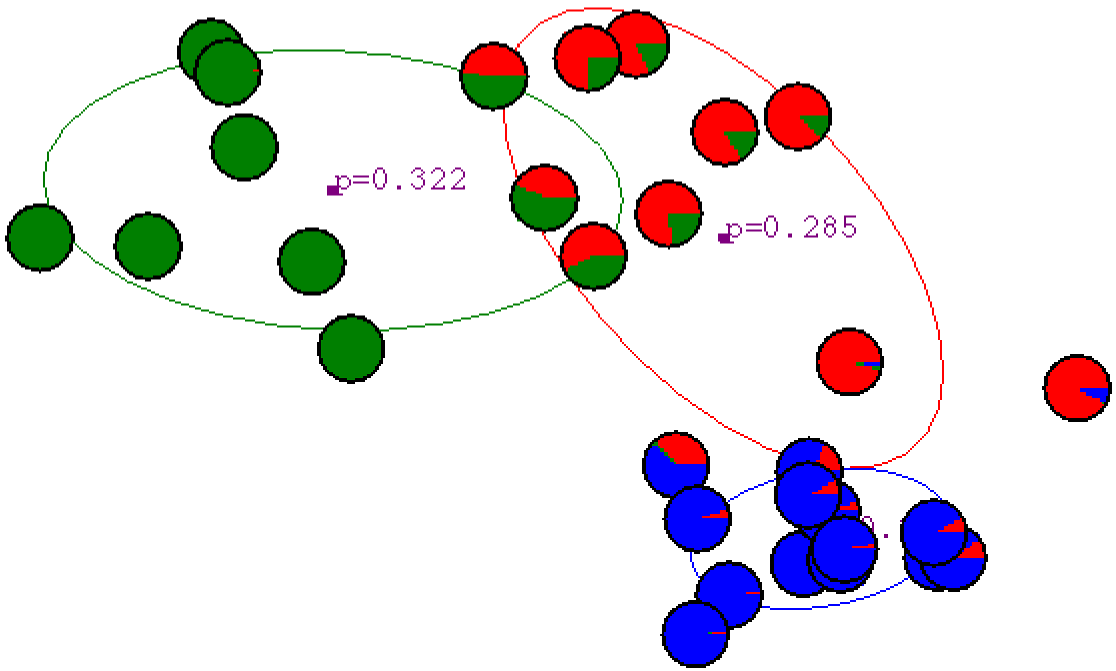
http://www.cs.cmu.edu/~awm/tutorials/

After 4th iteration



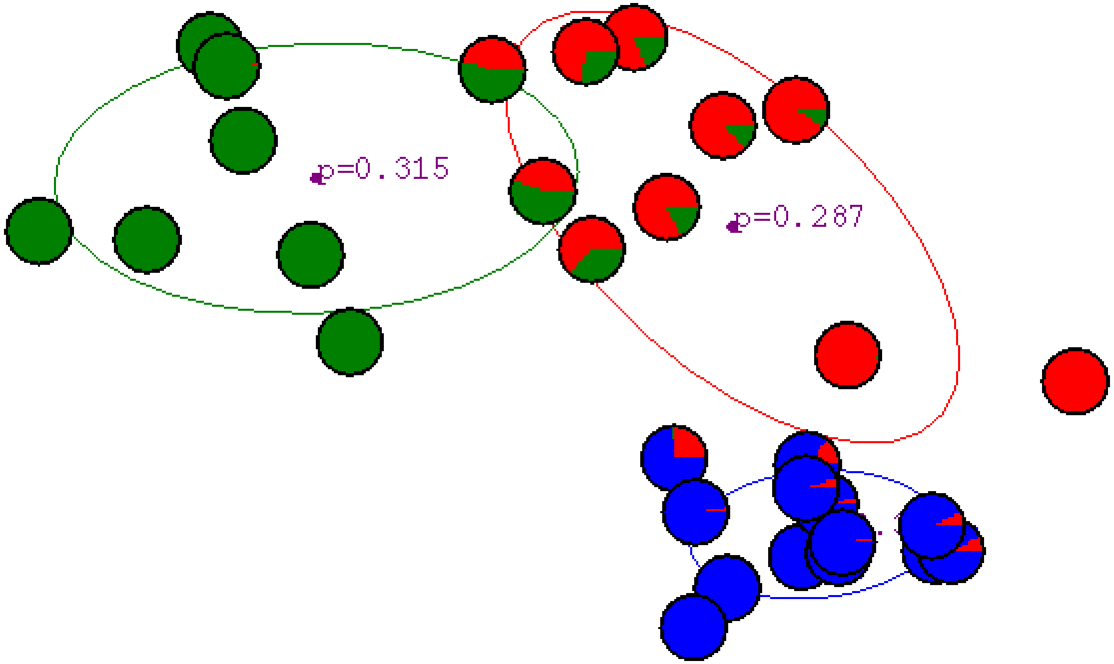
http://www.cs.cmu.edu/~awm/tutorials/

After 5th iteration



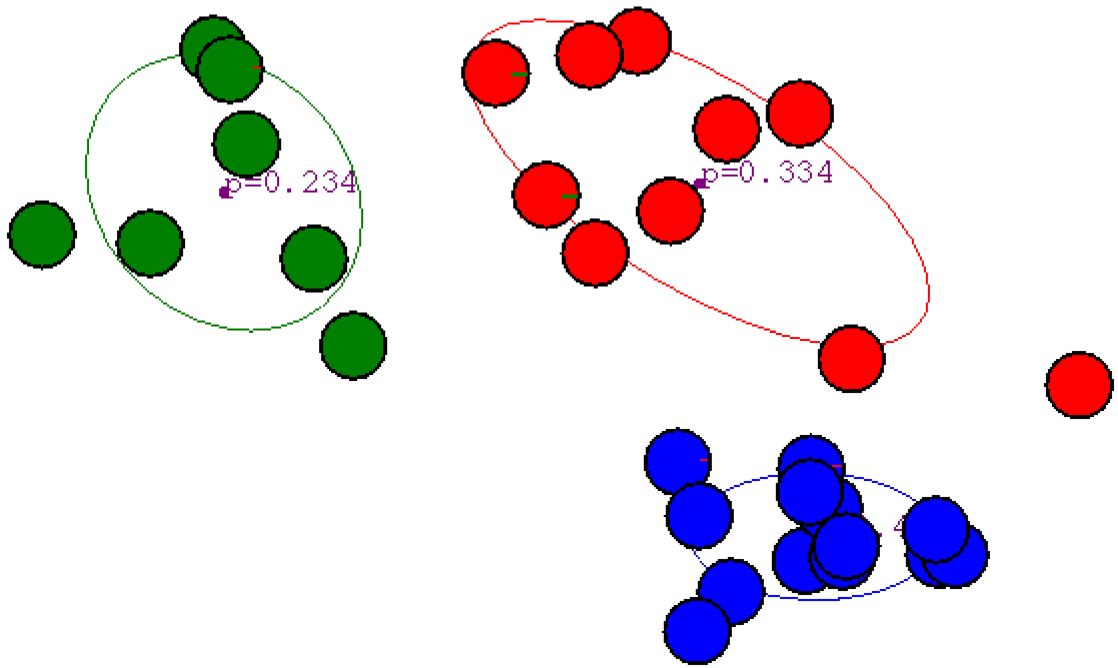
http://www.cs.cmu.edu/~awm/tutorials/

After 6th iteration



http://www.cs.cmu.edu/~awm/tutorials/

After 20th iteration



http://www.cs.cmu.edu/~awm/tutorials/