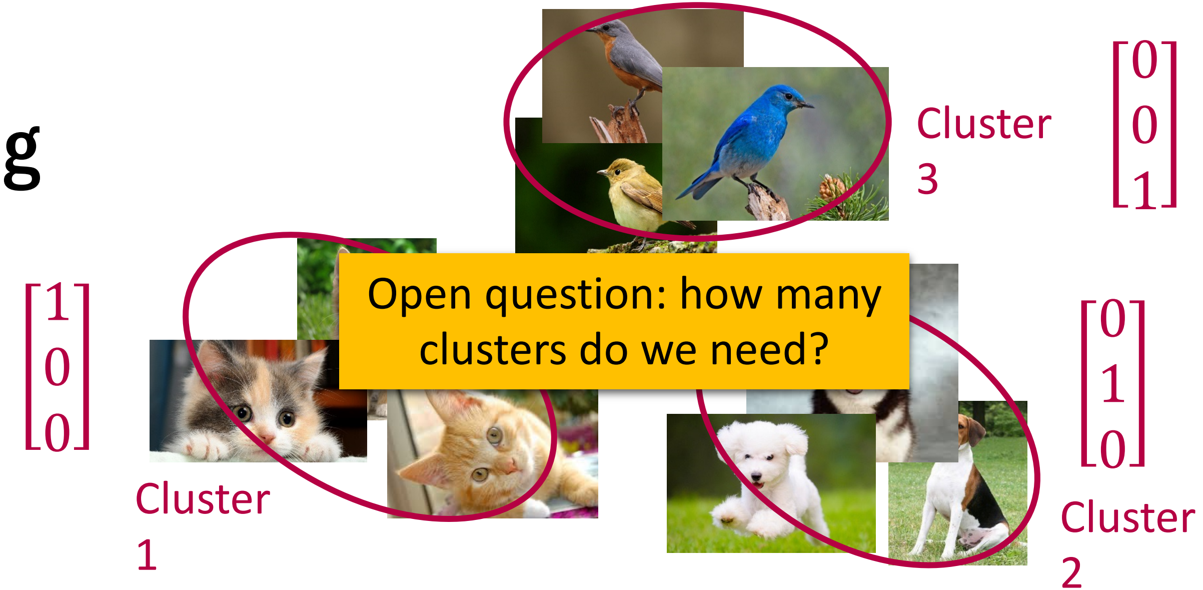
Clustering

Part I

Clustering


Cluster 3
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
Cluster 1
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
Cluster 2

Open question: how many clusters do we need?

- K-means
  - Clustering $X = \{x^1, \cdots, x^n, \cdots, x^N\}$ into K clusters
  - Initialize cluster center c^i, i=1,2, … K (K random x^n from X)
  - Repeat
    - For all x^n in X:

$$b_i^n \begin{cases} 1 & x^n \text{ is most “}\textbf{\textit{close}}\text{” to } c^i \\ 0 & \text{Otherwise} \end{cases}$$

    - Updating all c^i:

