CPSC 340:
Machine Learning and Data Mining

Hierarchical Clustering

Fall 2015

Admin

- Assignment 2 due Friday:
  - Submit everything (including code) in a PDF file.
- Extra office hours:
  - 3-4pm on Thursdays in X836.
- Calculus and linear algebra terms to review for next week:
  - Vector addition and multiplication: $\alpha x + \beta y$.
  - Inner-product: $x^T y$.
  - Matrix multiplication: Xw.
  - Solving linear systems: $Ax = b$.
  - Matrix inverse: X^{-1}.
  - Norms: $||x||$.
  - Gradient: $\nabla f(x)$.
  - Stationary points: $\nabla f(x) = 0$.
  - Convex functions: $f''(x) \geq 0$.

Parametric vs. Non-Parametric Models

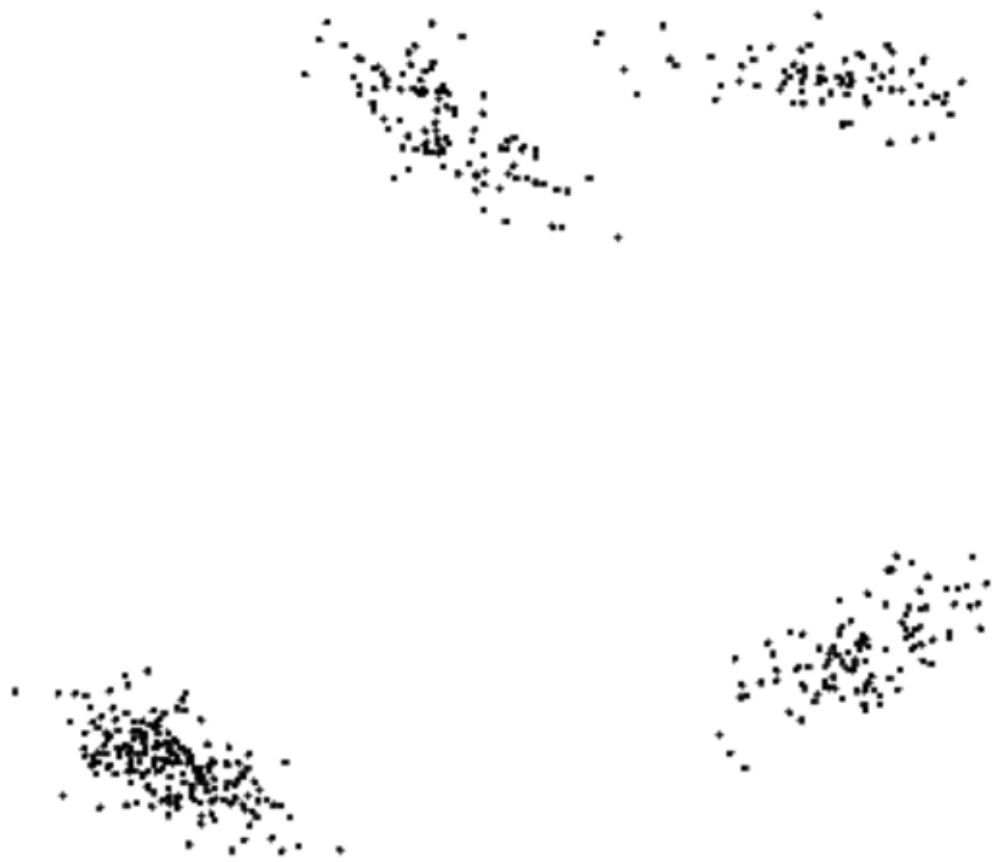
- Let ‘size’ be how much space the model takes to store.
  - For example: if ‘model = decisionTree(X,y)’, how big is ‘model’?
- Parametric models:
  - The ‘size’ of model does not depend on ‘n’ (number of training objects).
- Non-parametric models:
  - The ‘size’ of the model depends on the number of training objects ‘n’.
- Examples:
  - Fixed-depth decision tree: ‘size’ of the model is O(1) in terms of ‘n’.
  - K-nearest neighbours: ‘size’ of the model is O(n) in terms of ‘n’.

	Parametric:	Non-parametric:
Supervised:	decision trees naive Bayes random forest	KNN
Unsupervised:	K-means	density-based clustering Scatterplot

Cost of Density-Based Clustering

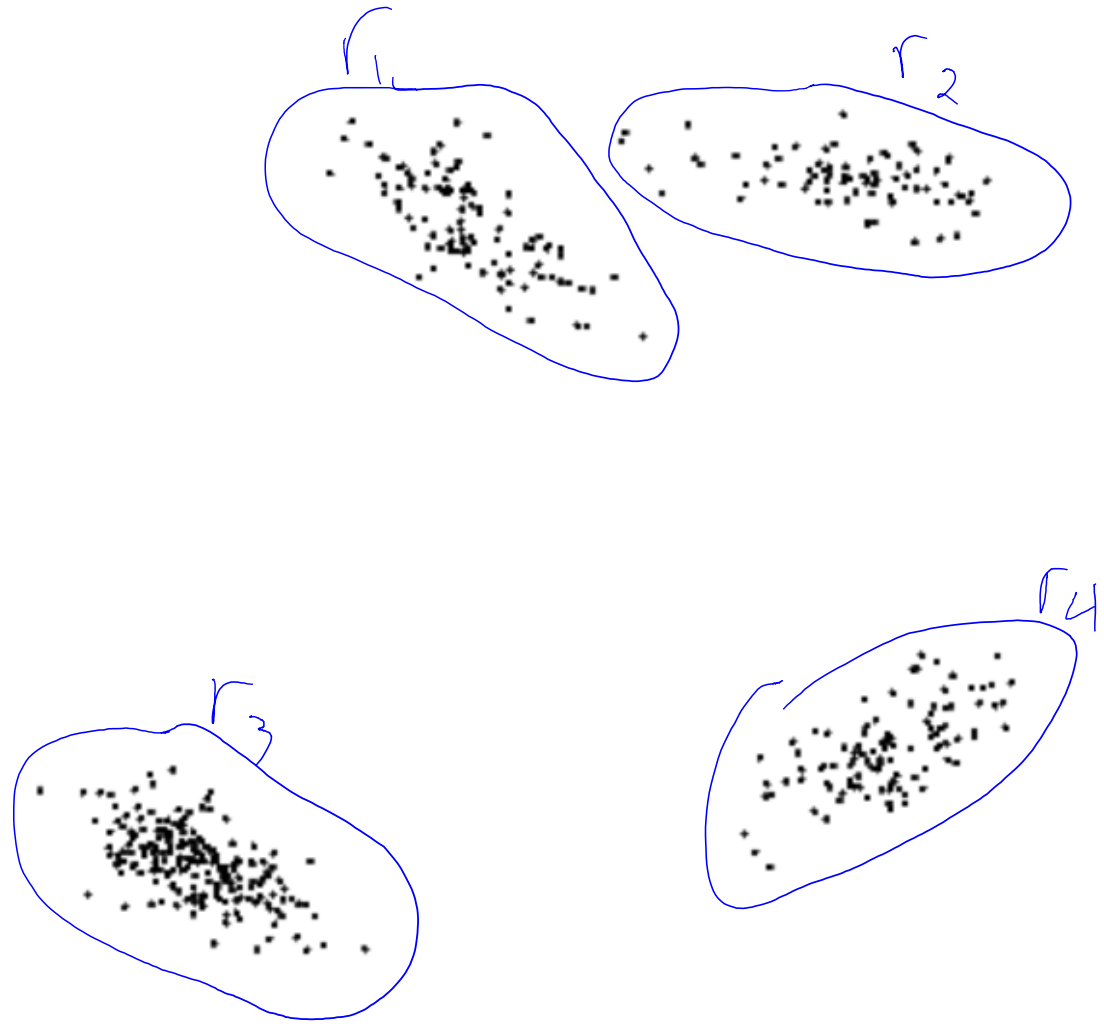
- To run DBSCAN, we need distance between all pairs of objects.
  - Cost of computing distance with ‘d’ features is O(d).
  - There are $O(n^2)$ pairs of objects.
  - So cost of computing all pairs of distances is $O(n^2d)$.
- Given the distances, total cost of all expand operations is $O(n^2)$.
- Not feasible is ‘n’ is large.
- How can we answer ‘closest points’ queries when ‘n’ is large?

Distance-Based Pruning



Distance-Based Pruning

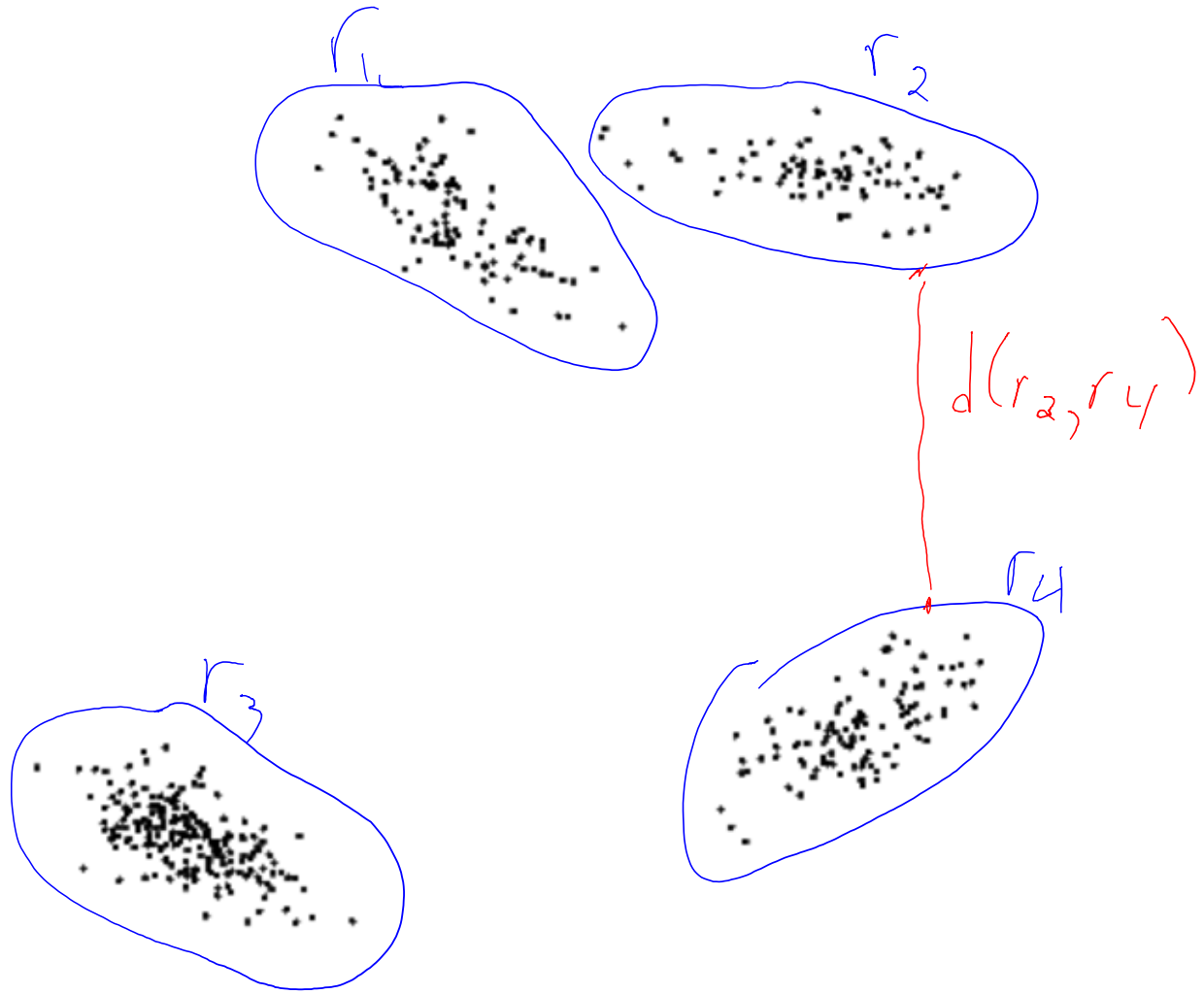
Divide objects into regions r_c.



