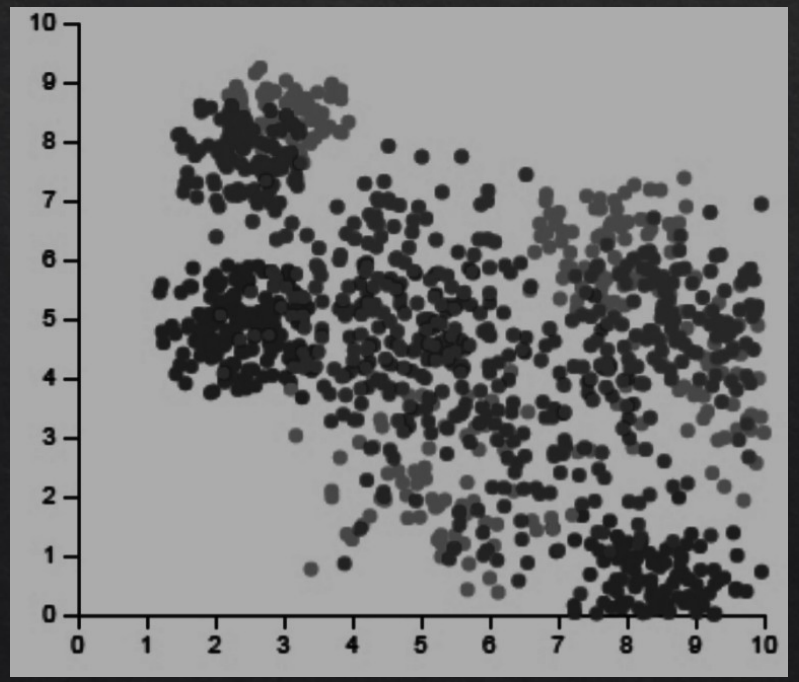
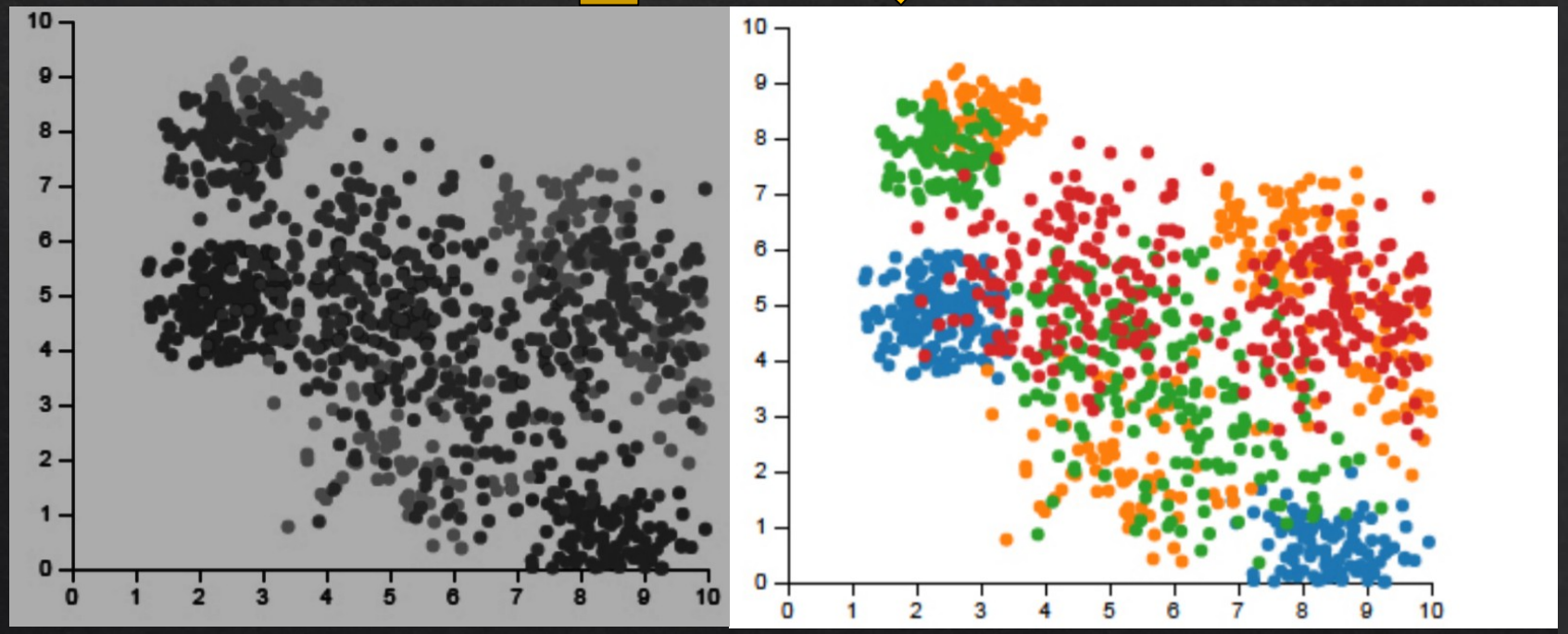
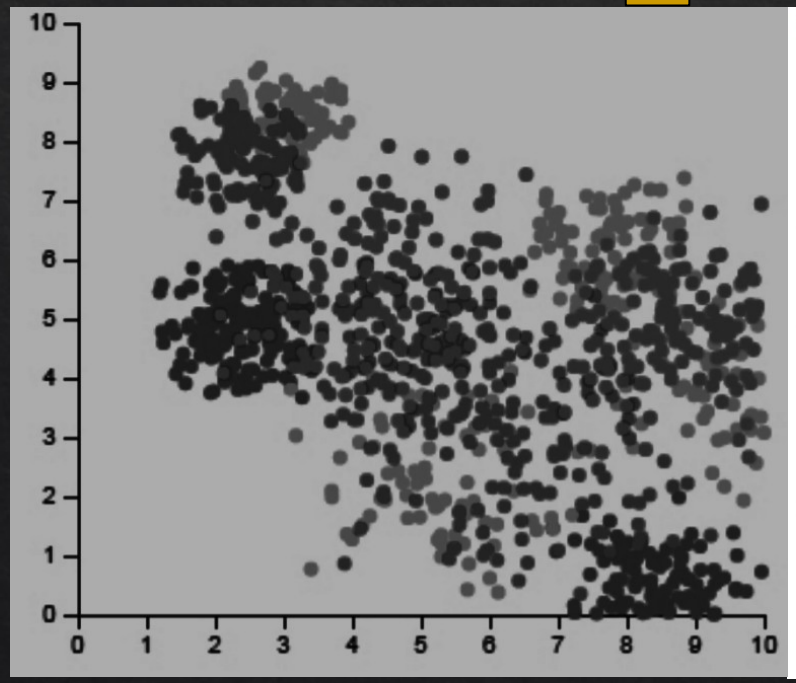
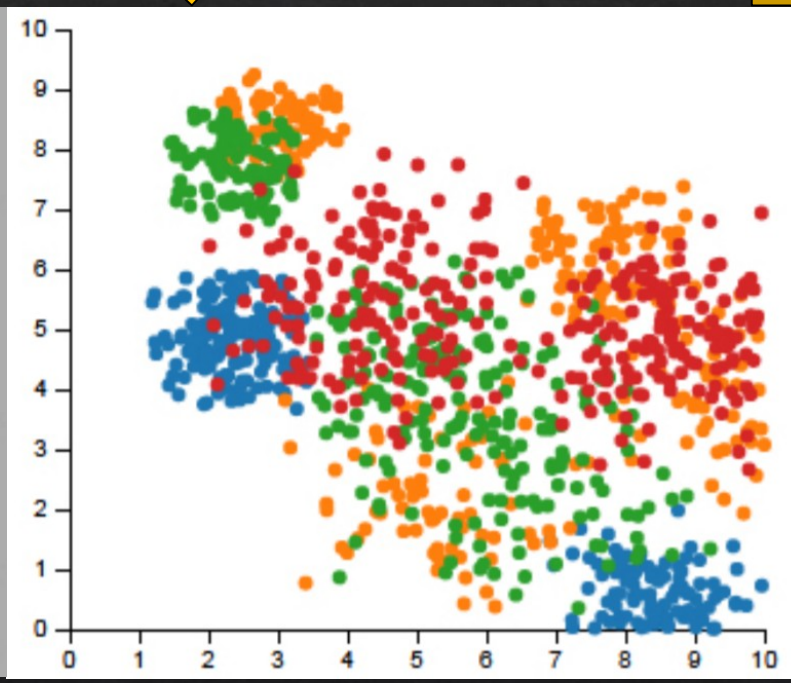
Cluster before You Color

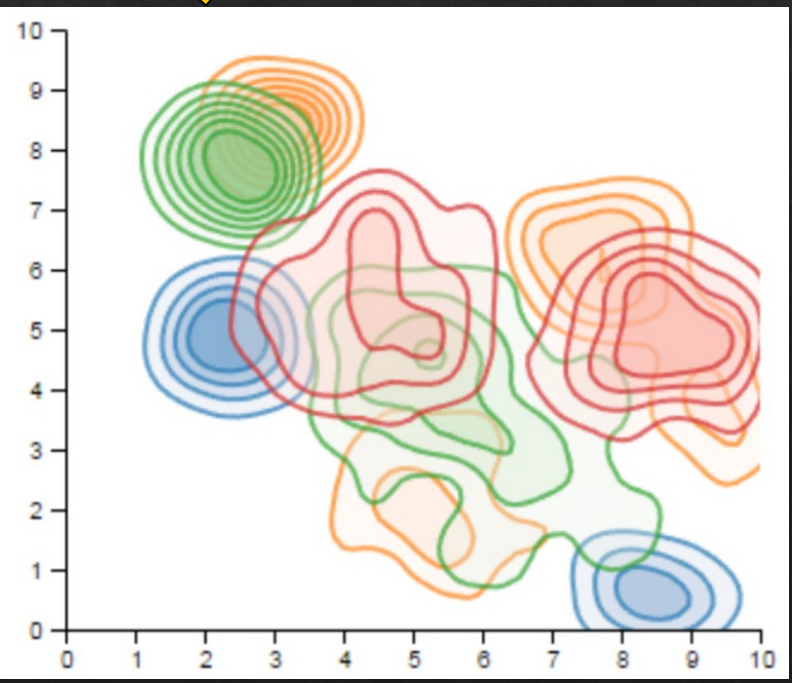
Debajyoti Mondal
University of Saskatchewan


10
9
8
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10


0 1 2 3 4 5 6 7 8 9 10







https://scikit-learn.org/stable/modules/clustering.html

https://github.com/d3/d3-contour

K-means Clustering

K-Means Clustering

Input: A point set S and an integer k
Output: A set of k points: $c_1, c_2, \ldots, c_k$, where the points of S which are closer to c_i form the ith cluster.

K-Means Clustering

Input: A point set S and an integer k
Output: A set of k points: $c_1, c_2, \ldots, c_k$, where the points of S which are closer to c_i form the ith cluster.

Why not just output a set of k random points?

K-Means Clustering

Input: A point set S and an integer k

Output: A set of k points: $_1$, $_2$, … , $_k$, where the points of S which are closer to $_i$ form the ith cluster.

Why not just output a set of k random points?