$$c^i = \sum_{x^n} b_i^n x^n \Big/ \sum_{x^n} b_i^n$$

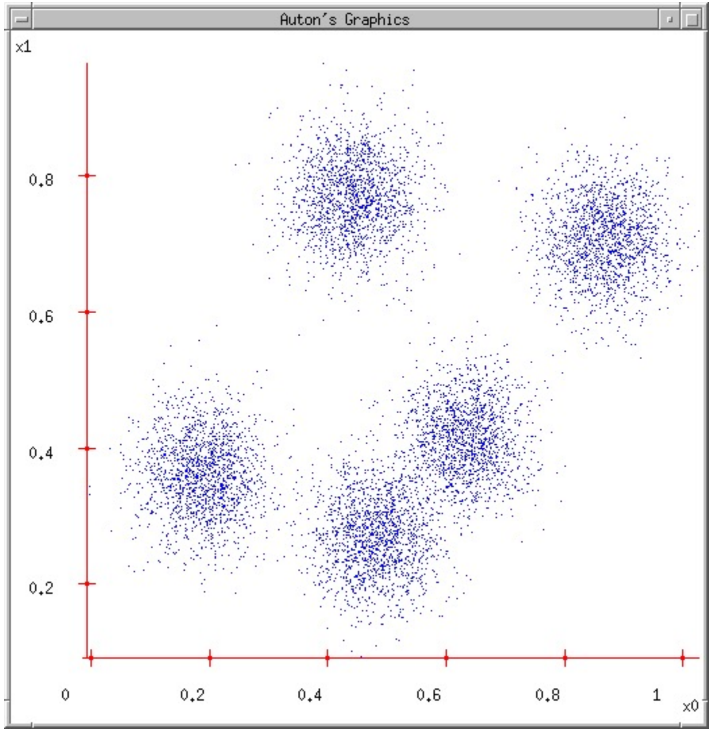
K-Means Clustering Algorithm

1. Choose # of clusters desired, *k*
2. Start with a partition into k clusters

   Often based on random selection of k centroids
3. At each step, move each record to cluster with closest centroid
4. Re-compute centroids, repeat step 3
5. Stop when moving records increases within-cluster dispersion

K-Means: Part I

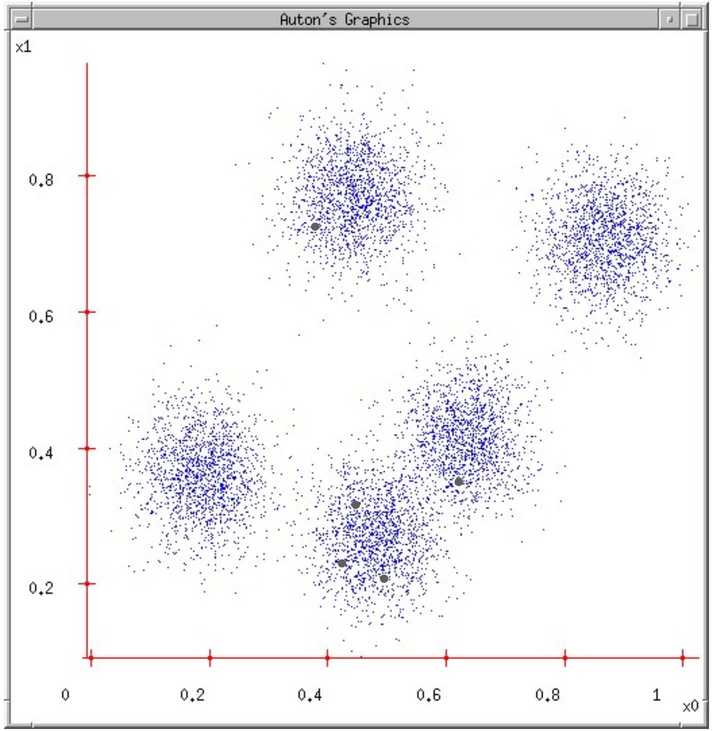
1. Ask user how many clusters they'd like. (e.g. k=5)