Distance-Based Pruning

Divide objects into regions r_c.

Let $d(r_1,r_2)$ be minimum distance between any part of region r_1 and r_2.

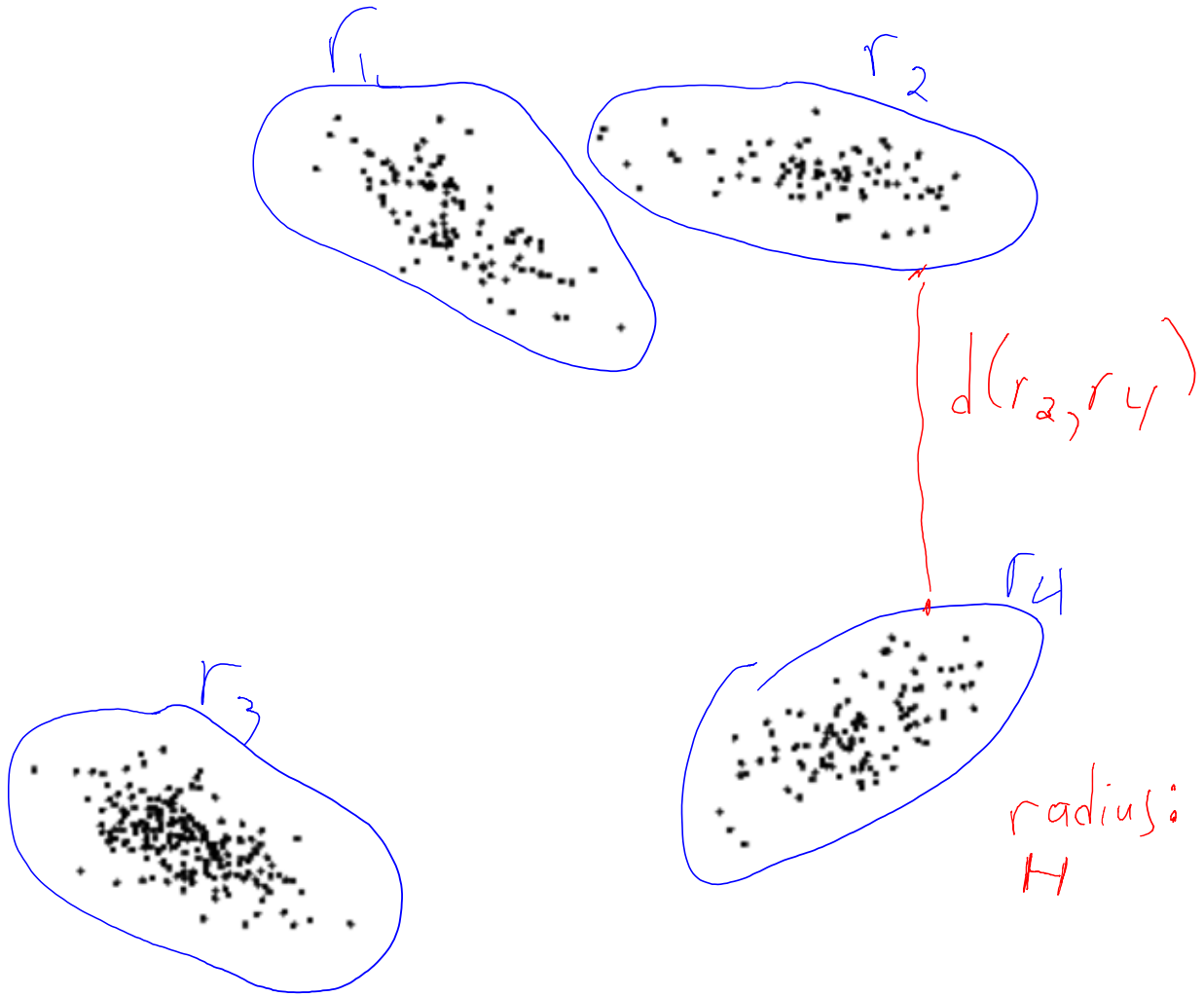


Distance-Based Pruning

Divide objects into regions r_c.

Let $d(r_1,r_2)$ be minimum distance between any part of region r_1 and r_2.

If x_i is in r_1 and x_j is in r_2, then $d(x_1,x_2) \geq d(r_1,r_2)$.

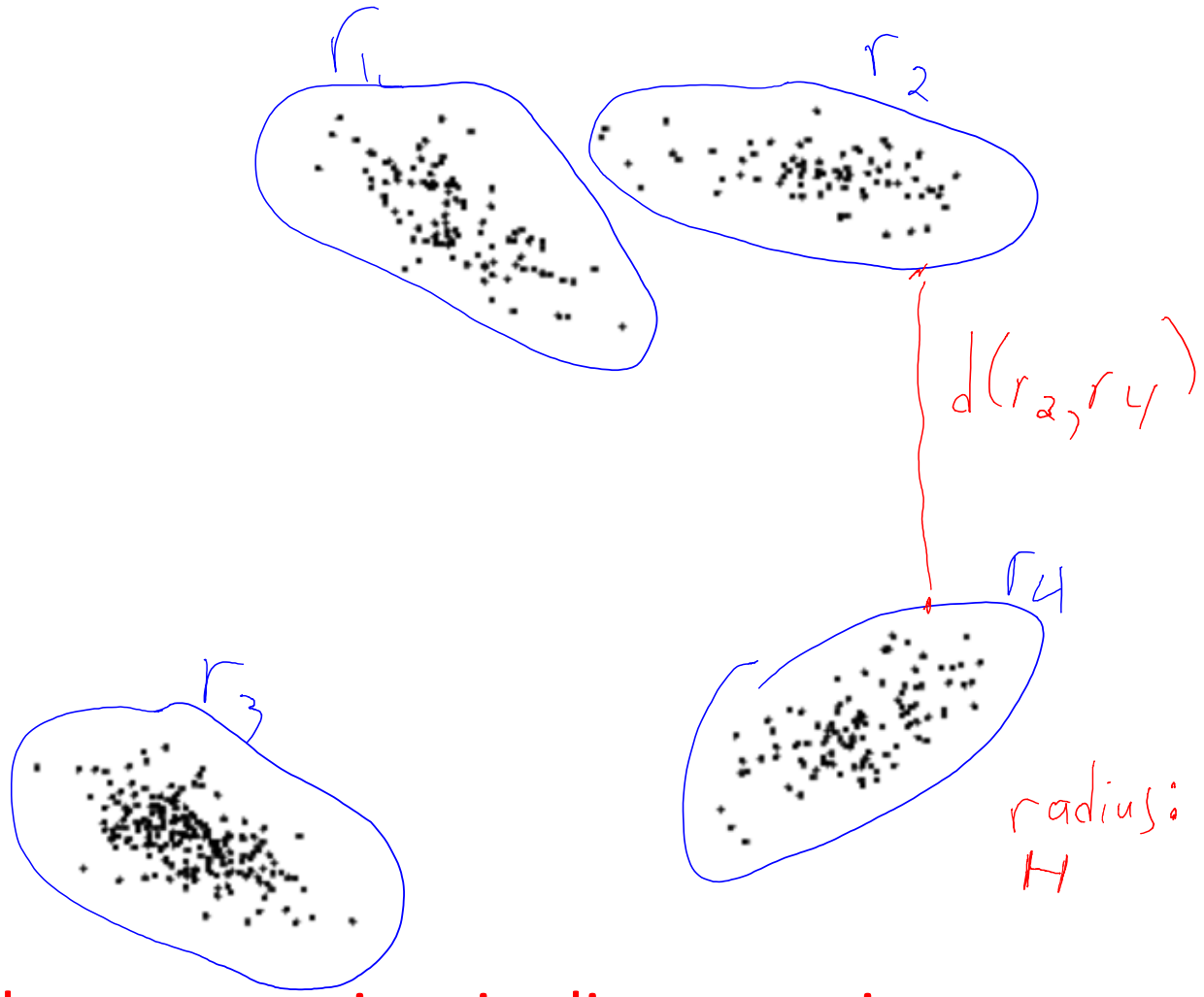


Distance-Based Pruning

Divide objects into regions r_c.

Let $d(r_1,r_2)$ be minimum distance between any part of region r_1 and r_2.

If x_i is in r_1 and x_j is in r_2, then $d(x_1,x_2) \geq d(r_1,r_2)$.