Minimize the mean distance between data points and their cluster centroid, i.e., within-cluster sum of squared distances. $\sum_{j=1}^{k} \sum_{x \in C_j}$ ¿ ¿ ¿

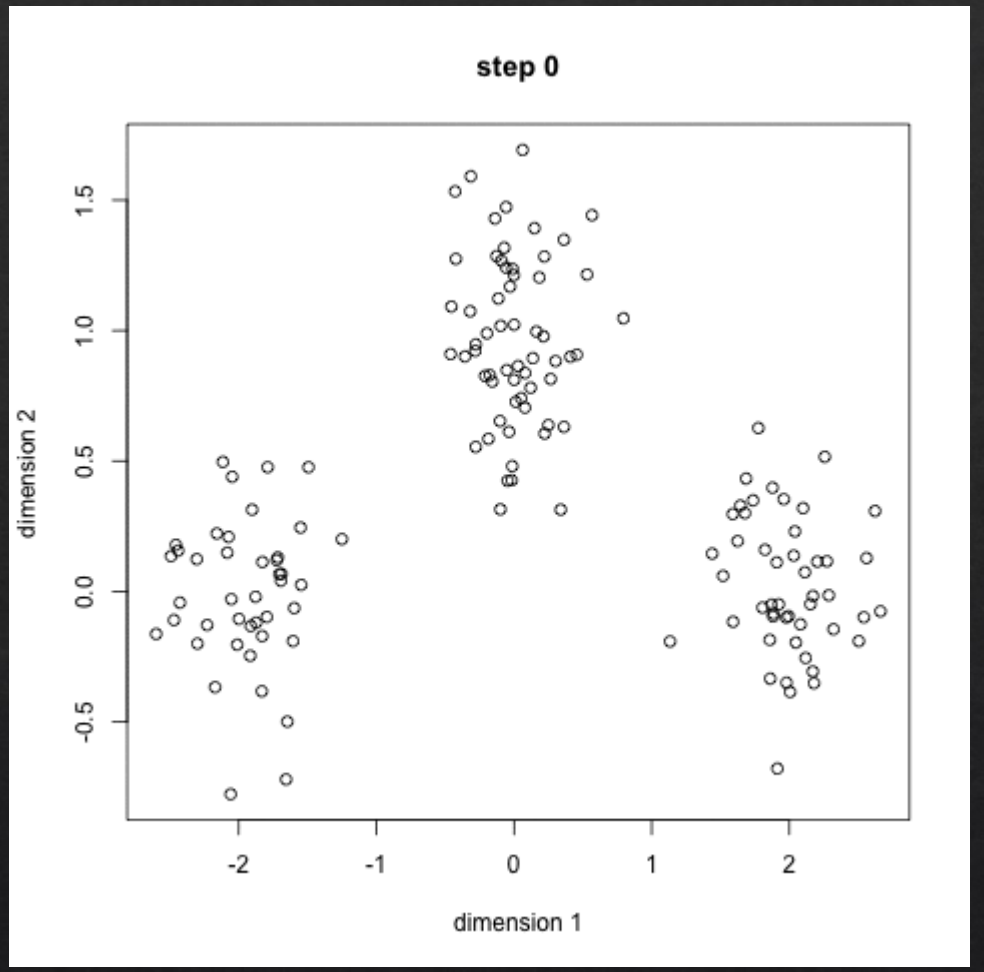
K-Means Clustering

- To begin, we first select a number of classes/groups to use and randomly initialize their respective center points.
- Classify each data point: classify the point to be in the group whose center is closest to it.
- Based on these classified points, we recompute the group center by taking the mean of all the points in the group.
- Repeat these steps for a set number of iterations or until the group centers do not change much between iterations.

K-Means Clustering

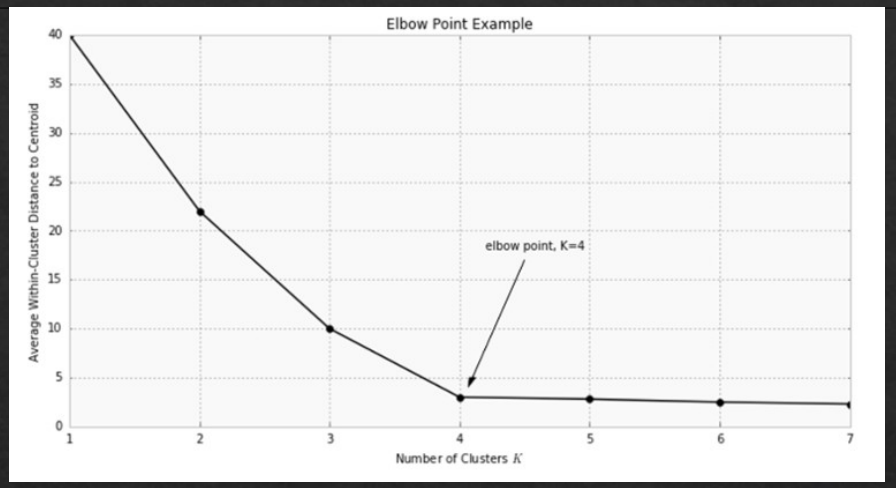
- To begin, we first select a number of classes/groups to use and randomly initialize their respective center points.
- Classify each data point: classify the point to be in the group whose center is closest to it.
- Based on these classified points, we recompute the group center by taking the mean of all the points in the group.
- Repeat these steps for a set number of iterations or until the group centers do not change much between iterations.

K-Means Clustering



Choosing K

- One common technique is to use the mean distance between data points and their cluster centroid.



- Increasing K decreases this mean distance metric

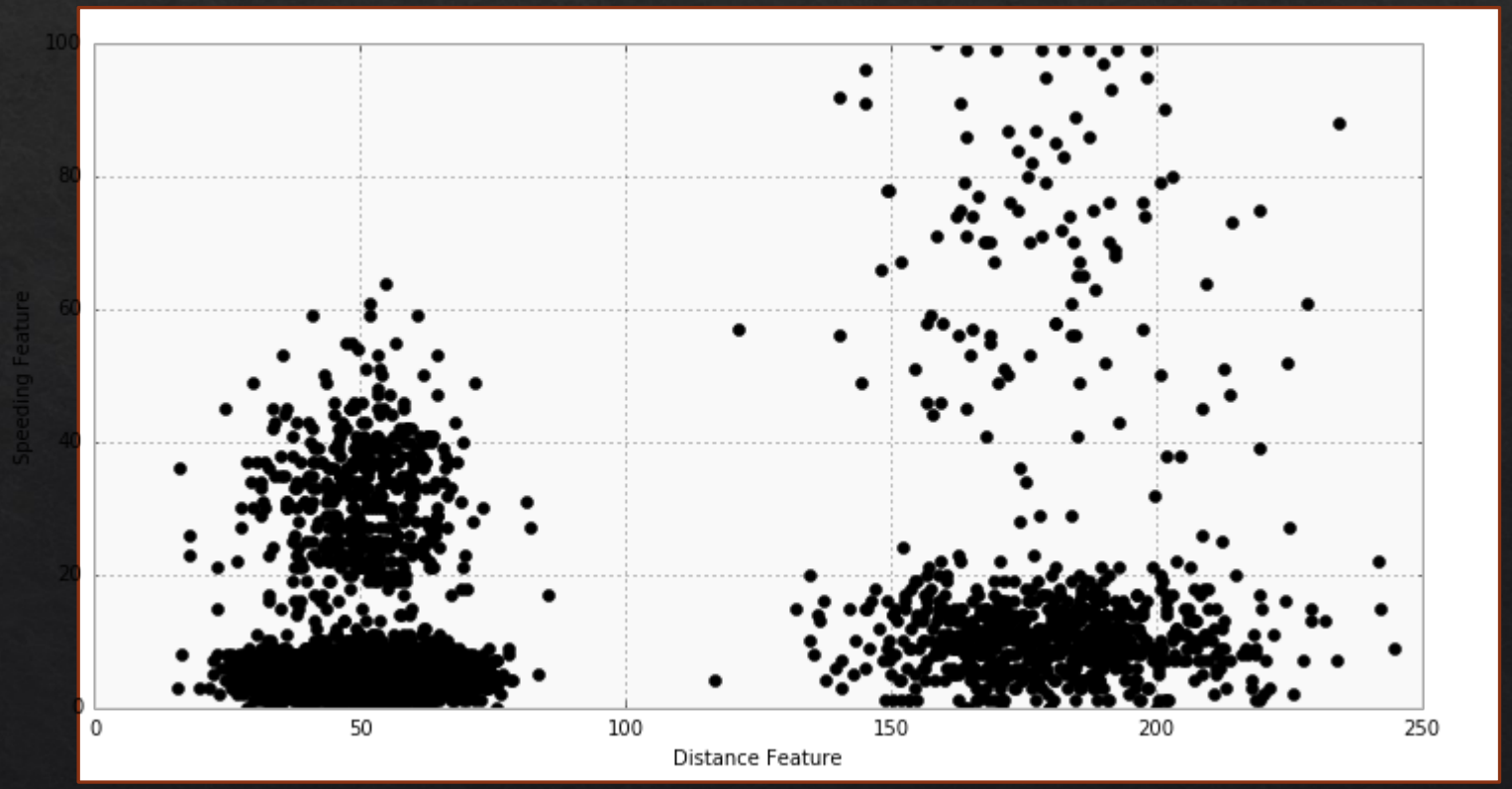
- We plot mean distance as a function of K and the elbow point, where the rate of decrease sharply shifts, can be used to roughly determine K.

Usages

- Document Classification
- Delivery Store Optimization
- Identifying Crime Localities
- Customer Segmentation
- Automatic Clustering of IT Alerts
- Insurance Fraud Detection
- …

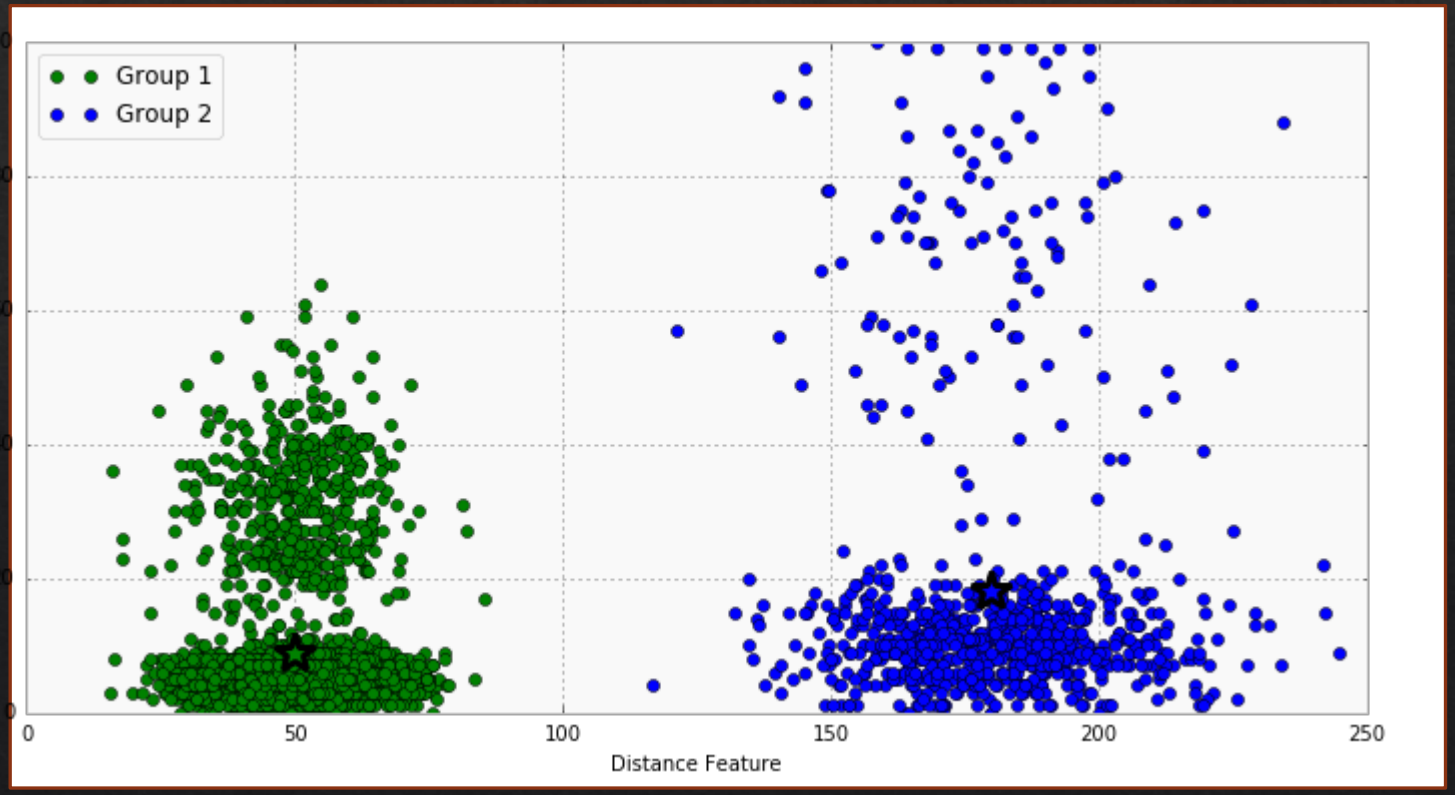
K-means Example

The chart below shows the dataset for 4,000 drivers, with the distance feature on the x-axis and speeding feature on the y-axis.



