Data Structures for Fast K-means

The Auton Lab

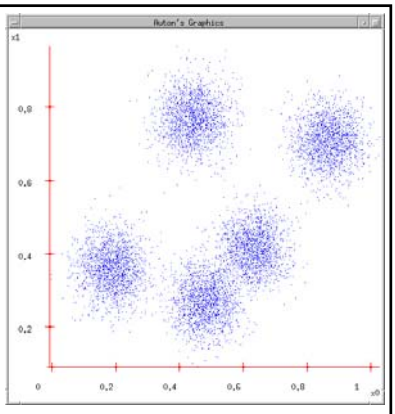
Carnegie Mellon University

www.autonlab.org

What does k-means do?

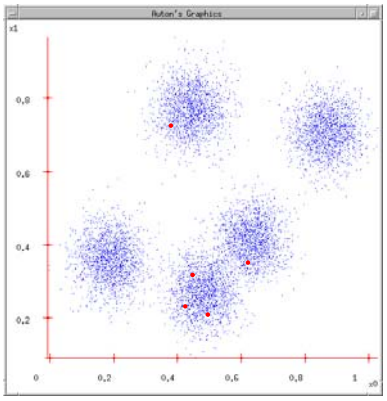
K-means

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



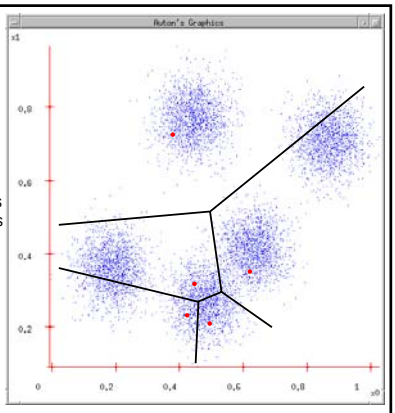
K-means

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



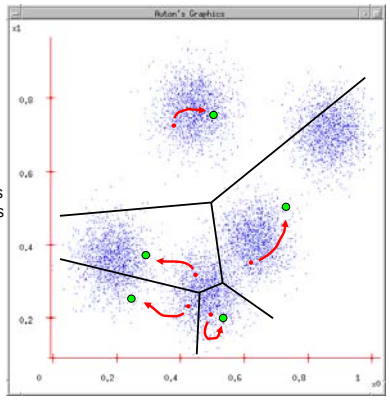
K-means

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)



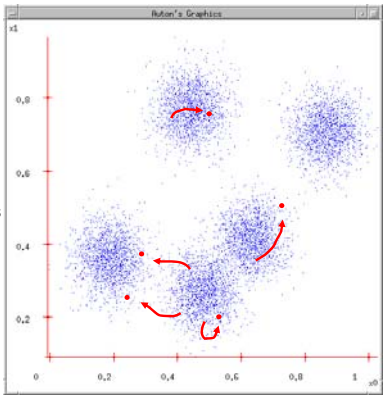
K-means

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

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!



7

K-means

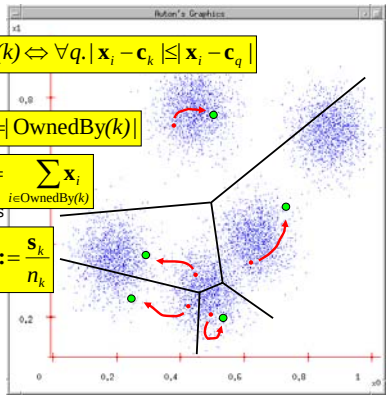
1. As clusters they'd like. *(e.g. k=5)*
2. Randomly gue cluster Center locations
3. Each datapoint out which Center it's closest to.
4. Each Center fin the centroid of points it owns

$$i \in \text{OwnedBy}(k) \Leftrightarrow \forall q. |\mathbf{x}_i - \mathbf{c}_k| \leq |\mathbf{x}_i - \mathbf{c}_q|$$

$$n_k = |\text{OwnedBy}(k)|$$

$$\mathbf{s}_k = \sum_{i \in \text{OwnedBy}(k)} \mathbf{x}_i$$

$$\mathbf{c}_k := \frac{\mathbf{s}_k}{n_k}$$



8

K-means search guts

I must compute $\Sigma \mathbf{x}_i$ of all points I own

I must compute $\Sigma \mathbf{x}_i$ of all points I own

I must compute $\Sigma \mathbf{x}_i$ of all points I own

I must compute $\Sigma \mathbf{x}_i$ of all points I own

9

K-means search guts

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

10

K-means search guts

I will compute $\Sigma \mathbf{x}_i$ of all points I own in left subrectangle, then right subrectangle, then add 'em