K-Means: Part II

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

2. Randomly guess k cluster Center locations

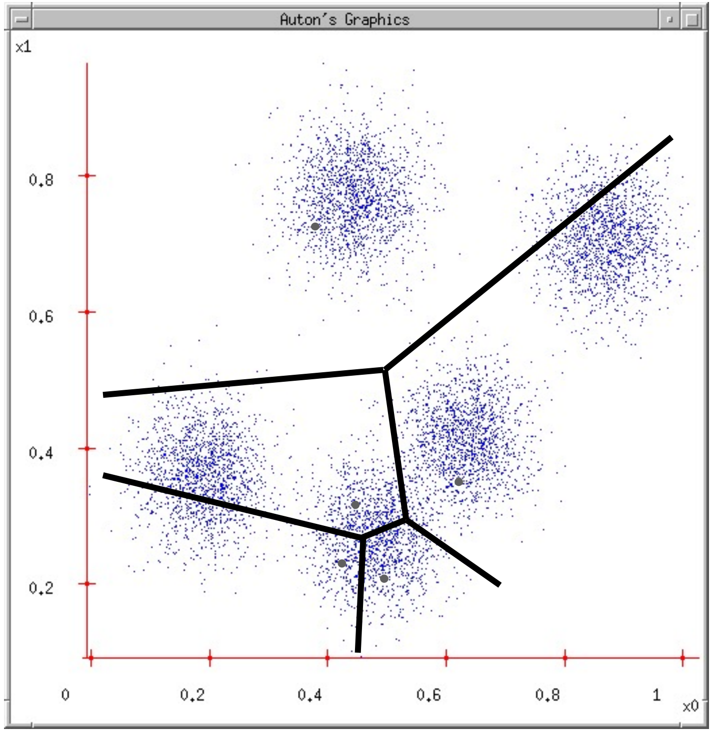


K-Means: Part III

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

2. Randomly guess k cluster Center locations

3. Each datapoint (blue) finds out which Center it's closest to. (Thus, each Center "owns" a set of datapoints). Black lines are the ownership boundaries of each centroid