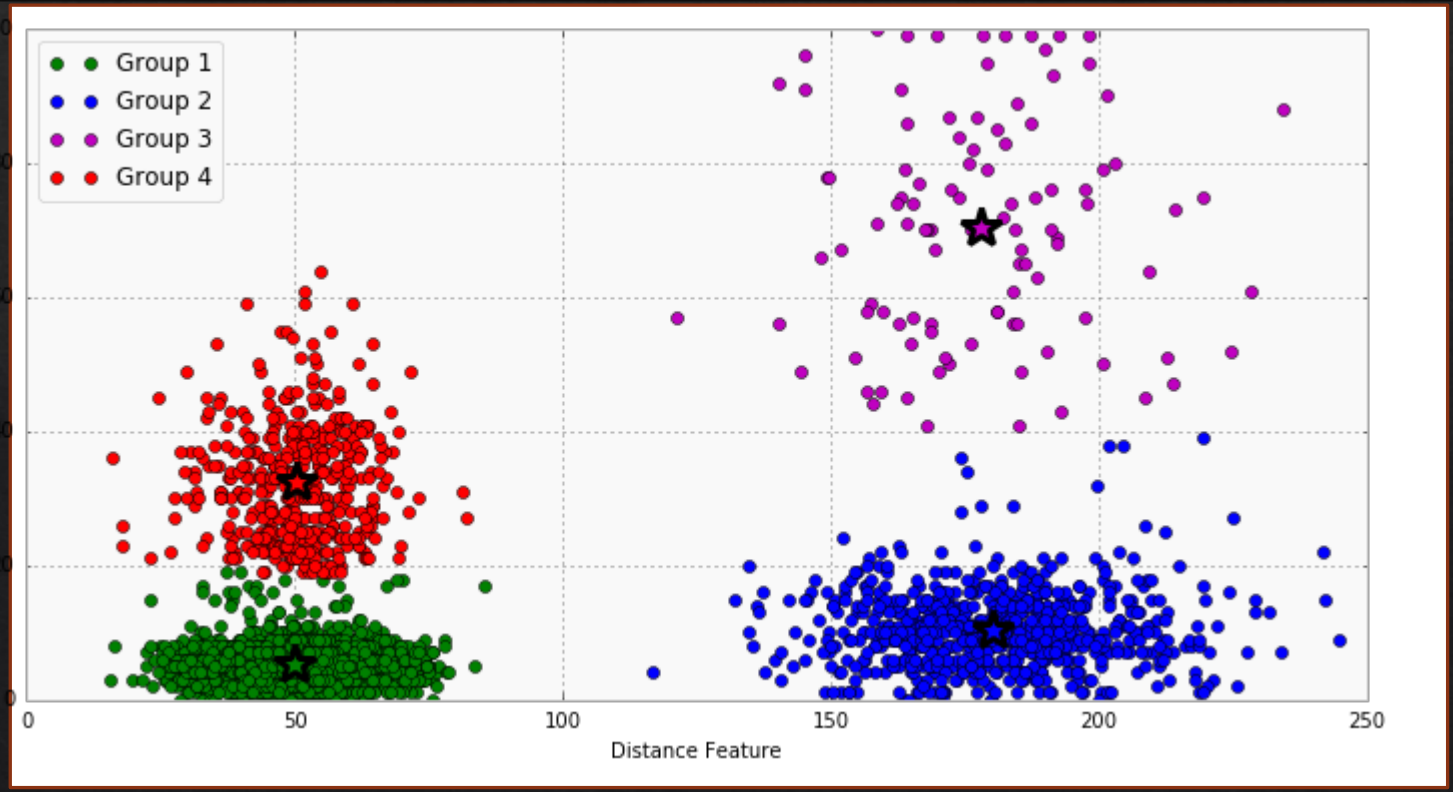
K-means when K=2

Using domain knowledge of the dataset, we can infer that Group 1 is urban drivers and Group 2 is rural drivers.



K-means when K=4

Speeding drivers have been separated from those who follow speed limits, in addition to the rural vs. urban divide.
The threshold for speeding is lower with the urban driver group than for the rural drivers, likely due to urban drivers spending more time in intersections and stop-and-go traffic.

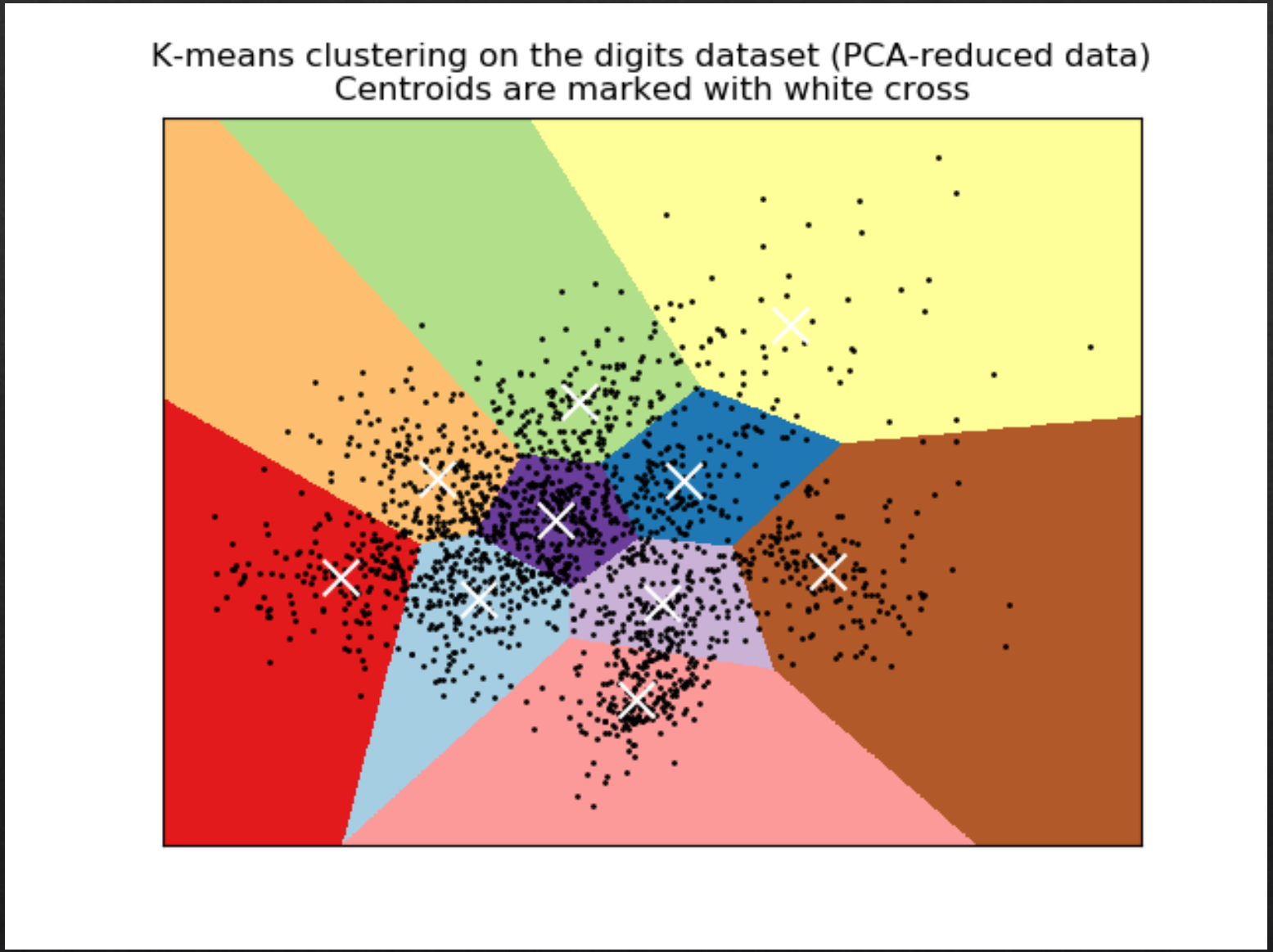


Feature Engineering

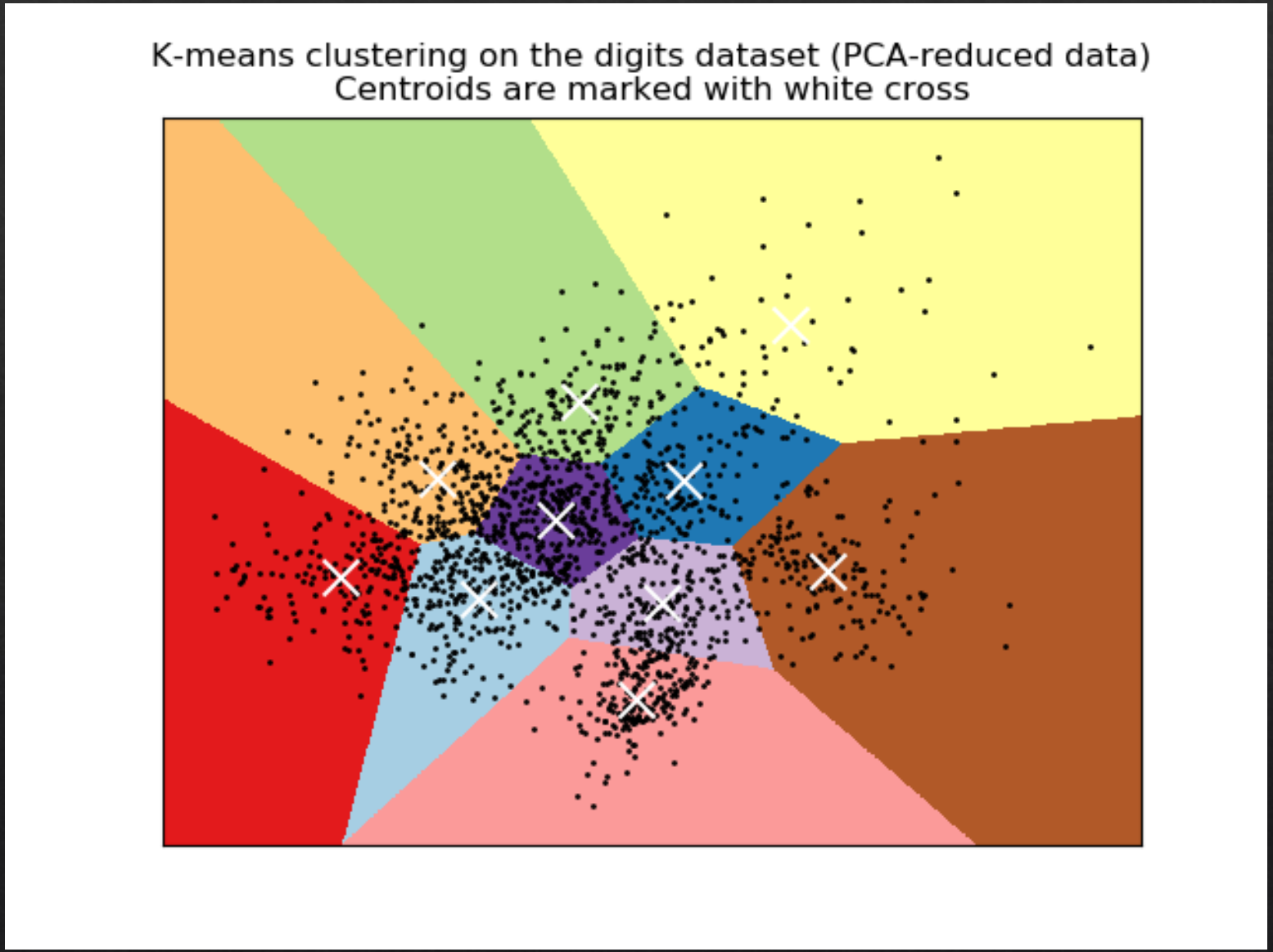
Additional Information: not within the scope of the course

- K-means clustering is a form of unsupervised machine learning
- Feature engineering is the process of using domain knowledge to choose which data metrics to input as features into a machine learning algorithm.
- Using meaningful features that capture the variability of the data is essential for the algorithm to find the naturally-occurring groups.
- Categorical data needs to be encoded.