I will compute $\Sigma \mathbf{x}_i$ of all points I own in left rectangle, then right subrectangle, then add 'em

I will compute $\Sigma \mathbf{x}_i$ of all points I own in left rectangle, then right subrectangle, then add 'em

I will compute $\Sigma \mathbf{x}_i$ of all points I own in left rectangle, then right subrectangle, then add 'em

11

In recursive call...

1. Find center nearest to rectangle
2. For each other center: can it own any points in rectangle?

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

12

Pruning

I will just grab $\Sigma \mathbf{x}_i$ from the cached value in the node

1. Find center nearest to rectangle
2. For each other center: can it own any points in rectangle?
3. If not, PRUNE

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

13

Pruning not possible at previous level...

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

1. Find center nearest to rectangle
2. For each other center: can it own any points in rectangle?
3. If yes, recurse...

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

But.. maybe there's some optimization possible anyway?

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

14

Blacklisting

1. Find center nearest to rectangle
2. For each other center: can it own any points in rectangle?
3. If yes, recurse...

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

A hopeless center never needs to be considered in any recursion

I will compute $\Sigma \mathbf{x}_i$ of all points I own in rectangle

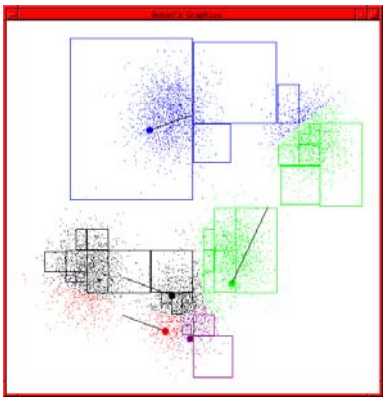
15

Example

Advance apologies: in Black and White this example will deteriorate

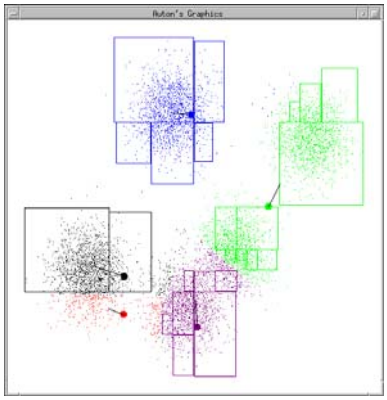
Example generated by:

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)



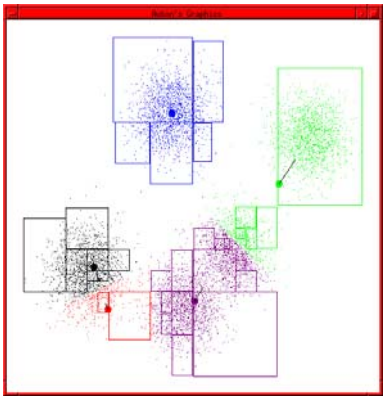
16

K-means continues ...



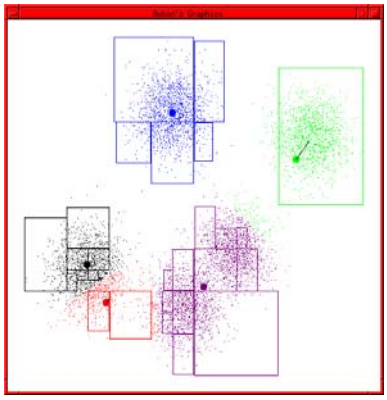
17

K-means continues ...