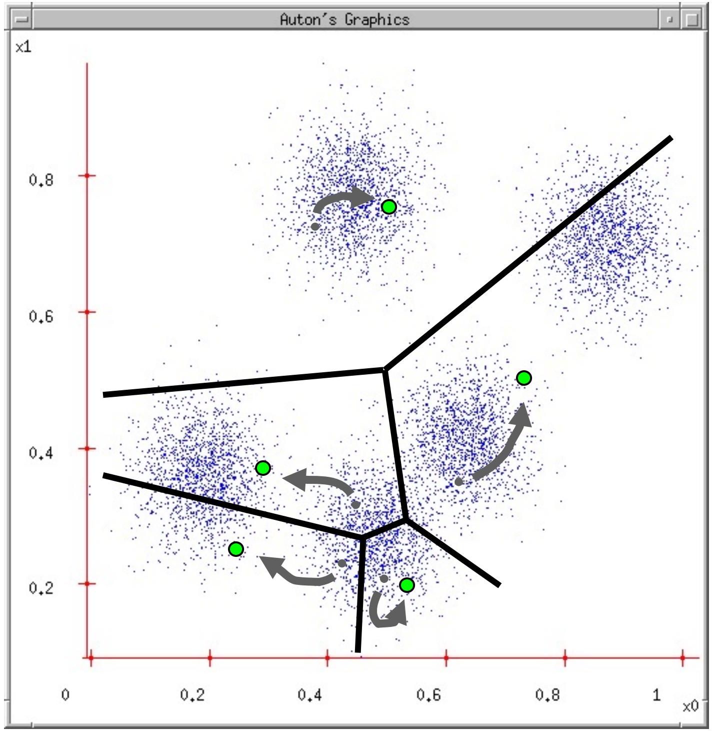
K-Means: Part IV

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

2. Randomly guess k cluster Center locations

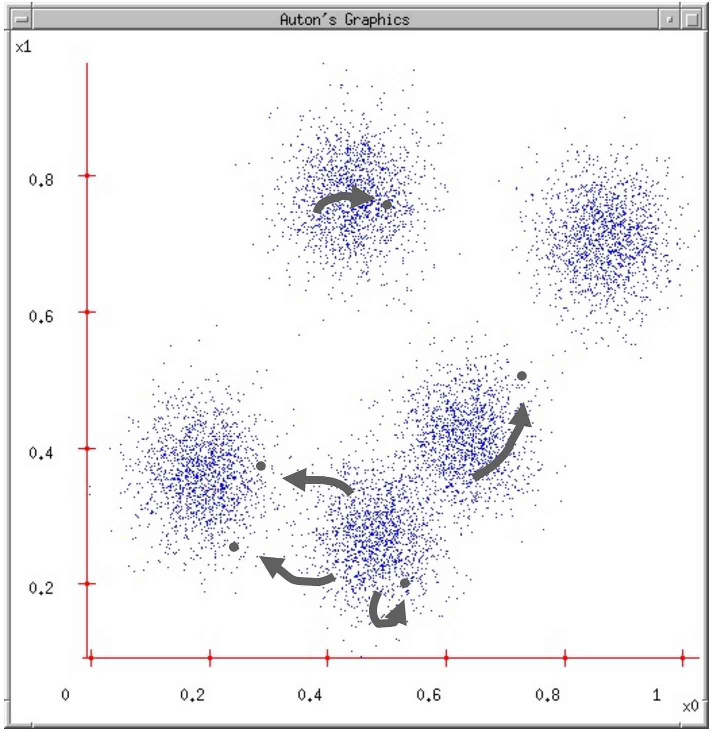
3. Each datapoint finds out which Center it's closest to.

4. Each Center finds the centroid of the points it owns



K-Means: Part V

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…
5. …and jumps there
6. …Repeat until terminated!

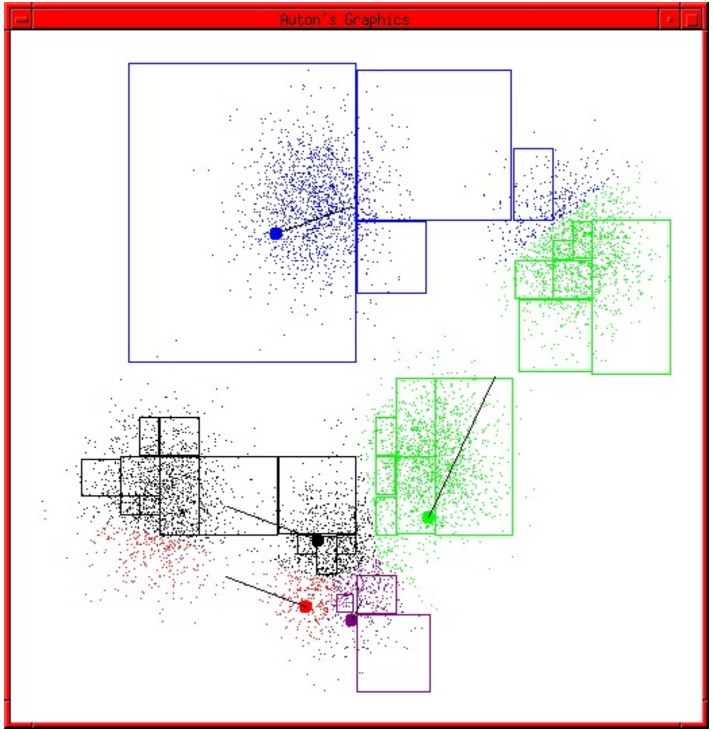


K-Means: Part VI

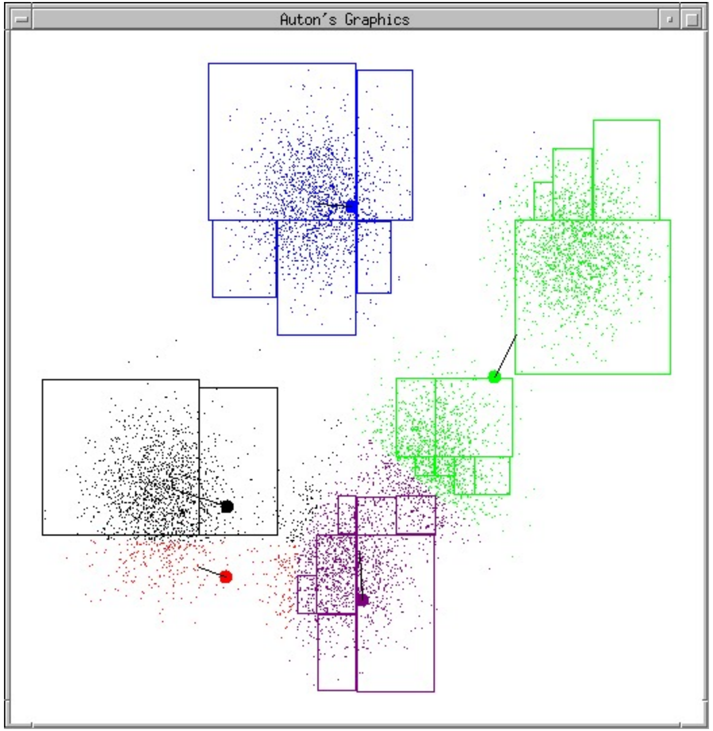
Advance apologies: in Black and White this example will deteriorate