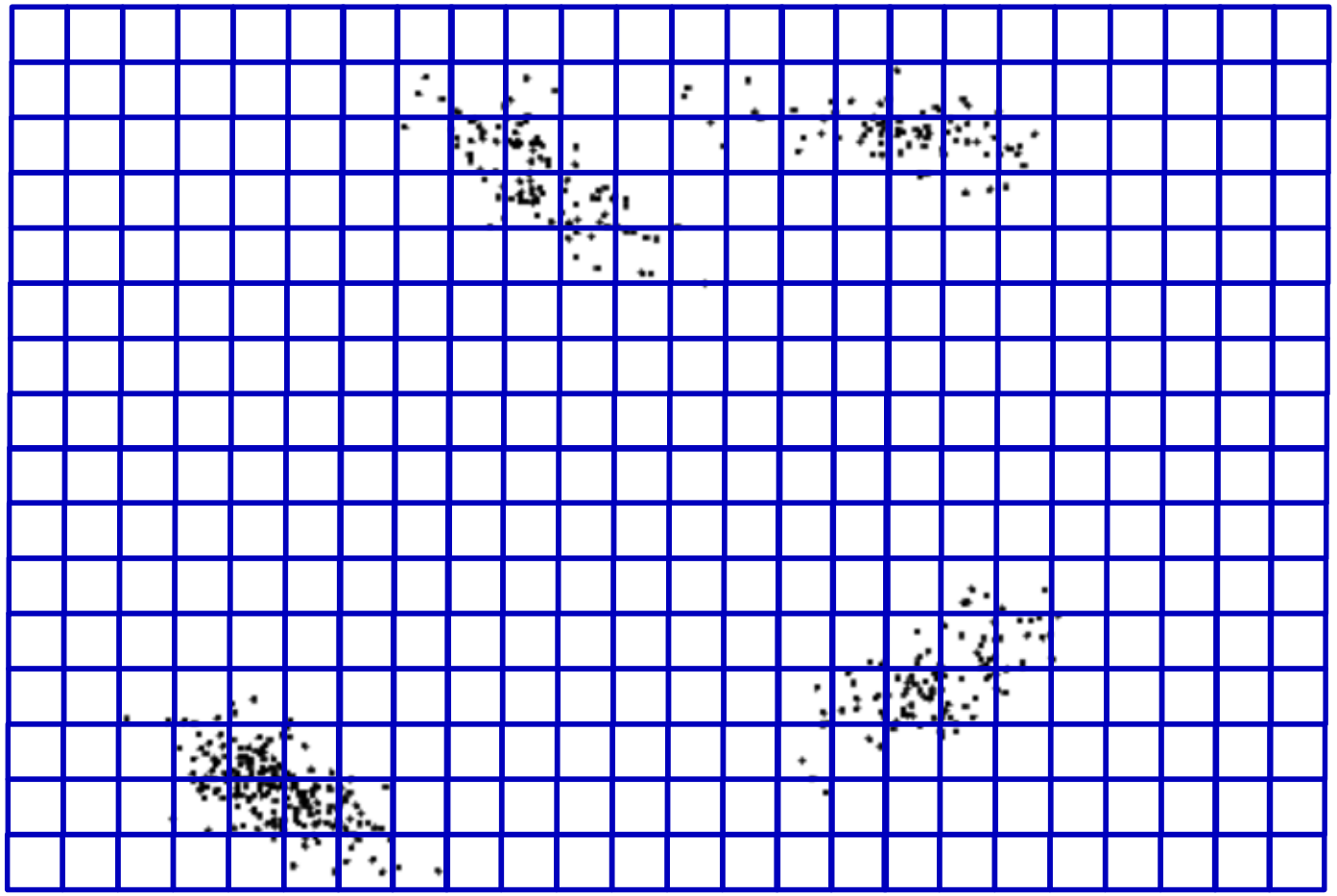
Don't have to compute distances between points in distant regions.

Grid-Based Pruning

- Assume we want to find objects within a distance of ‘r’.

Divide space into squares of length r.

Assign each object to region.

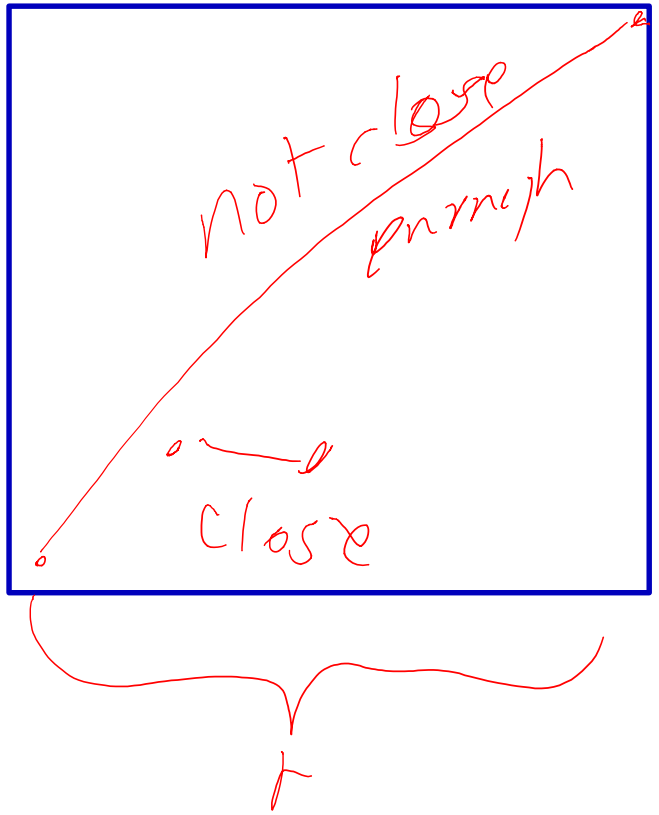


Square-Shaped Regions

- A computationally-nice region for two-dimensional data:
  - Square with side length of ‘r’.

Square-Shaped Regions

- A computationally-nice region for two-dimensional data:
  - Square with side length of ‘r’.

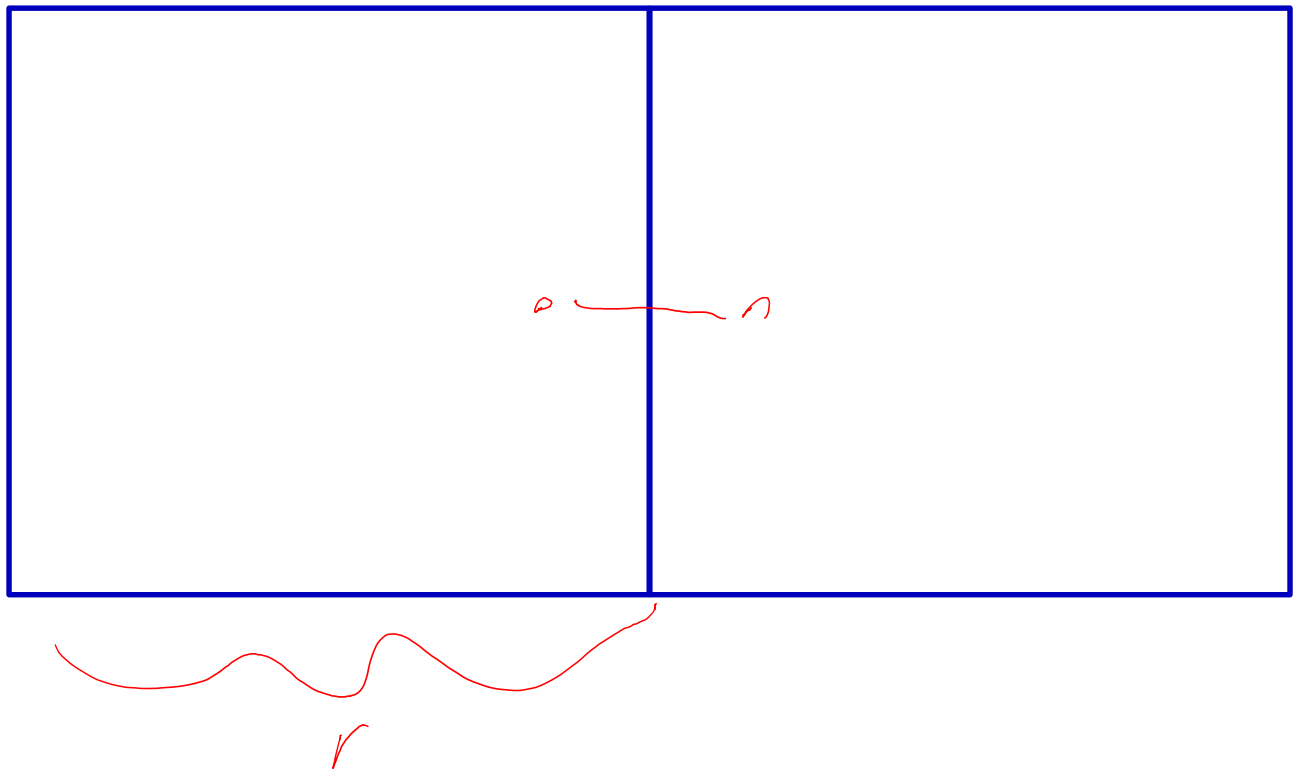


Points inside square can have distance less than ‘r’.

Square-Shaped Regions

- A computationally-nice region for two-dimensional data:
  - Square with side length of ‘r’.