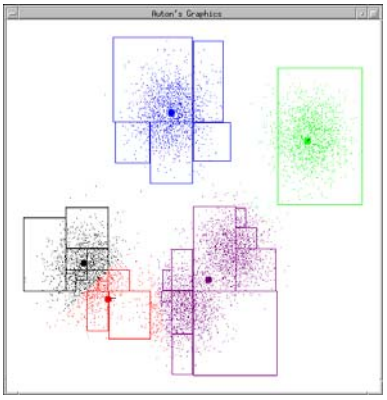


18

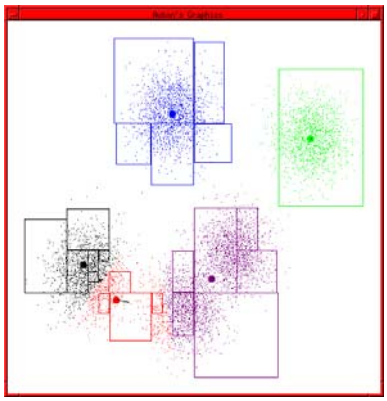
K-means continues ...



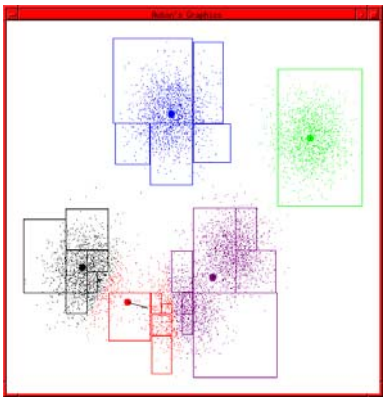
K-means continues ...



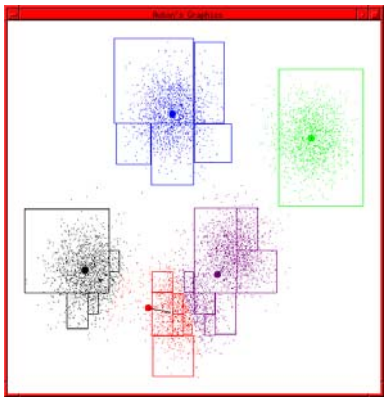
K-means continues ...



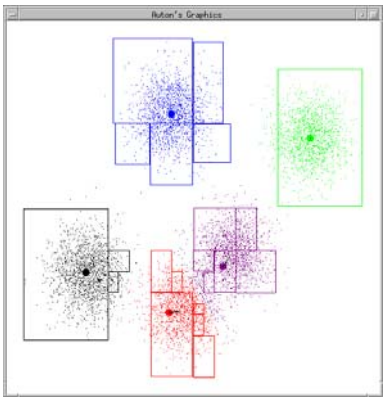
K-means continues ...



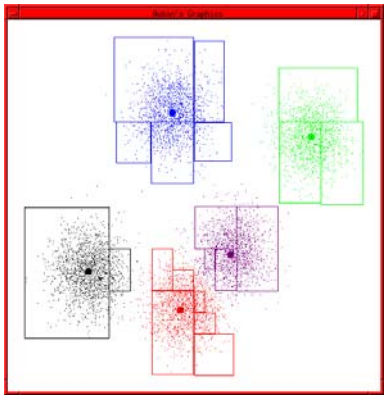
K-means continues ...



K-means continues ...



K-means terminates

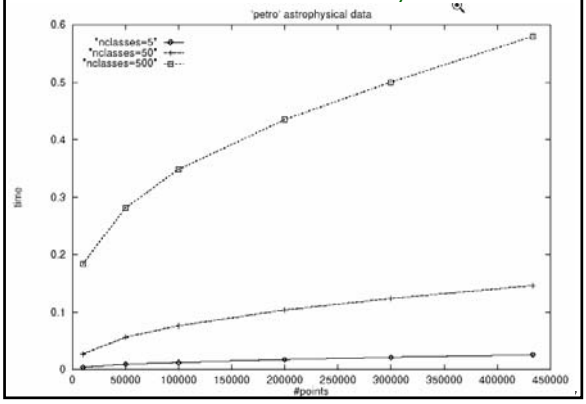


Comparison to a linear algorithm

points	blacklisting	naive	speedup
50000	2.02	52.22	25.9
100000	2.16	134.82	62.3
200000	2.97	223.84	75.3
300000	1.87	328.80	176.3
433208	3.41	465.24	136.6

Astrophysics data (2-dimensions)

Performance vs #records, #classes



Performance vs #dimensions, #classes