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

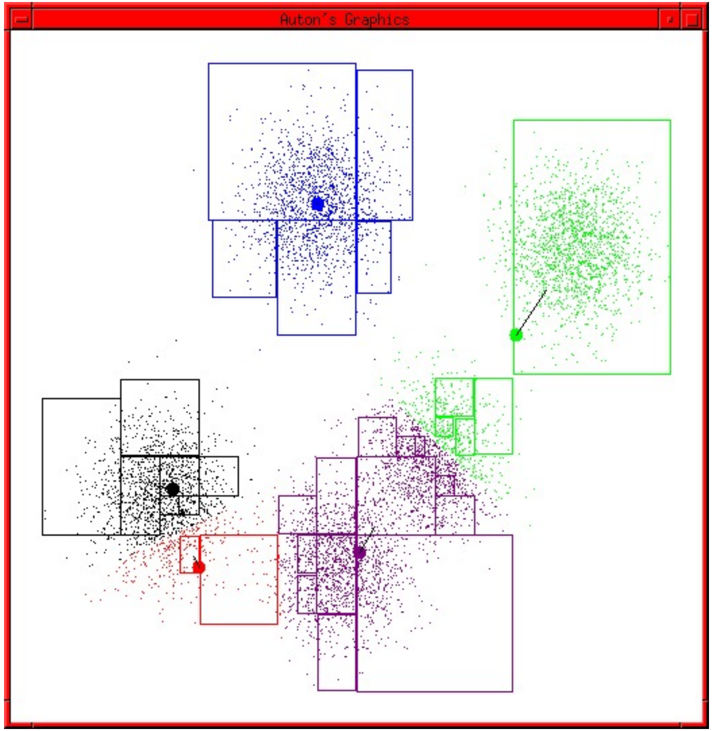
Dan Pelleg and Andrew Moore. Accelerating Exact k-means Algorithms with Geometric Reasoning. Proc. Conference on Knowledge Discovery in Databases 1999, (KDD99) (available on www.autonlab.org/pap.html)