K-Means clustering on the handwritten digits data



The background is a **VORONOI DIAGRAM** and the cells are called **VORONOI CELLS**



Does K-means try to optimize any error?

- The K-means algorithm aims to minimize the within-cluster sum-of-squares:

$$\sum_{j=1}^{k} \sum_{x \in C_j} ¿¿¿$$

- Running time: O(I * K * n * d)
  - n : number of points
  - K : number of clusters
  - I : number of iterations
  - d : number of attributes

Does K-means try to optimize any error?

- The K-means algorithm aims to minimize the within-cluster sum-of-squares:

$$\sum_{j=1}^{k} \sum_{x \in C_j} ¿¿¿$$

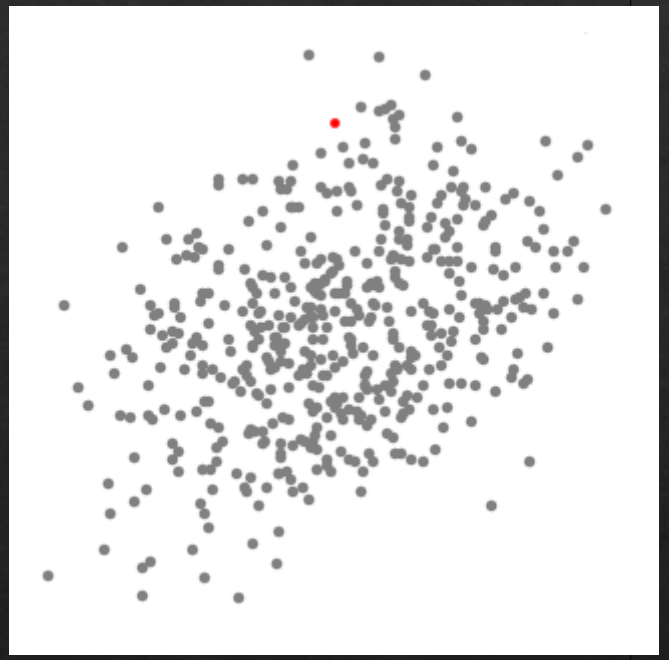
- Running time: O(*IKnd*), or linear if I,K,d are constant.
  - n : number of points
  - K : number of clusters
  - I : number of iterations
  - d : number of attributes

If *d* is large, apply PCA before K-means

Mean-Shift Clustering

Mean-Shift Clustering

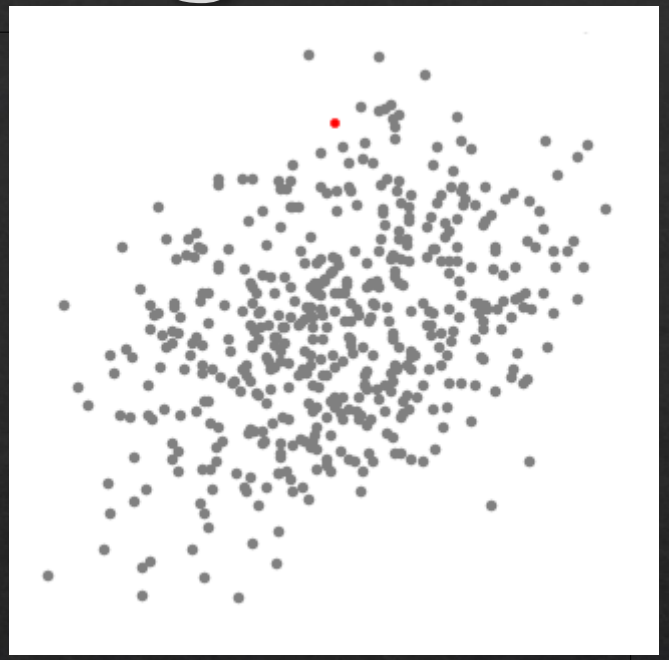
- First consider a set of points in two-dimensional space.
- Start with a circular sliding window centered at a point C (randomly selected) and having radius r determines the kernel.
- Mean-shift shifts this kernel iteratively to a higher density region on each step until convergence.



Mean-Shift Clustering

- How to Shift?

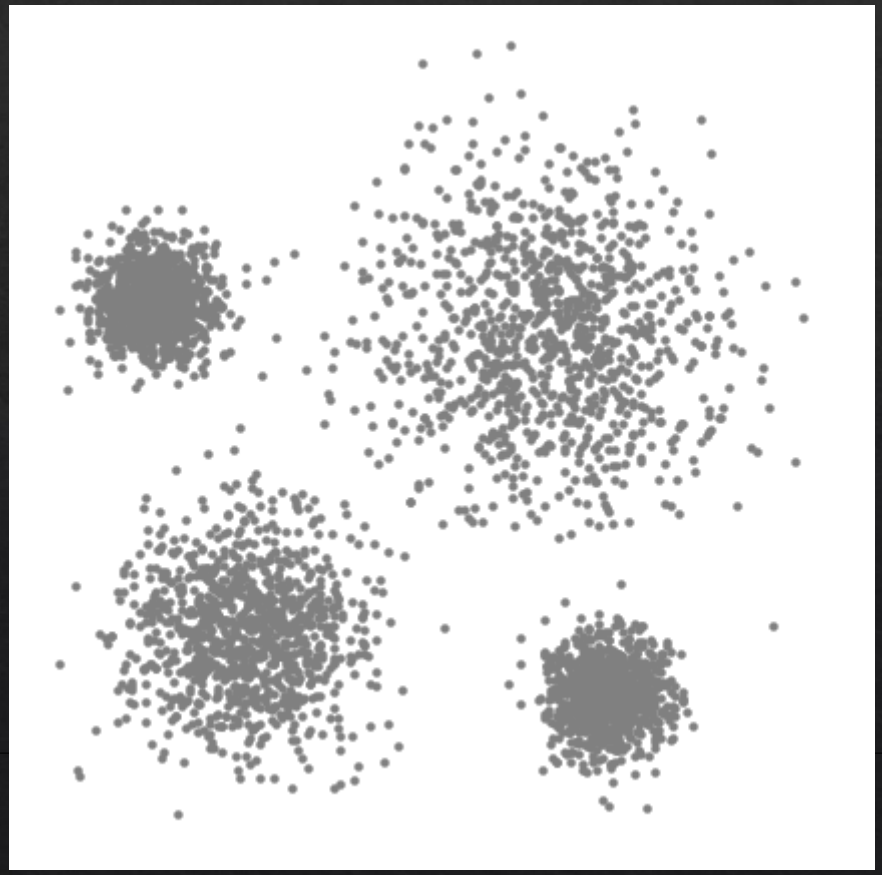
- shift the center point to the mean
  of the points within the window/circle



By shifting to the mean of the points in the window it will gradually move towards areas of higher point density!

Mean-Shift Clustering

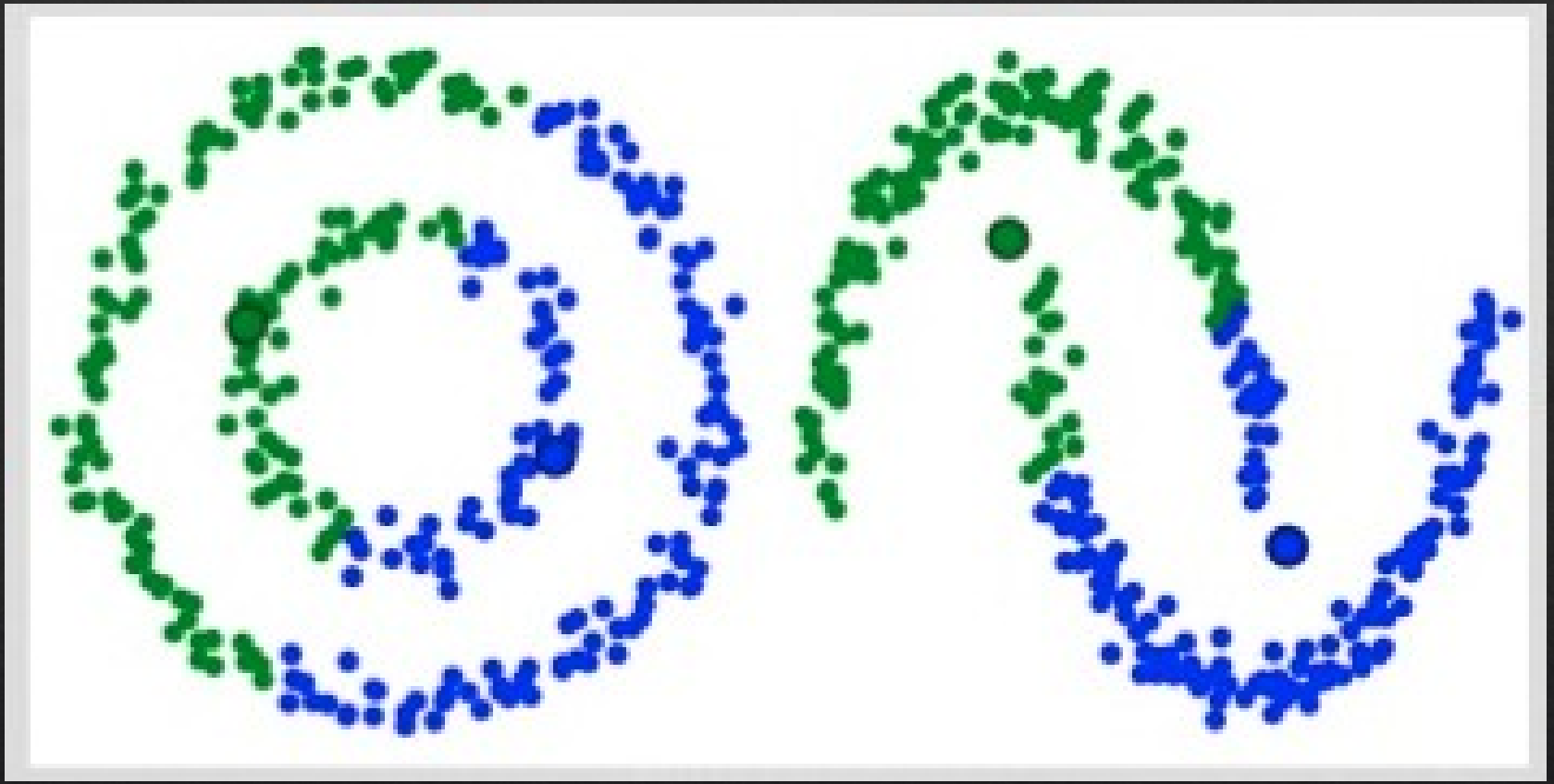
- This process is done with many sliding windows until all points lie within a window.
- When multiple sliding windows overlap the window containing the most points is preserved.
- The data points are then clustered according to the sliding window in which they reside.