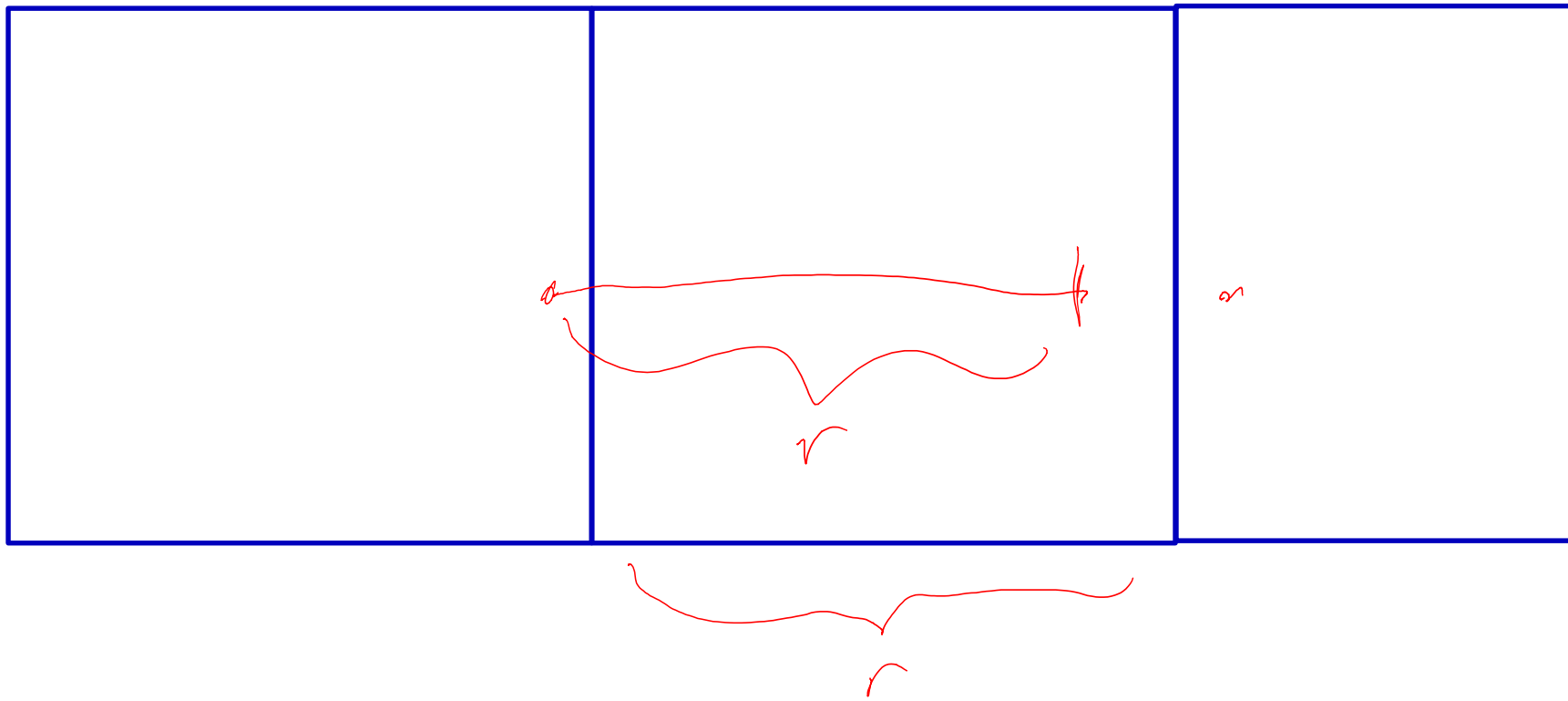
Points in adjacent squares can also have distance ‘r’.



Square-Shaped Regions

- A computationally-nice region for two-dimensional data:
  - Square with side length of ‘r’.

Non-neighbouring squares must have Distance at least ‘r’.

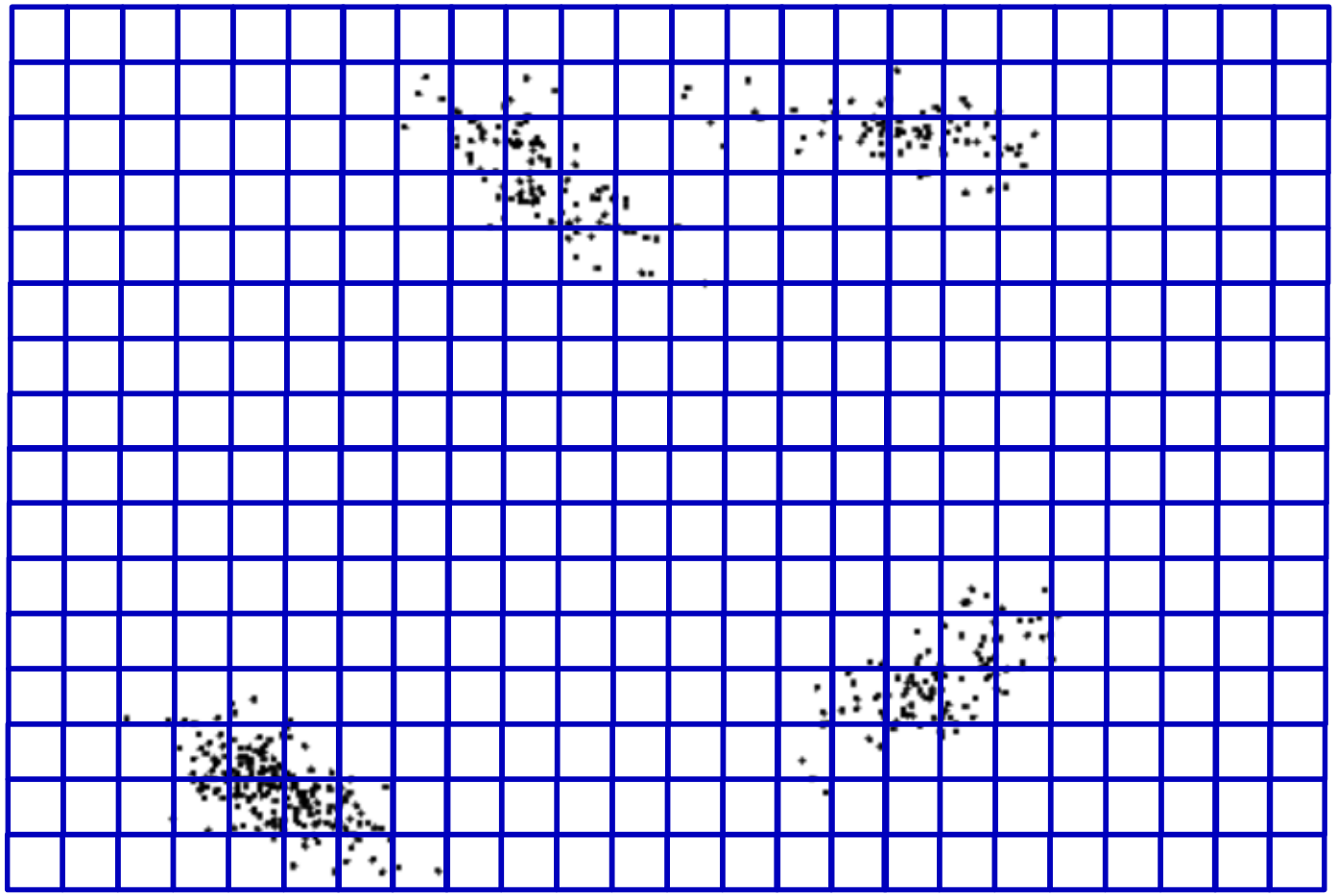


Grid-Based Pruning

- Assume we want to find objects within a distance of ‘r’.

Divide space into squares of length r.

Assign each object to region.

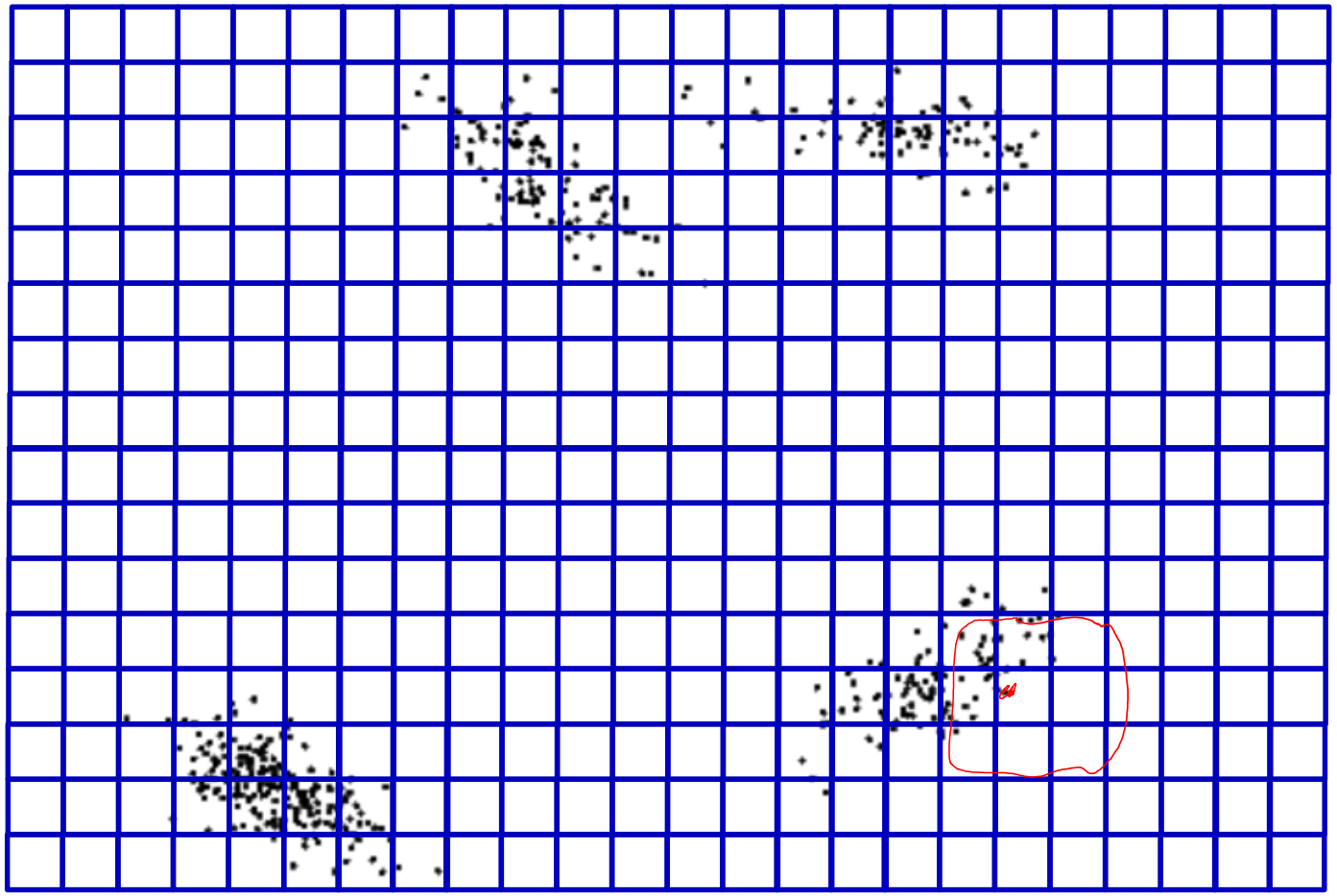


Grid-Based Pruning

- Assume we want to find objects within a distance of ‘r’.

Divide space into squares of length r.

Assign each object to region.