K-Means: Part VII

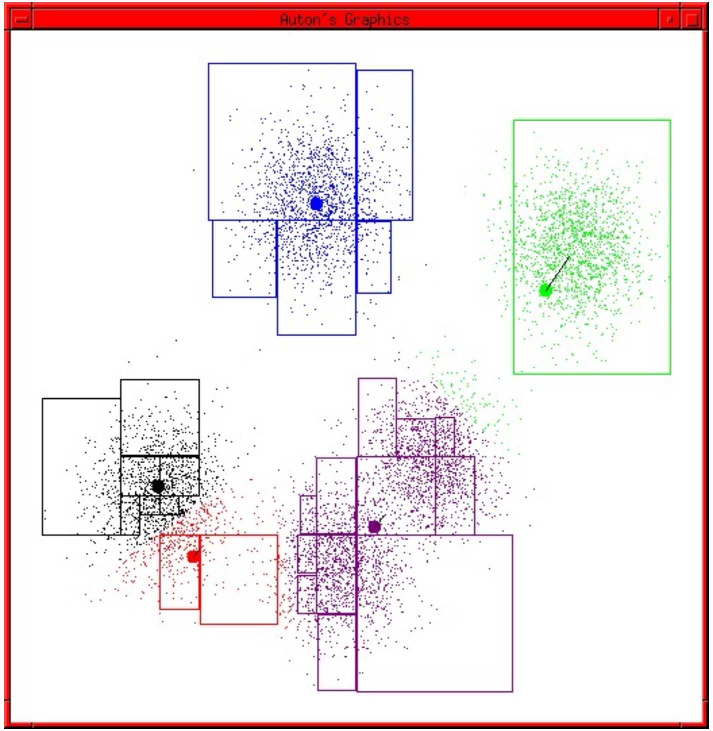


K-Means: Part VIII



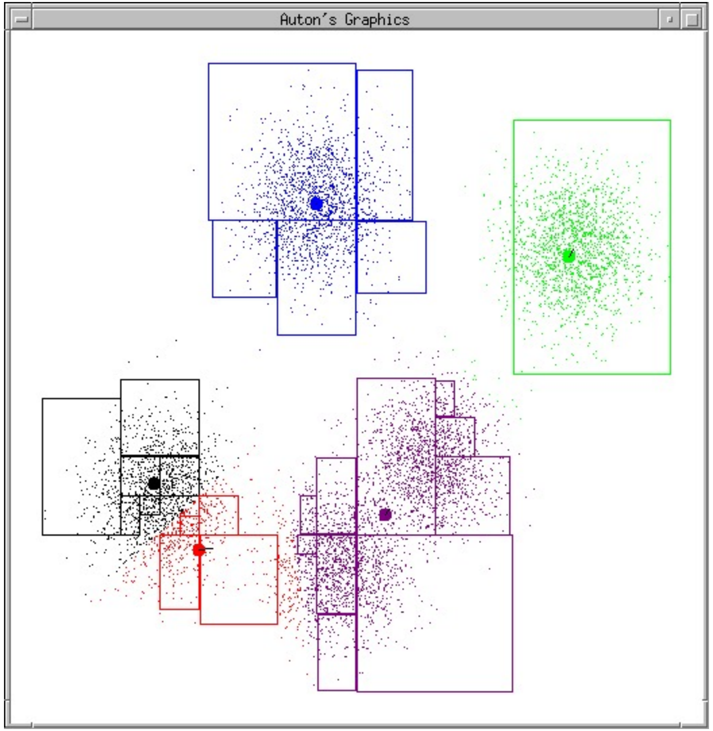
Copyright © 2001, 2004, Andrew W. Moore

K-Means: Part IX

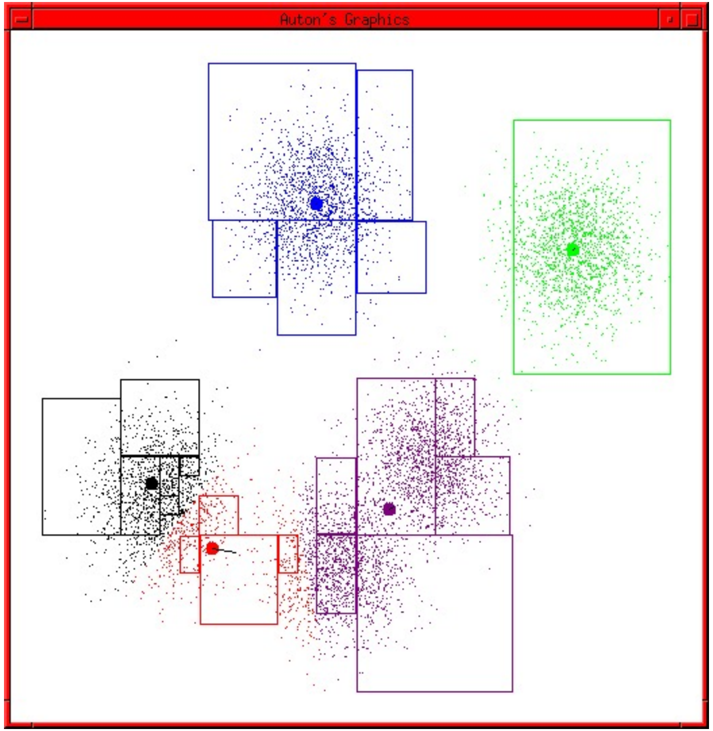