Mean-Shift Clustering

- No need to guess the number of cluster centers!
- Slower than K-means --- Why?
- Selection of window size is difficult

Failure Cases



Class Activity:

How to design a clustering algorithm to separate the two

O-shapes or U- shapes in the picture?

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

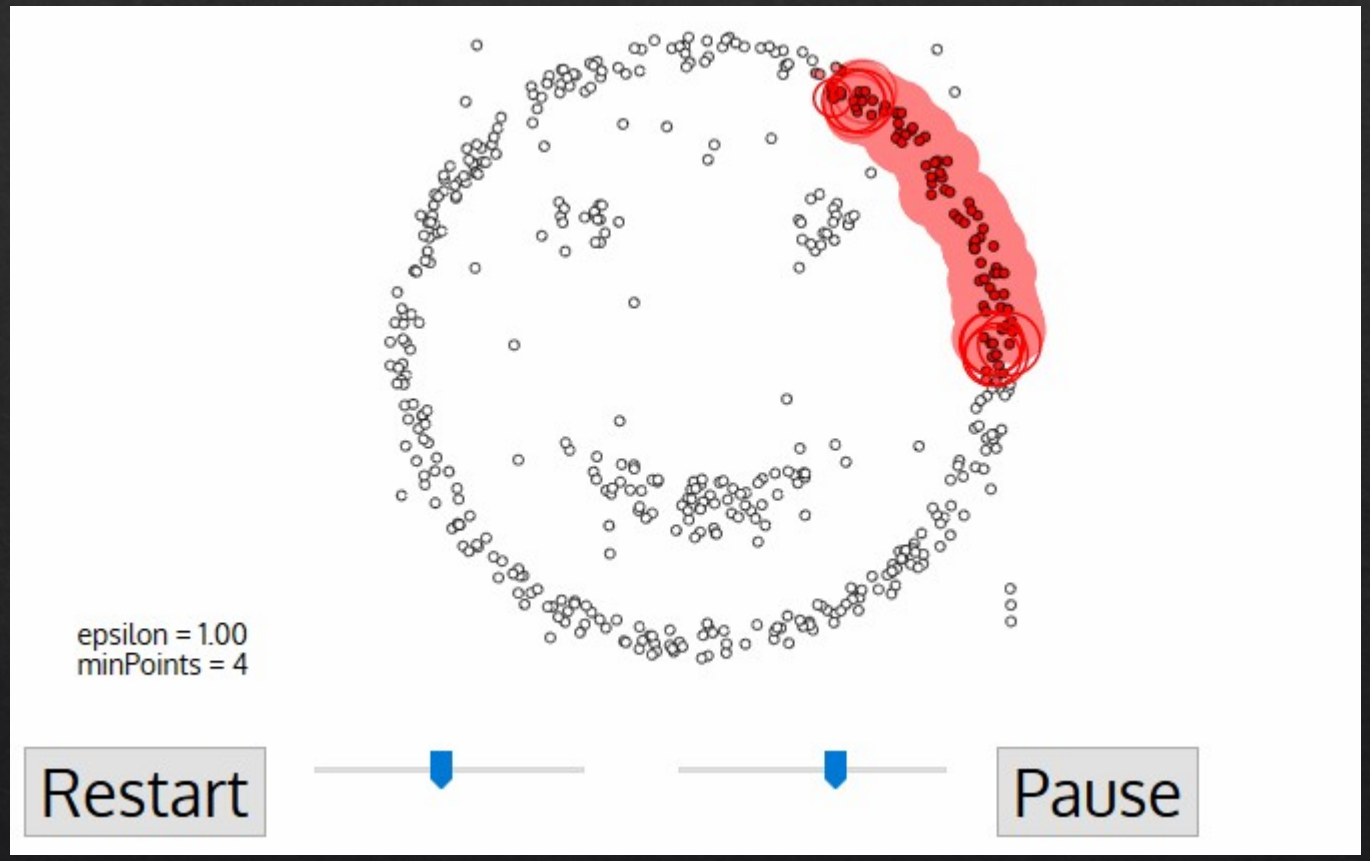
DBSCAN

- Start with an arbitrary starting data point X that has not been visited. The neighborhood of this point is extracted using a distance epsilon ε.
- If the number of points in the neighborhood > minPoints, then the clustering process starts and the current data point becomes the first point in the new cluster. Otherwise, this datapoint is just a noise.
- The points in the neighborhood of X is assigned the cluster of X and the process is repeated with these newly added points.

DBSCAN

- Once we're done with the current cluster, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.
- This process repeats until all points are marked as visited.
- At the end of this, all points have been visited, each point well have been marked as either belonging to a cluster or being noise.

DBSCAN



https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

https://www.naftaliharris.com/blog/visualizing-k-means-cluste

DBSCAN

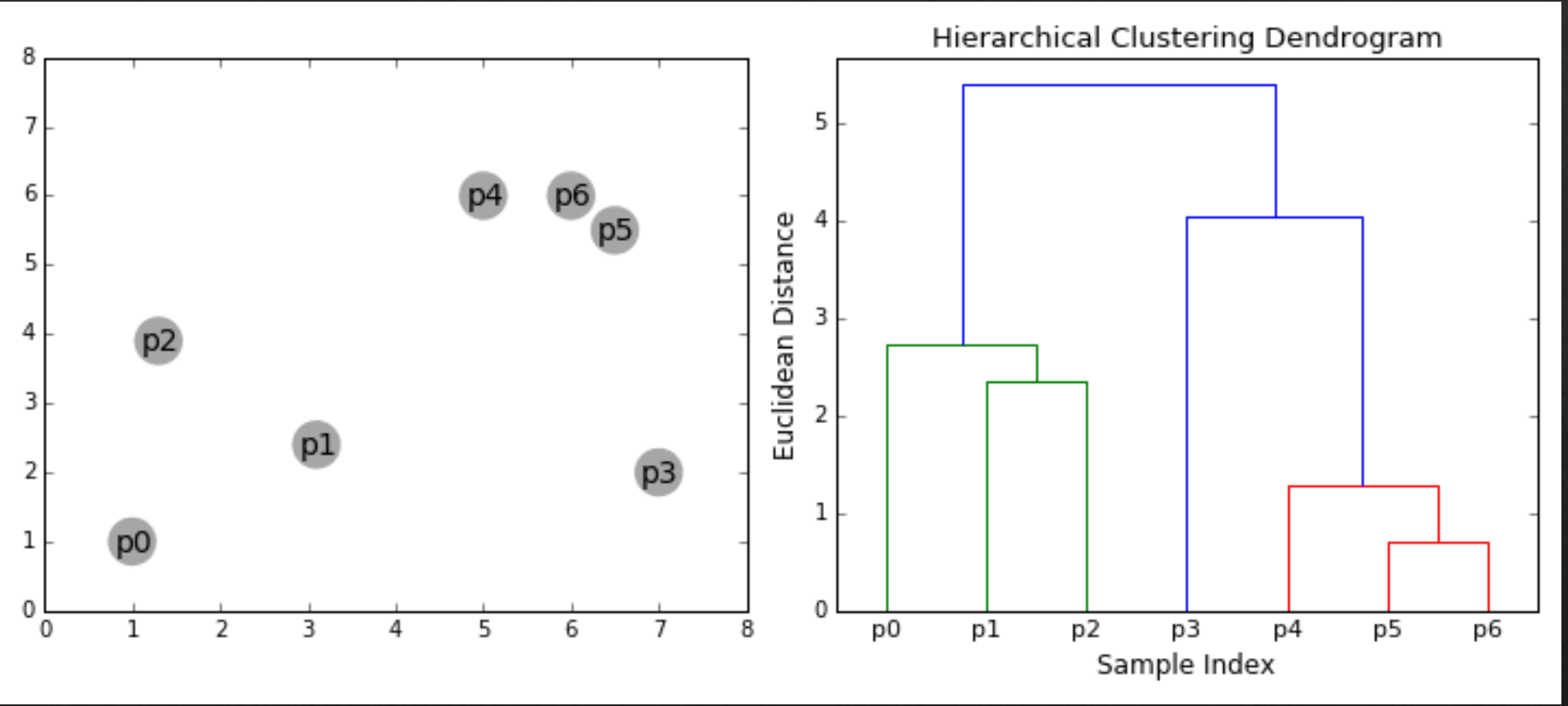
- Does not require a pre-set number of clusters
- Identifies outliers as noises unlike mean-shift which simply throws them into a cluster
- Able to find arbitrarily sized and arbitrarily shaped clusters
- Depending on parameters, maybe slower than K-means [requires query data structure for fast implementation]
- Doesn't perform as well as others when the clusters are of varying density. How to choose epsilon ε and minPoint?

Agglomerative Hierarchical Clustering

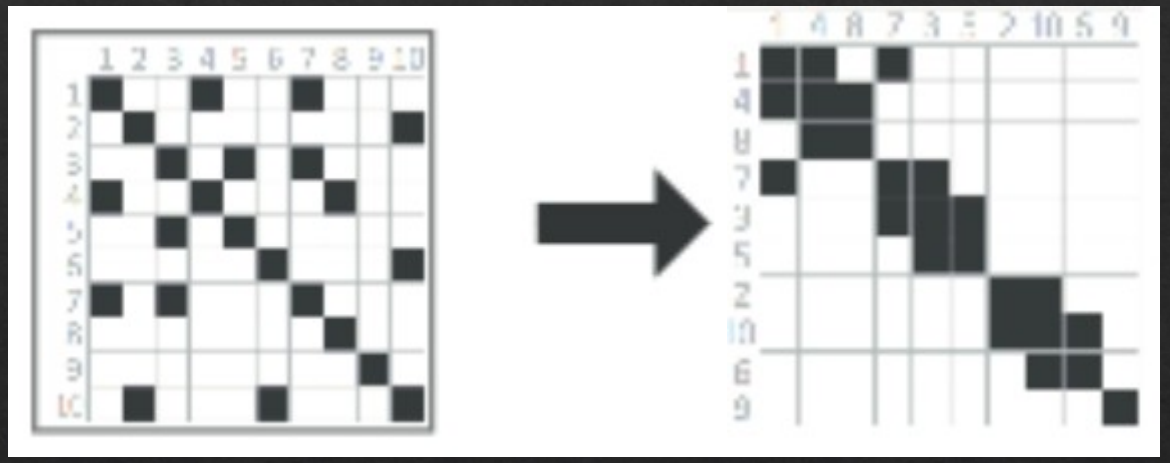
Hierarchical Clustering

- Start by treating each data point as a single cluster
- Choose a distance metric that measures the distance between two clusters
- On each iteration we combine two clusters into one. The two clusters to be combined are selected as those with the smallest distance
- Repeat until we have the expected number of clusters