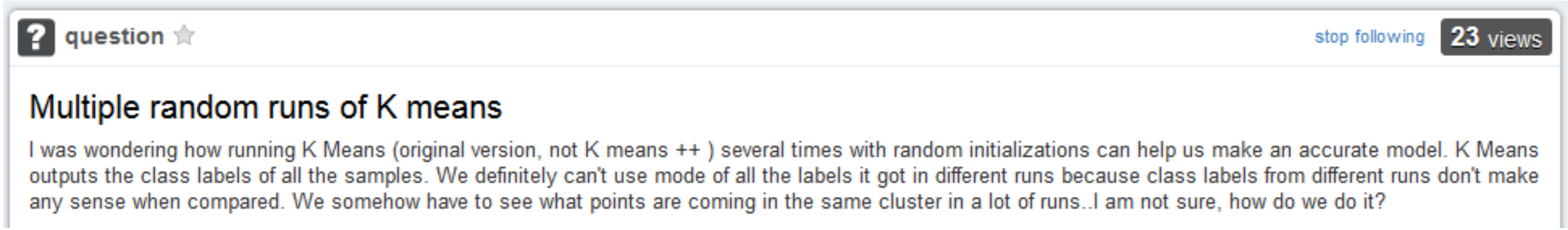


- Check objects in same region.
- Check objects in adjacent regions.

Curse of Dimensionality

- Works great in low dimensions, if not all points in adjacent squares.
- In high-dimensions:
  - Hyper-cubes of length ‘r’ are tiny: few points will fall into each cube.
  - Number of adjacent cubes increases exponentially:
    - 8 in 2D, 26 in 3D, 80 in 4D, 252 in 5D, 3^d-1 in n-D.
- Could make bigger hyper-cubes that include many points:
  - Now number of within-cube distance calculations can get big.
- Keywords for further reading: quad-trees, R-trees, ball-trees.
  - Can find exact nearest neighbours without knowing ‘r’.
  - Worst case performance is still bad in high dimensions.
- Fast but approximate nearest neighbours: locality-sensitive hashing.
  - Random transformations into low-dimensional spaces.

Ensemble Clustering


question

stop following 23 views

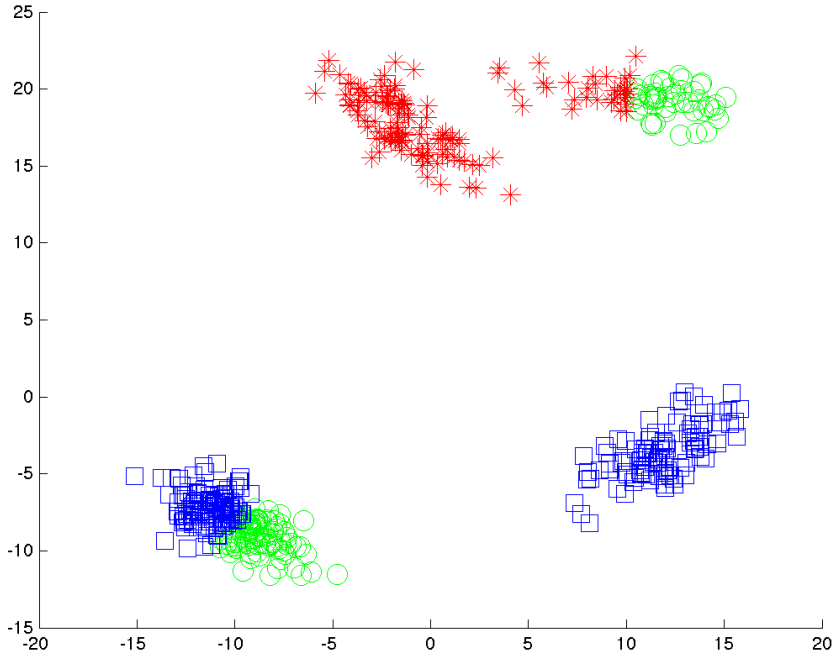
Multiple random runs of K means

I was wondering how running K Means (original version, not K means ++) several times with random initializations can help us make an accurate model. K Means outputs the class labels of all the samples. We definitely can't use mode of all the labels it got in different runs because class labels from different runs don't make any sense when compared. We somehow have to see what points are coming in the same cluster in a lot of runs..I am not sure, how do we do it?

- We can consider ensemble methods for clustering.
  - "Consensus clustering"
- It's a good/important idea:
  - Bootstrapping is widely-used.
  - "Do clusters change if the data was slightly different?"
- But we need to be careful about how we combine models.

Ensemble Clustering

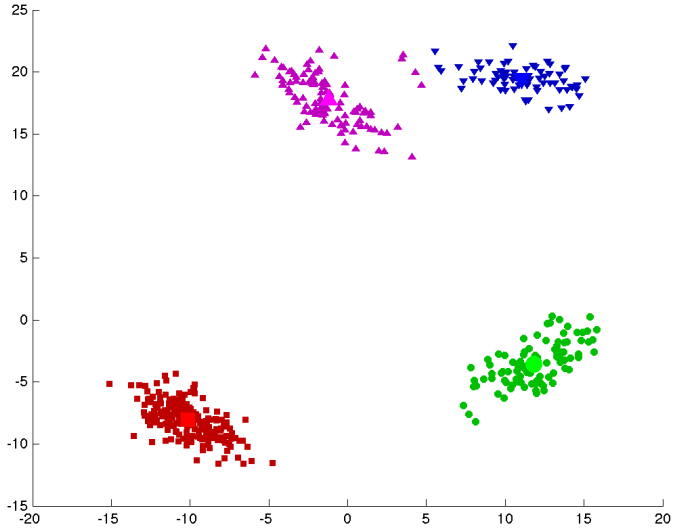
- E.g., run k-means 20 times and then cluster using the mode.
- Normally, averaging across models doing different things is good.

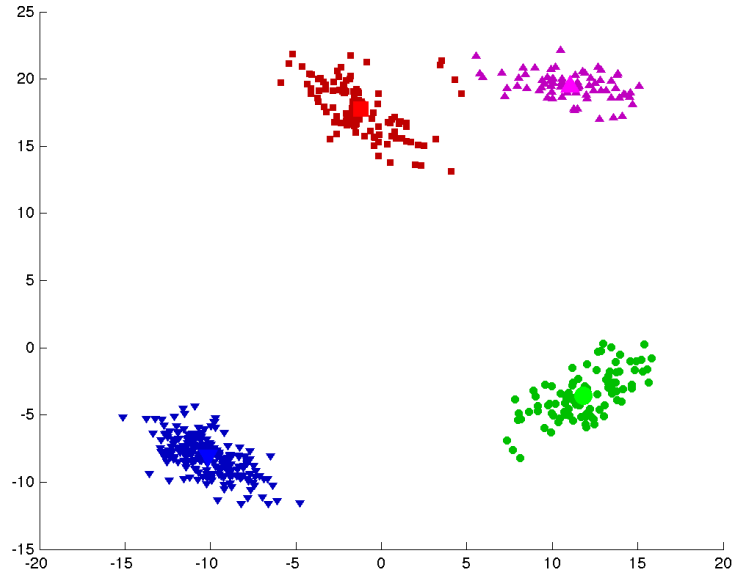


- But this is a bad ensemble method: worse than k-means on its own.

Label Switching Problem

- This doesn't work because of label switching problem.
  - The cluster labels are meaningless.
  - We could permute the labels and get the same clustering:

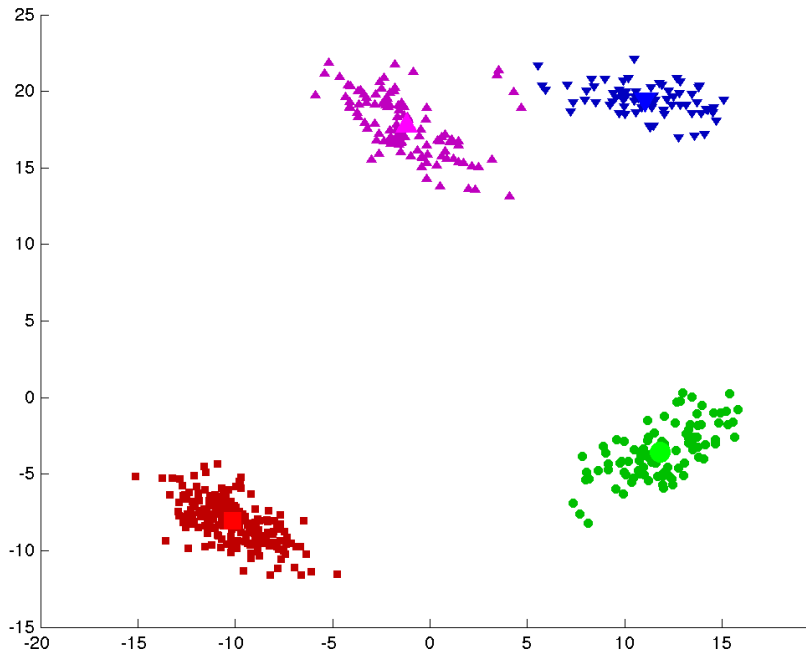


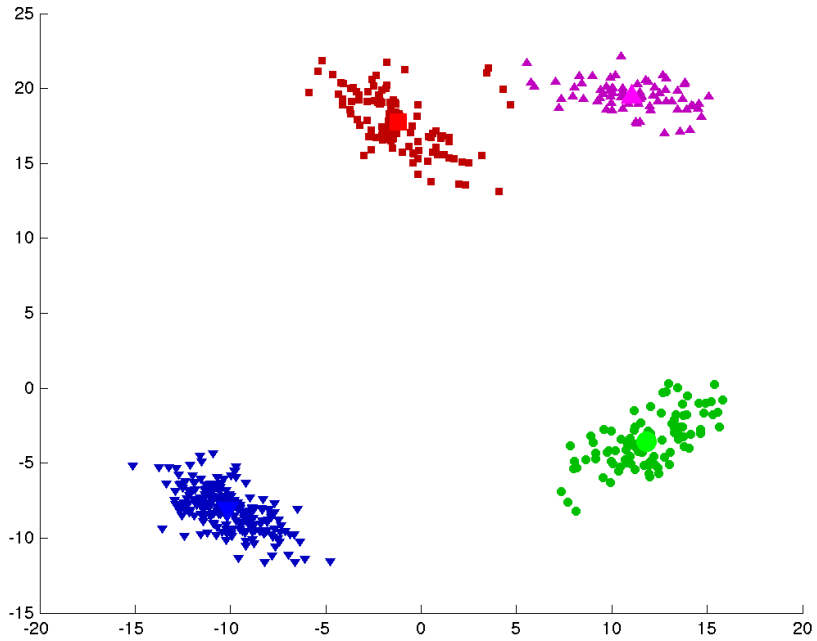


  - All clusters become equally likely as number of initializations increases.