Copyright © 2001, 2004, Andrew W. Moore

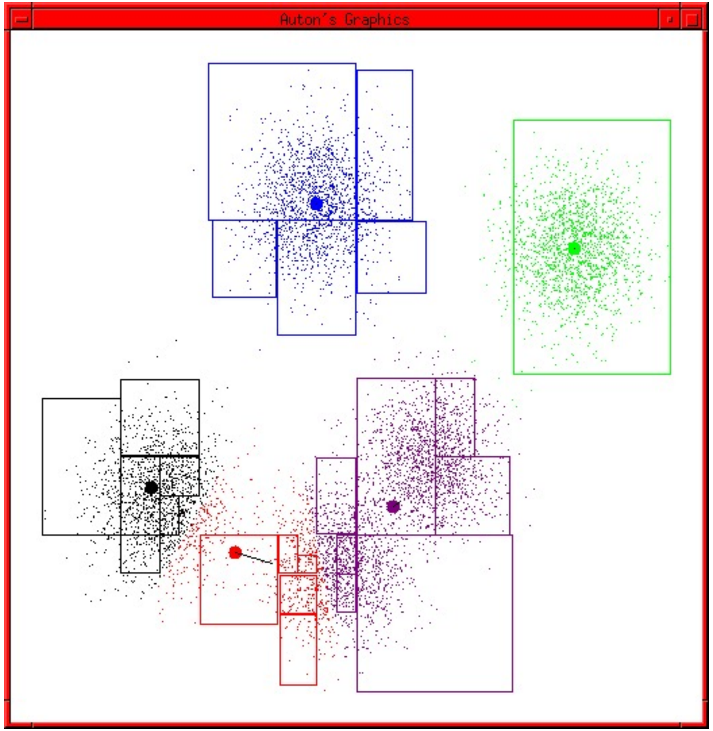
K-Means: Part X



K-Means: Part XI

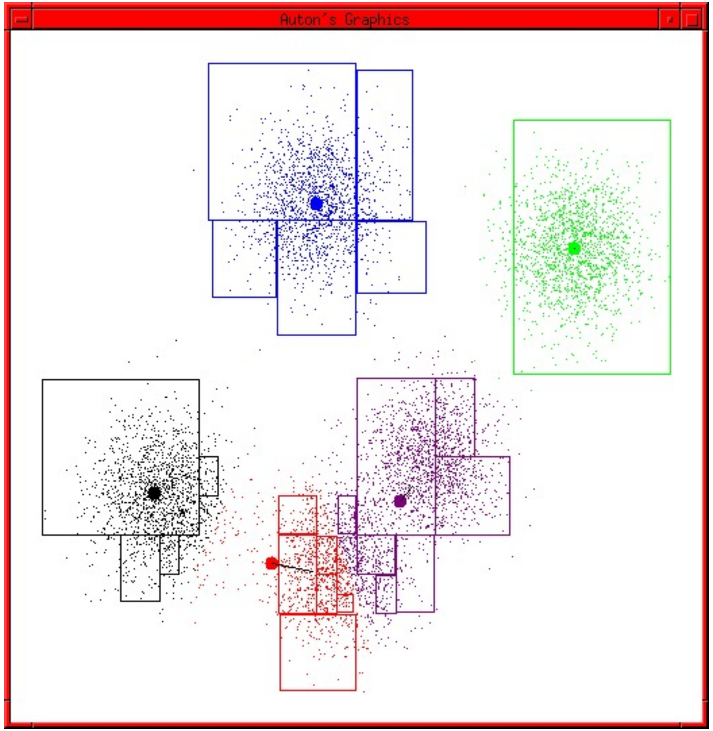


K-Means: Part XII

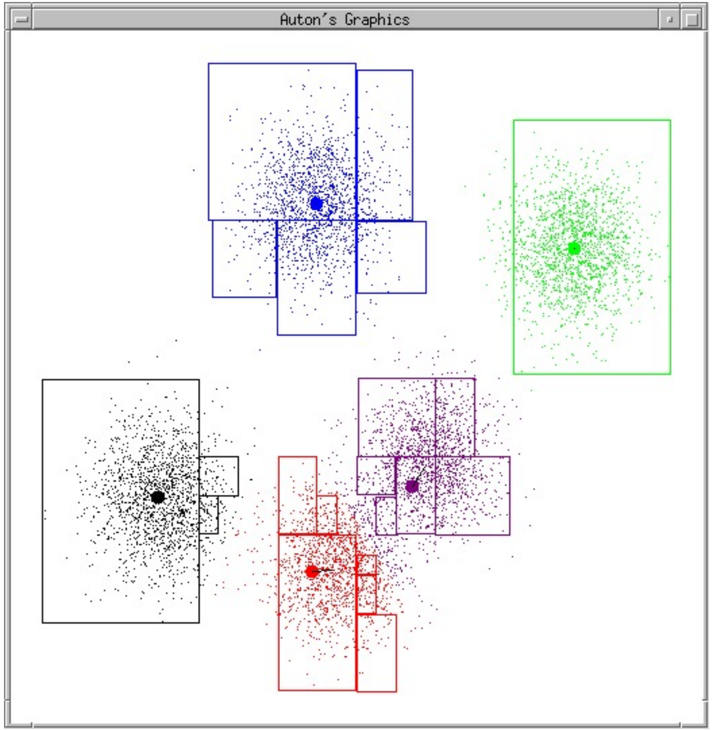