Hierarchical Clustering Example

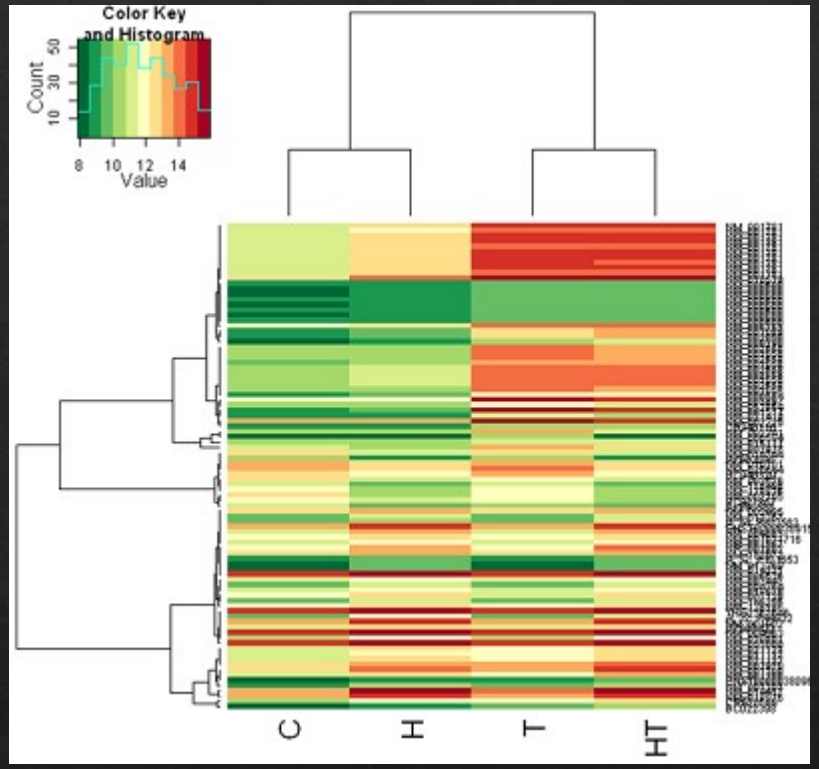


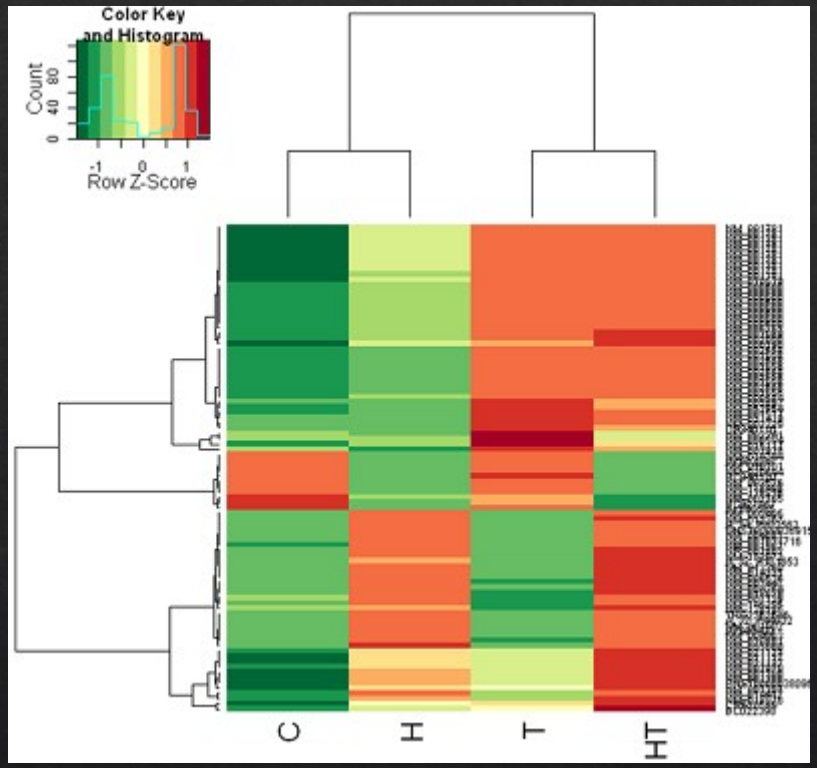
Good Clustering Makes a Difference



We can reorder the rows and columns to reveal cluster information

Good Clustering Makes a Difference



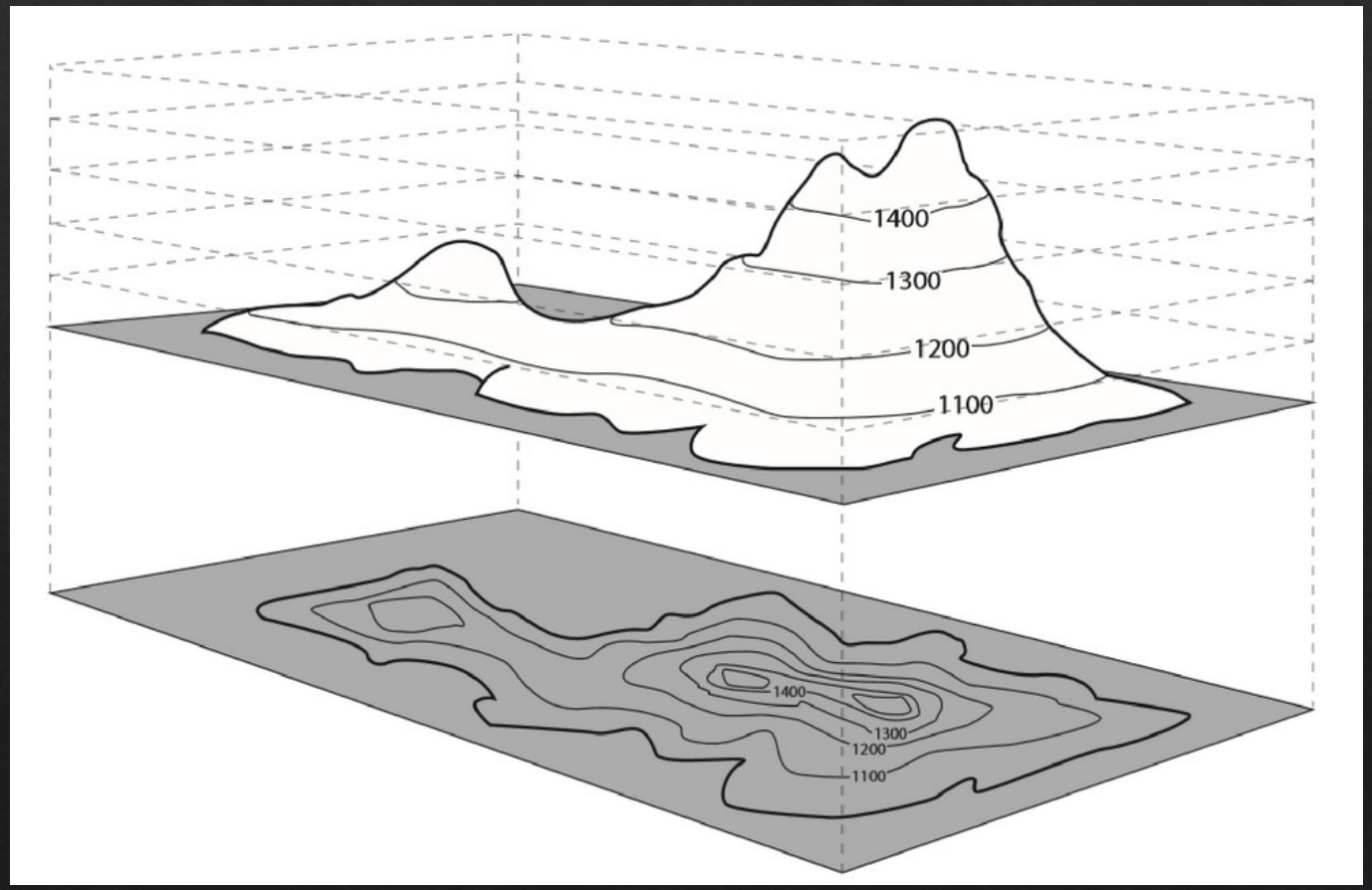


If the cell adjacencies are not important, then clustering on a heatmap may reveal interesting cluster information

Hierarchical Clustering

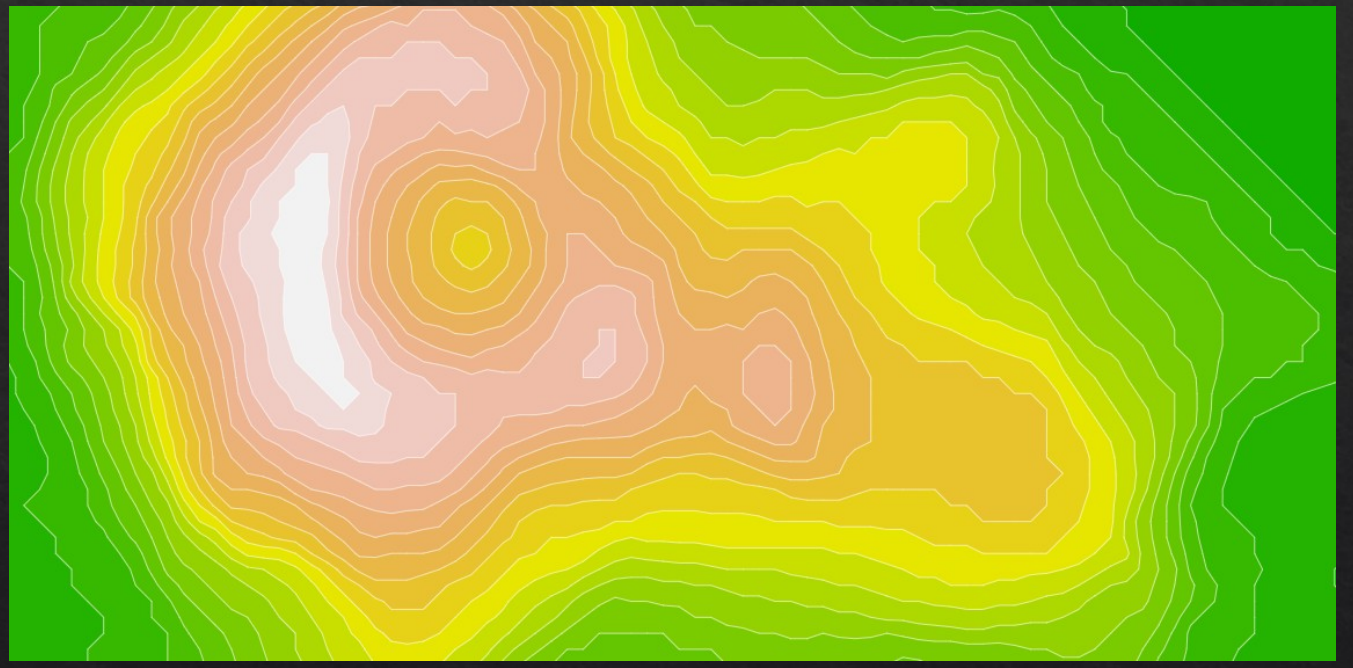
- Does not require a pe-set number of clusters
- Can recover hierarchical structure!
- Dendograms are great for visualization
- Slower – all pair distance computation [requires query data structure for fast implementation]