Solutions to Label Switching Problem

- Alternative is avoid depending on specific labels:
  - Don't ask "is point x_i in red square cluster?", which is meaningless.
  - Ask "is point x_i in the same cluster as x_j?", which is meaningful.





UBClustering Algorithm

- Let's define a new ensemble clustering method: UBClustering.

1. Run k-means with 'T' different random initializations.
2. For each object i and j:
   - Count the number of times x_i and x_j are in the same cluster.
   - Define p(i,j) = count(x_i in same cluster as x_j)/T.
3. Put x_i and x_j in the same cluster if p(i,j) > 0.5.

- We'll merge clusters in step 3 if i or j are already assigned.

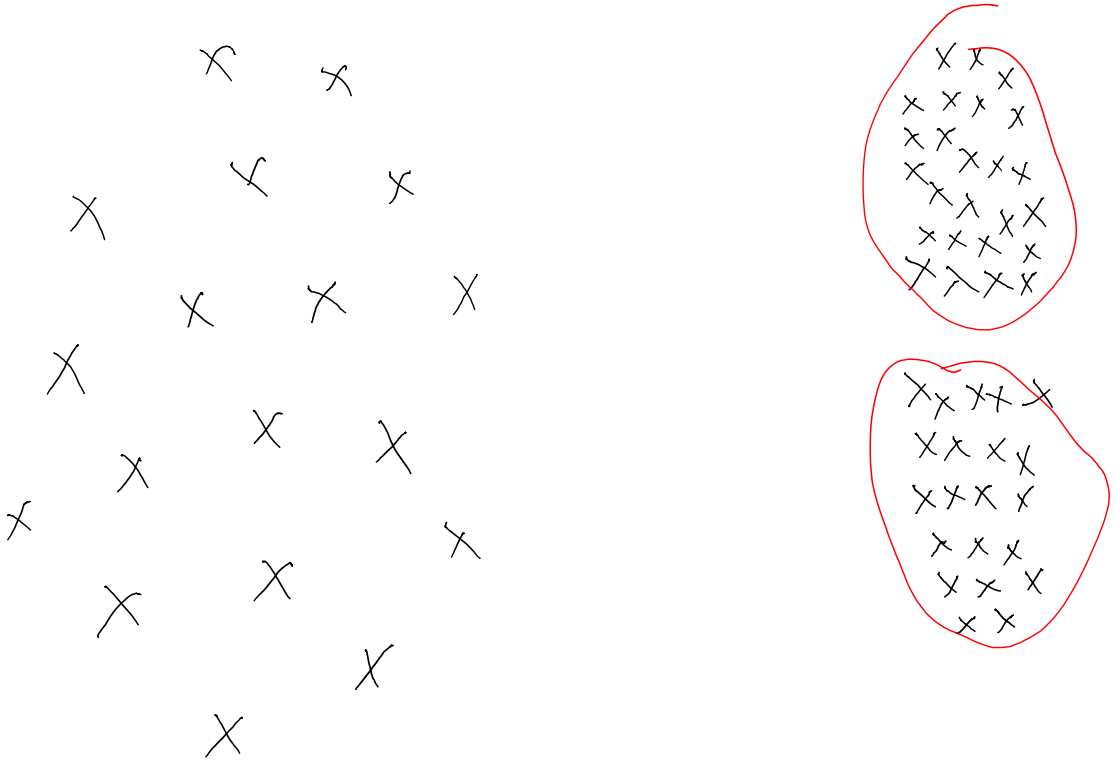
UBClustering Algorithm

- Not all points in same cluster will have $p(i,j) > 0.5$, but each i has some j in same cluster with $p(i,j) > 0.5$.
- Not all points get assigned to a cluster:
  - Some points may not frequently get clustered with any other points.
- You can implement UBClustering using DBSCAN.
  - Q1 on Assignment 3.

UBClustering Algorithm

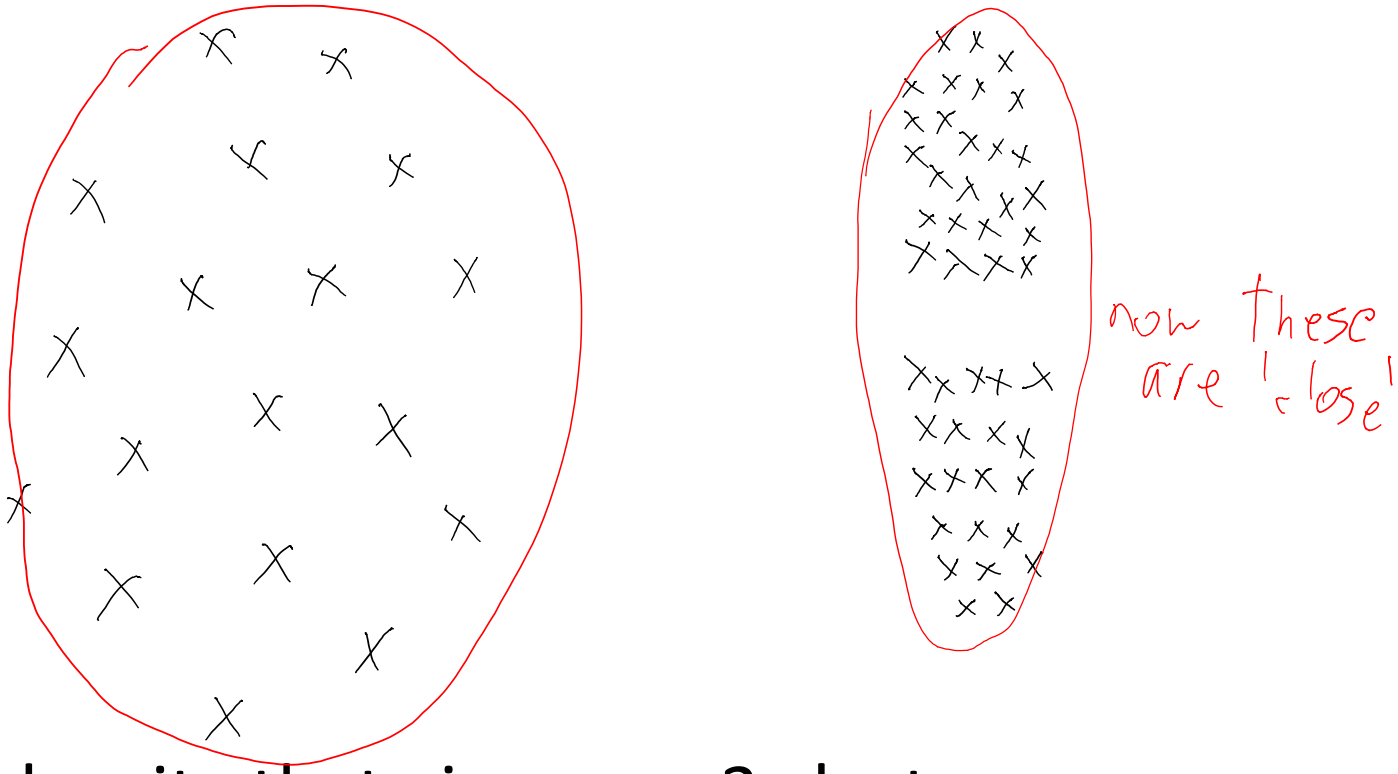
Differing Densities

- Consider density-based clustering on this data:



Differing Densities

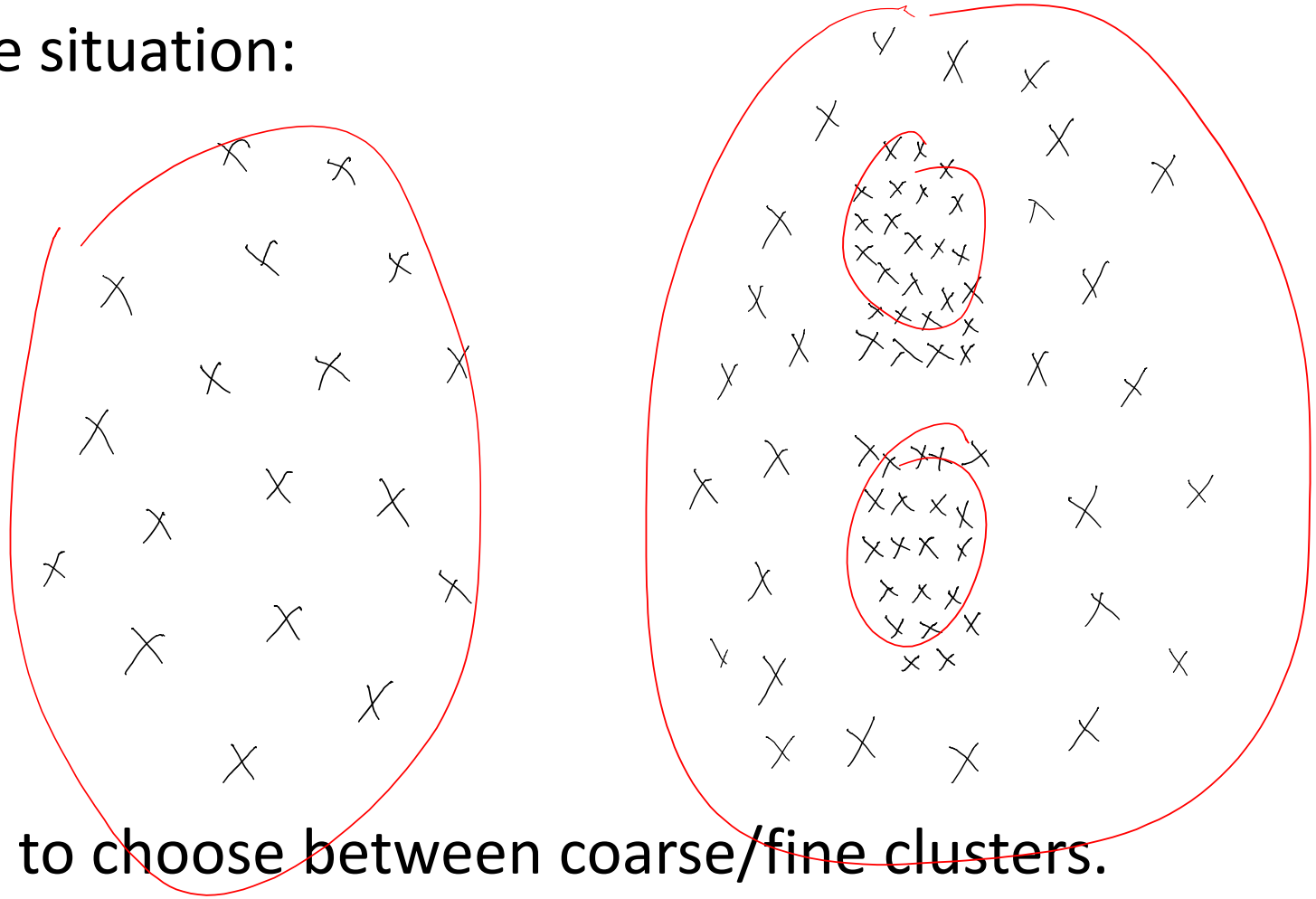
- Consider density-based clustering on this data:



- There may no density that gives you 3 clusters.