Copyright © 2001, 2004, Andrew W. Moore

K-Means: Part XIII

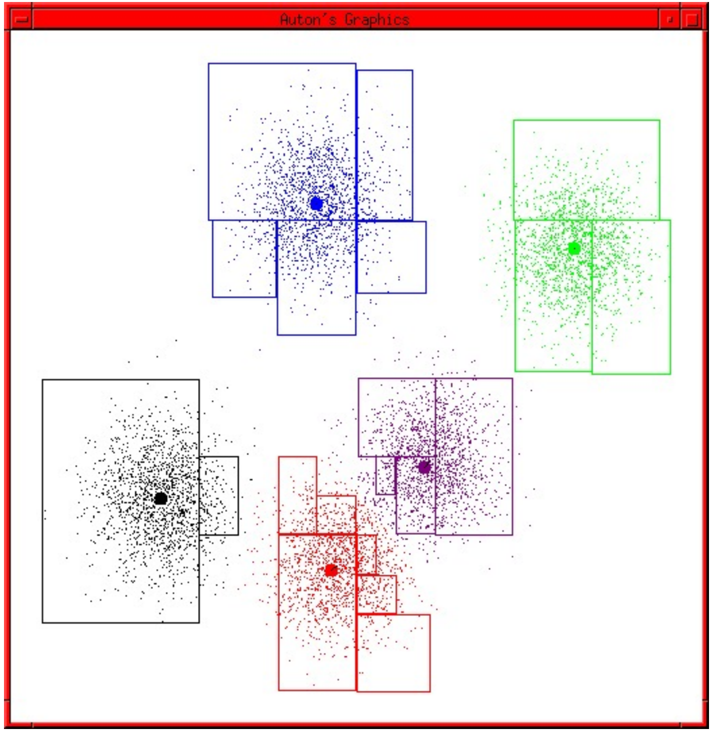


K-Means: Part XIV



Copyright © 2001, 2004, Andrew W. Moore

K-Means Terminates


Auton's Graphics