Differing Densities

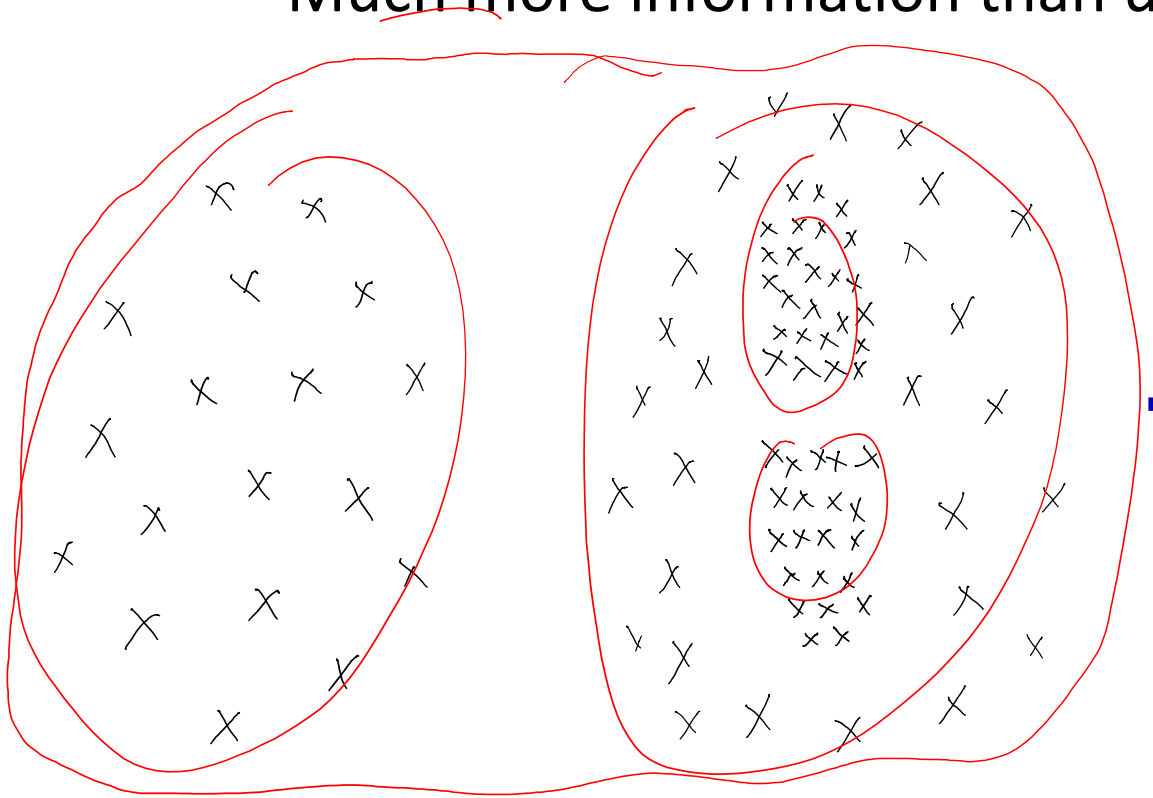
- Here is a worse situation:

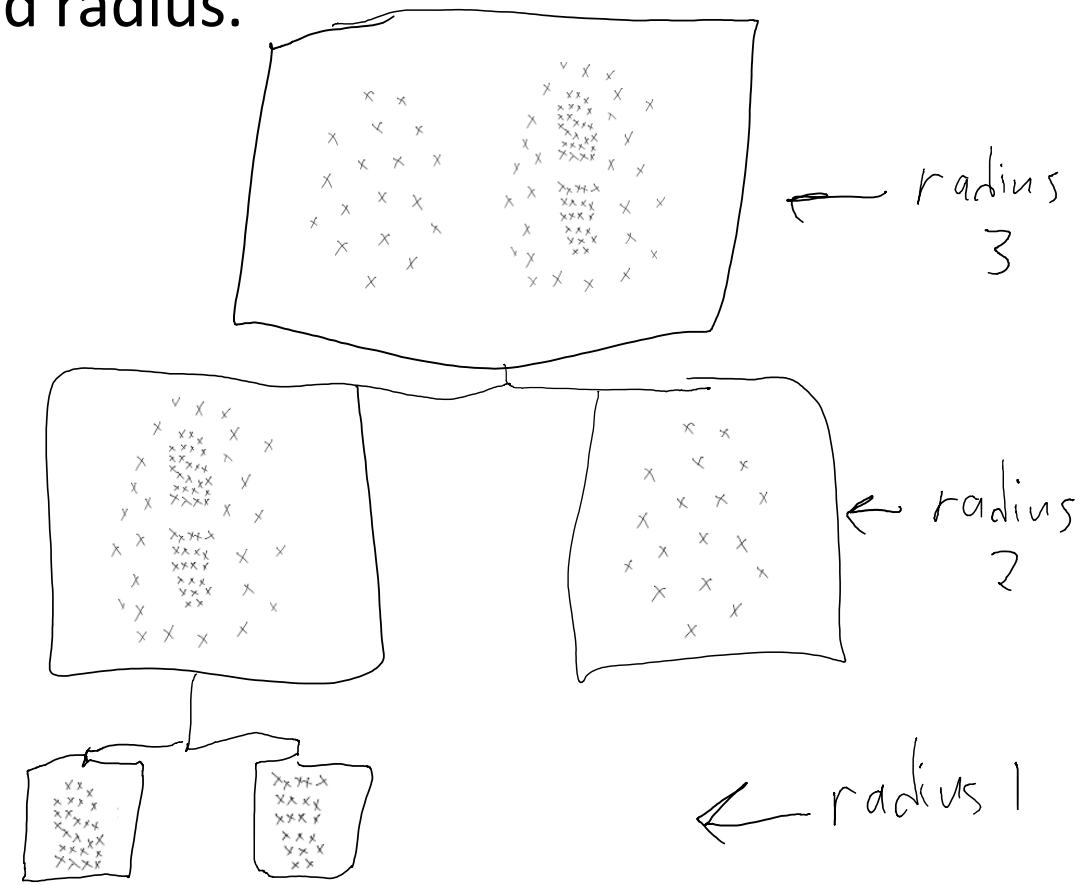


- Now you need to choose between coarse/fine clusters.
- Instead of fixed clustering, we often want hierarchical clustering.

Density-Based Hierarchical Clustering

- A simple way to make a hierarchical DBSCAN:
  - Fix minPoints, record clusters as you vary the radius.
  - Much more information than using a fixed radius.



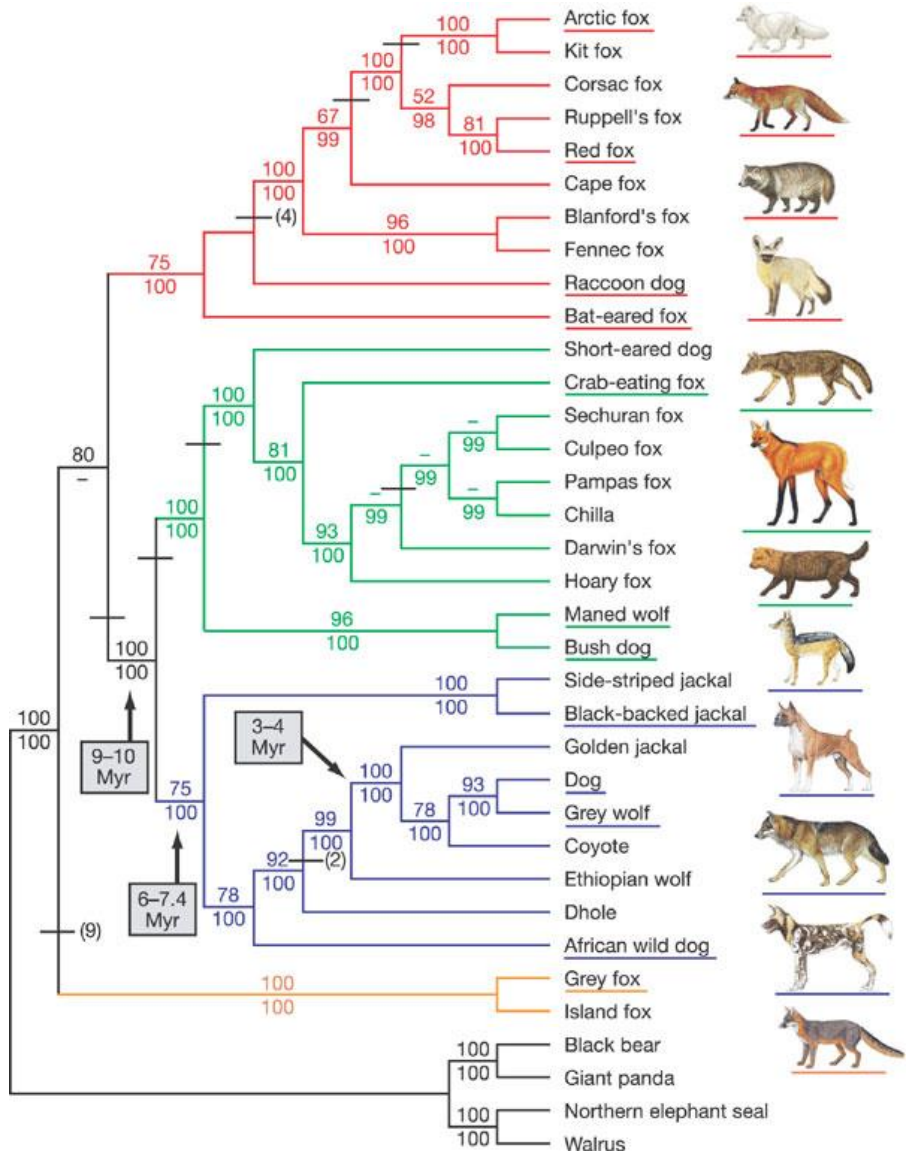


http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=00018e

Application: Phylogenetics

- We sequence the genomes of a set of existing animals (e.g., canids).
- Can we construct the ‘tree of life’?

- Comments on this application:
  - On the right are individuals.
  - As you go left, clusters merge.
  - Merges are ‘common ancestors’.
  - Distance function often approximate how much change needed.
  - Length of lines often approximates time needed for change to happen.
  - Numbers are often #clusterings across bootstrap samples.
  - ‘Outgroups’ (walrus, panda) are a sanity check.



http://www.nature.com/nature/journal/v438/n7069/fig_tab/nature04338_F10.html

Agglomerative (Bottom-Up) Clustering

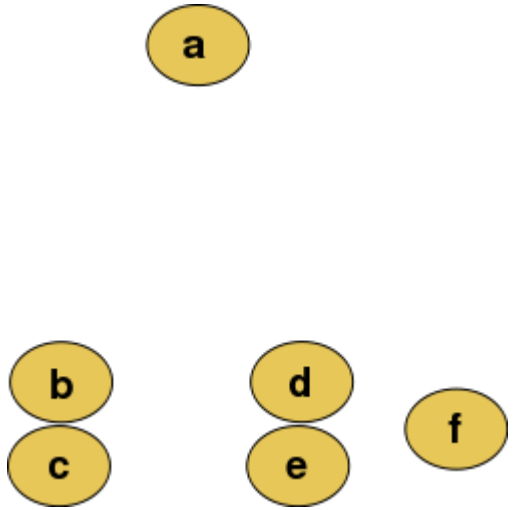
1. Start with each point being its own cluster.
2. Until we only have one cluster left:
   - Merge the two ‘closest’ clusters.

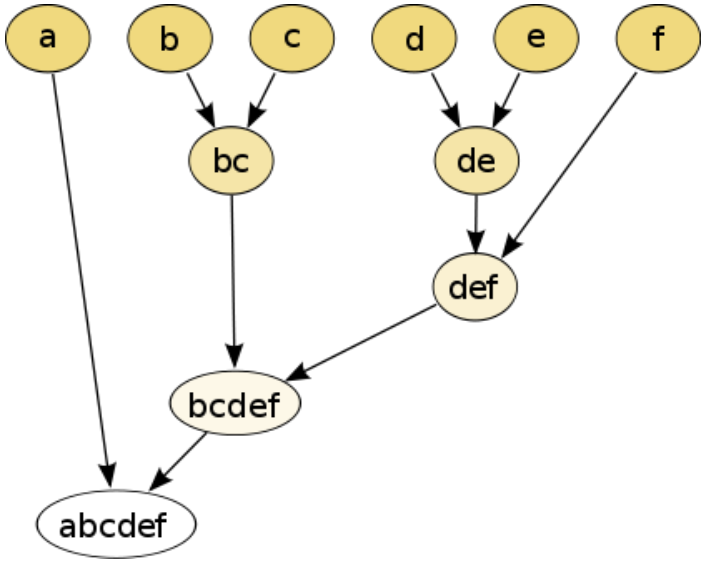
- Reinvented by different fields under different names (UPGMA).
- There are a variety of ways to define ‘closest’ clusters:
  - Single-link: minimum distance between points in clusters.
  - Complete-link: maximum distance between points in clusters.
  - Average-link: average distance between points in clusters.
  - Centroid-link: distance between cluster centers.
  - More exotic: decrease in average distance to cluster center.

Agglomerative (Bottom-Up) Clustering

1. Start with each point being its own cluster.
2. Until we only have one cluster left:
   – Merge the two ‘closest’ clusters.

- Reinvented by different fields under different names (UPGMA).

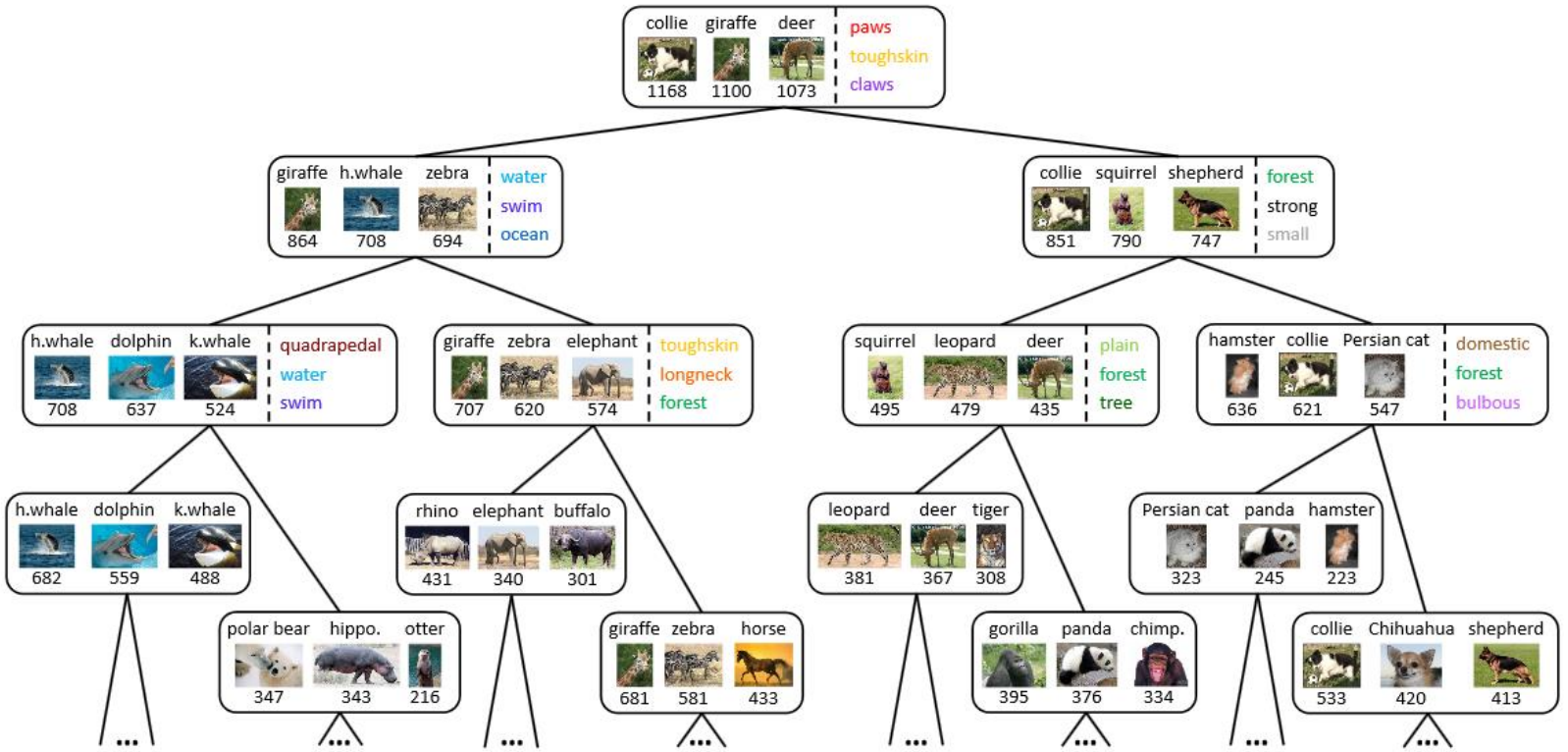




https://en.wikipedia.org/wiki/Hierarchical_clustering

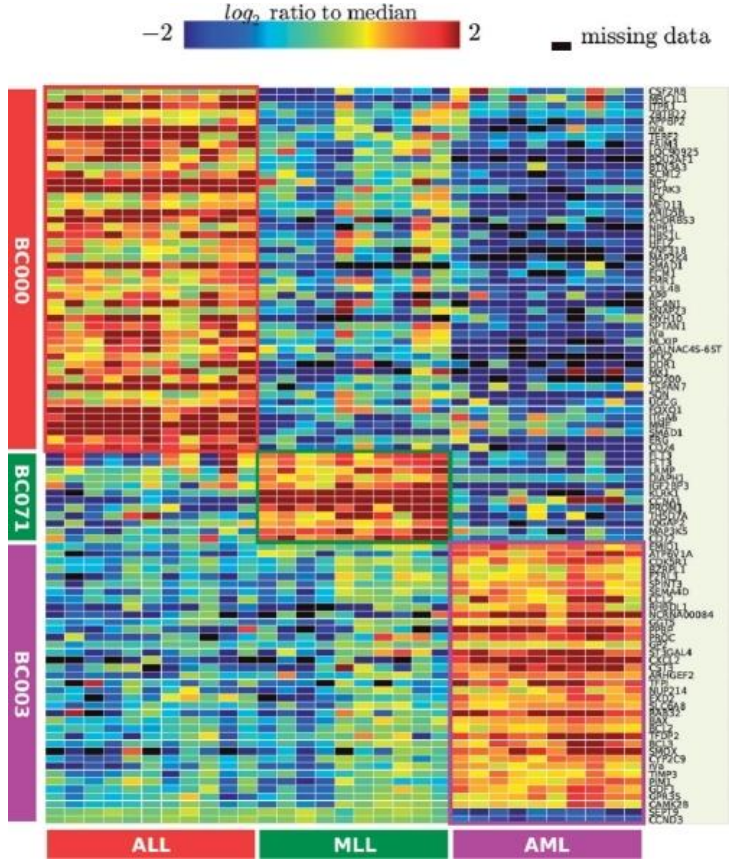
Divisive (Top-Down) Clustering

- Start with all objects in one cluster, then start dividing.
- E.g., run k-means on a cluster, then run again on resulting clusters.
  - A clustering analogue of decision tree learning.



Other Clustering Methods

- Mixture models:
  - Probabilistic clustering.
- Fuzzy c-means:
  - Allows objects to belong to multiple clusters.
- Sparse clustering:
  - Try to form clusters using a subset of the features.
- Biclustering:
  - Cluster objects and features.
- Other families of clustering methods:
  - Mean-shift, graph-based, spectral.



http://openi.nlm.nih.gov/detailedresult.php?img=2731891_gkp491f3&req=4

Summary

- Parametric vs. non-parametric: does size of model depend on ‘n’?
- Region-based pruning: for finding closest objects among huge number of objects.
- Ensemble clustering: can work well (UBClustering) but need to account for label switching.
- Hierarchical clustering: much more informative than static clustering.
- Agglomerative vs. divisive hierarchical clustering.

- Next time: “customers who bought this item also bought”.