Contouring



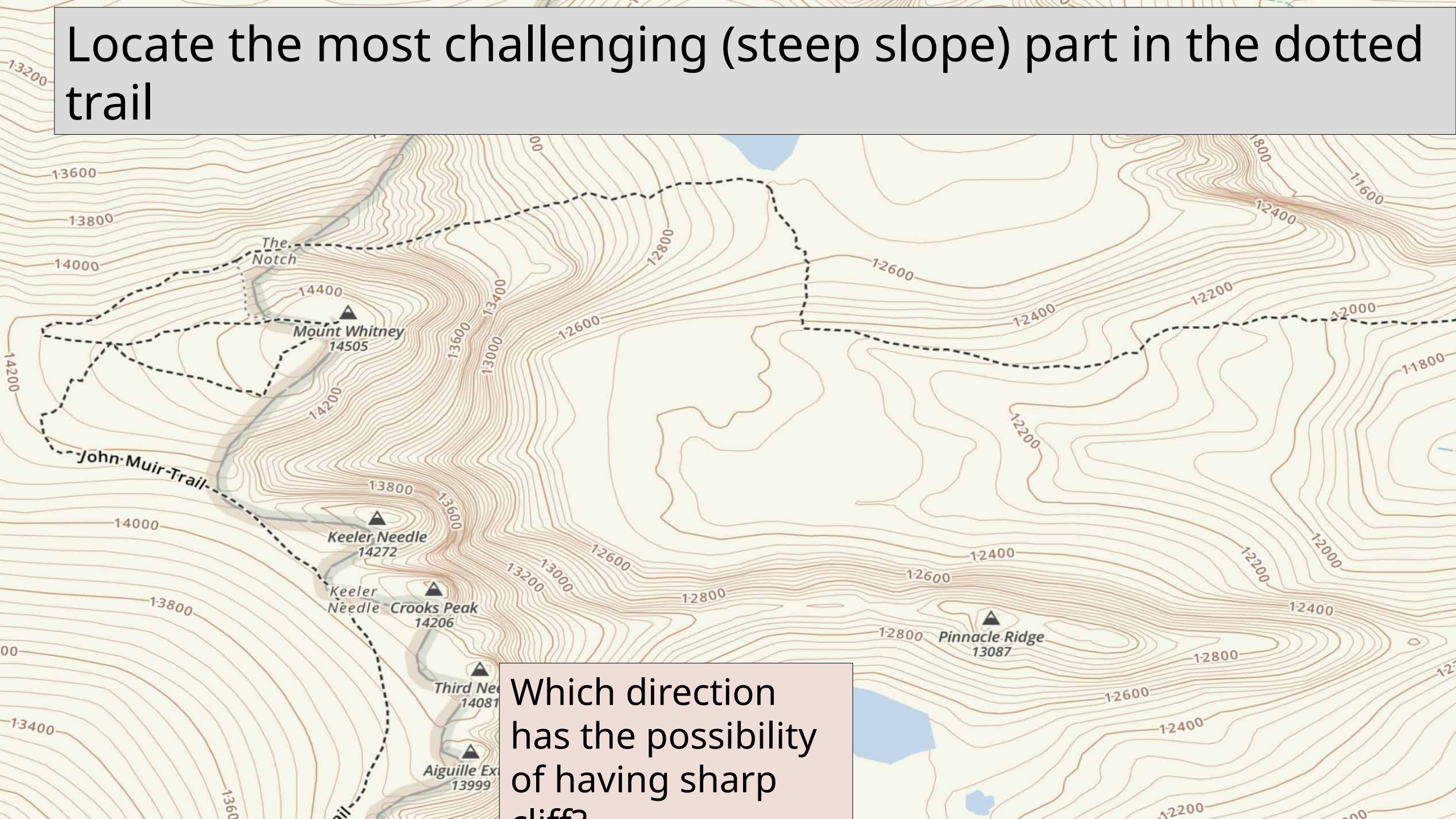
Contouring

Isoline – a line following a single data level, or isovalue

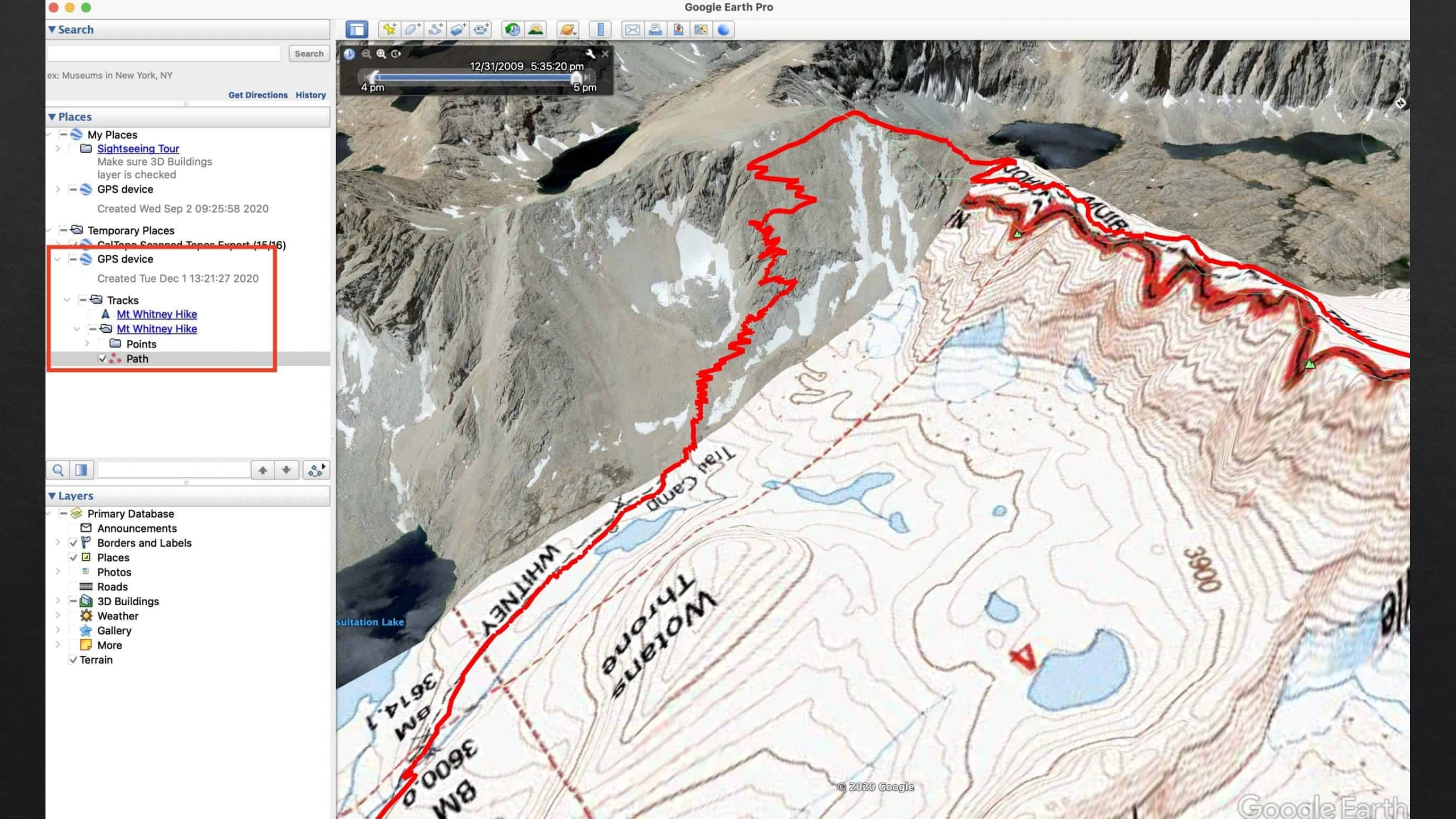


https://bl.ocks.org/mbostock/4241134

Locate the most challenging (steep slope) part in the dotted trail

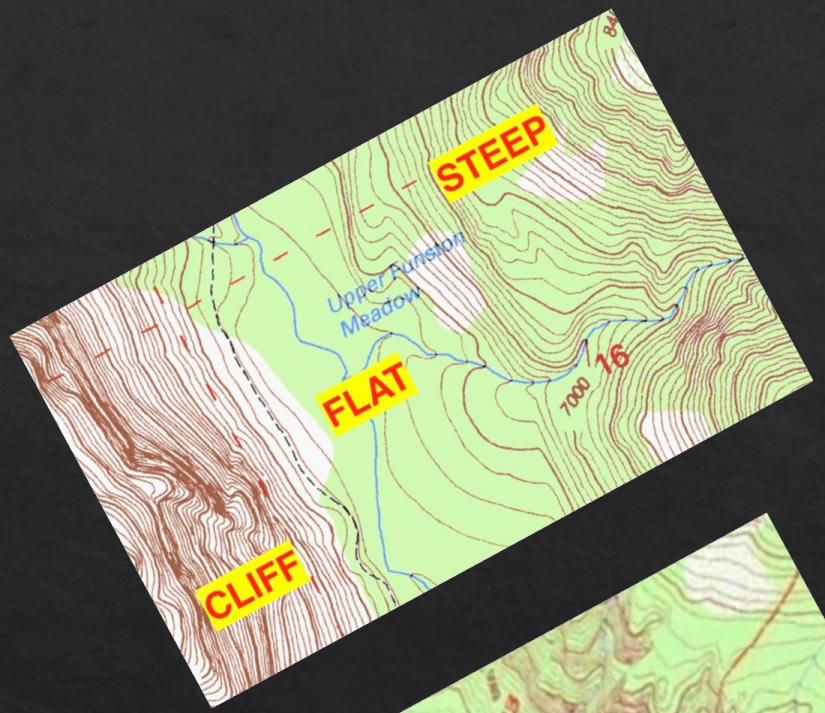


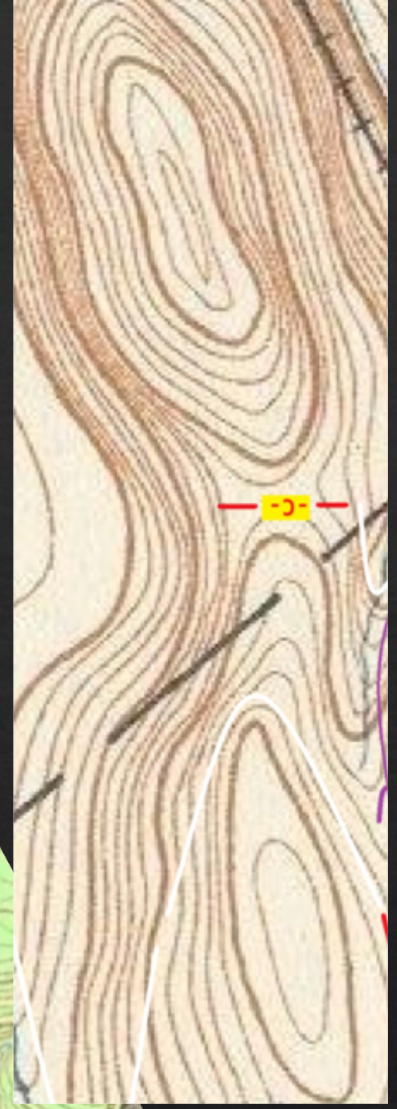

Mount
Whitney
nt in
U.S.
14494'
Iceberg
Lake
MOUNTAINEER'S ROUTE
12500
11500
13500
Keeler
Needle
Pinnacle
Ridge
12250
ZONE


Google Earth Pro
Search
Search
ex: Museums in New York, NY
Get Directions
History
Places
My Places
Sightseeing Tour
Make sure 3D Buildings
layer is checked
GPS device
Created Wed Sep 2 09:25:58 2020
Temporary Places
GPS device
Created Tue Dec 1 13:21:27 2020
Tracks
Mt Whitney Hike
Mt Whitney Hike
Points
Path
Layers
Primary Database
Announcements
Borders and Labels
Places
Photos
Roads
3D Buildings
Weather
Gallery
More
Terrain
12/31/2009 5:35:20 pm
4 pm
5 pm
Trail Camp
WHITNEY
Wotans Throne
3900
© 2020 Google
Google Earth

Class Activity:

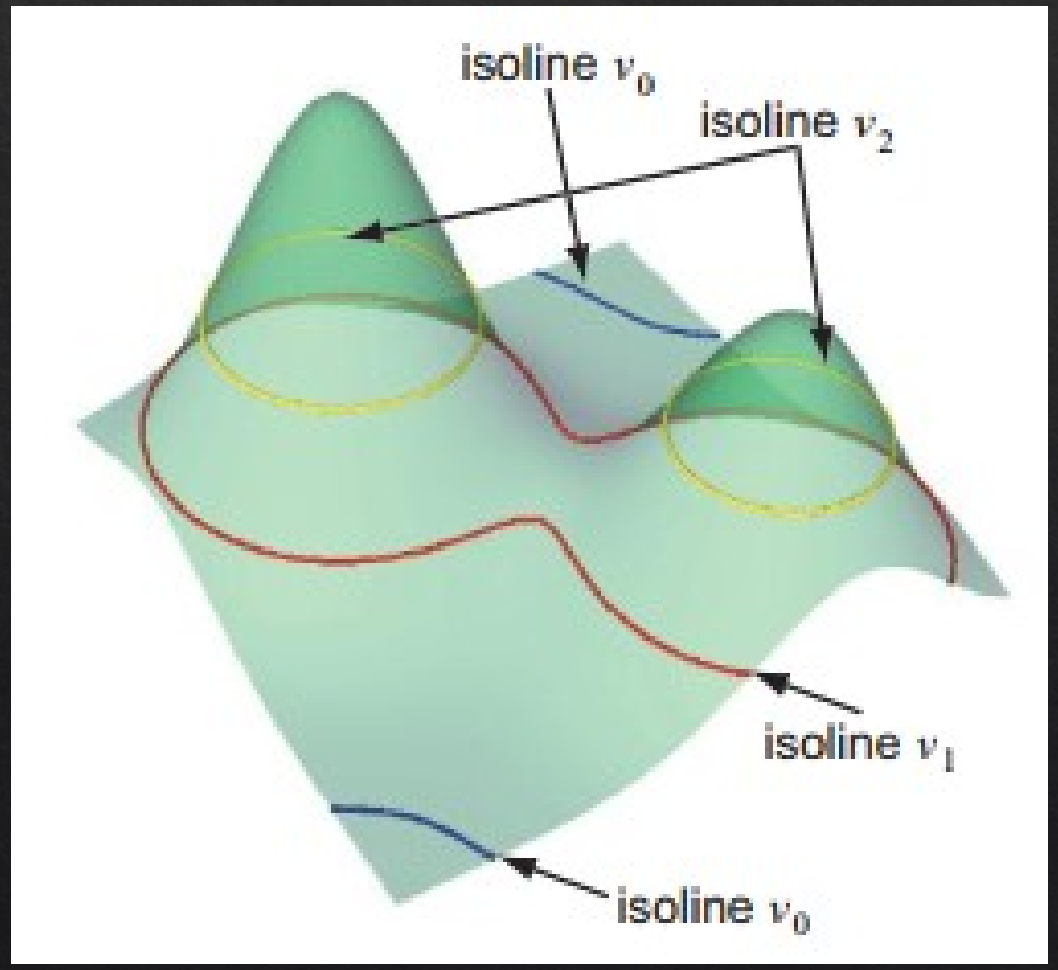
Draw a hypothetical 2D contour plot for the given setting

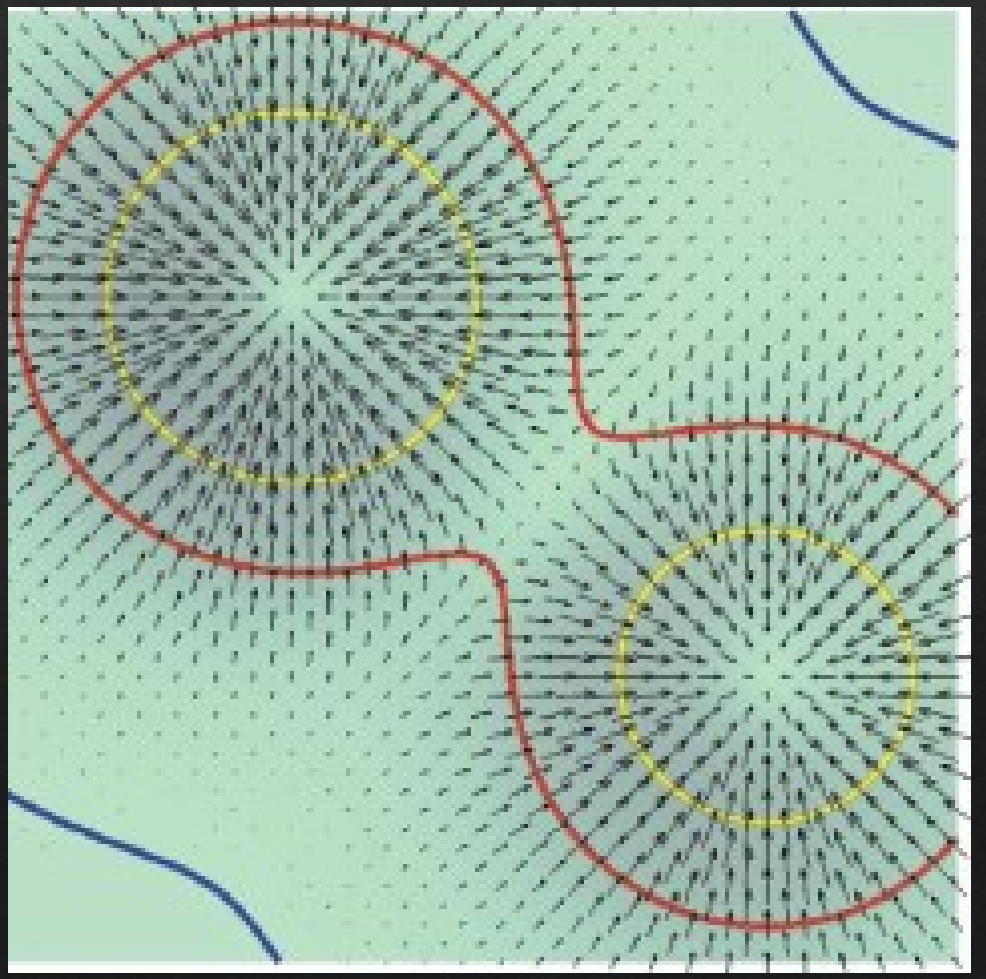

STEEP
Upper Funston Meadow
FLAT
CLIFF
7000
16


-C-


Canyon
River
6800
Mile 140

Contouring

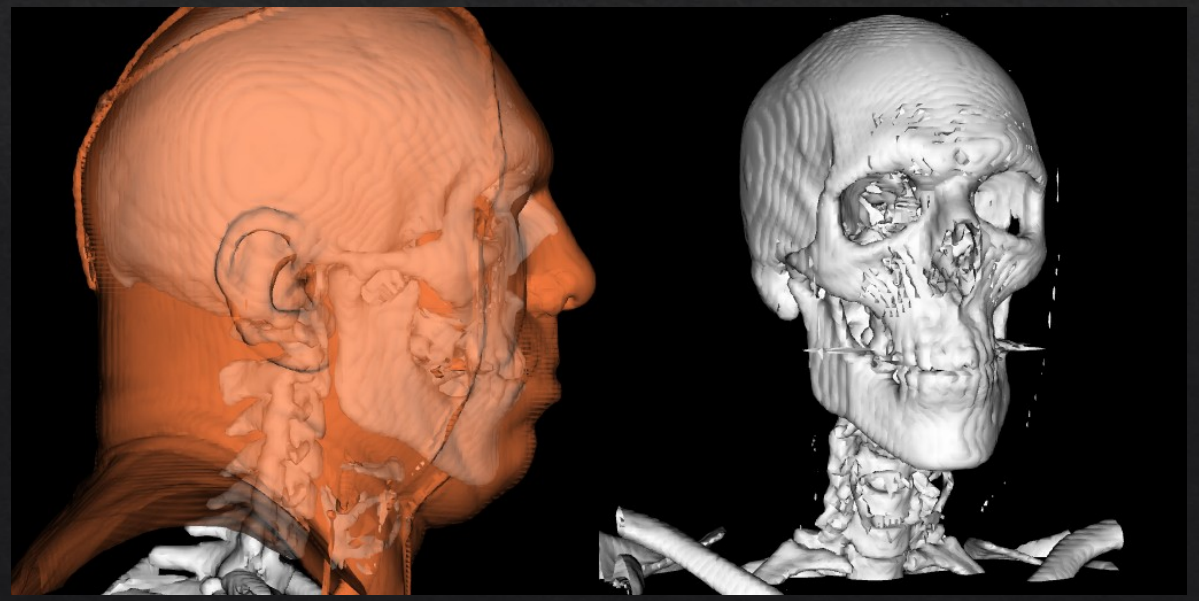


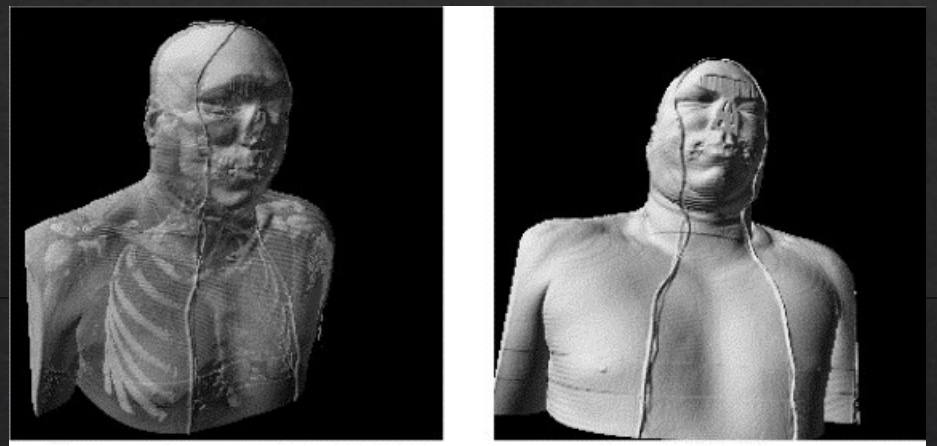


Contouring

Isoline – a line following a single data level in 2D, or isovalue

Iso-surface – a surface following a single data level in 3D





(a) two isosurfaces (skin+bone)

(b) one isosurface (skin)

(c) one isosurface (bone)

Marching Square

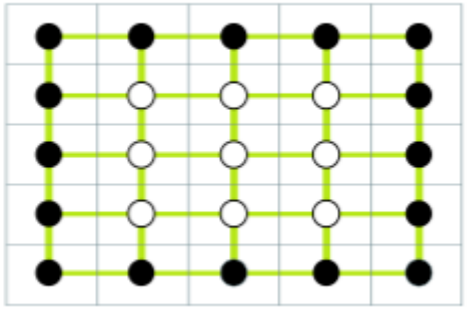
1	1	1	1	1
1	2	3	2	1
1	3	3	3	1
1	2	3	2	1
1	1	1	1	1

Threshold with iso-value

Threshold 1.5

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Binary image to cells



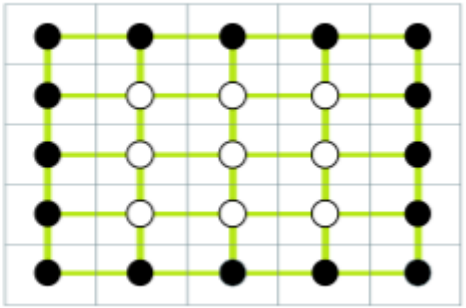
https://en.wikipedia.org/wiki/Marching_squares

1	1	1	1	1
1	2	3	2	1
1	3	3	3	1
1	2	3	2	1
1	1	1	1	1

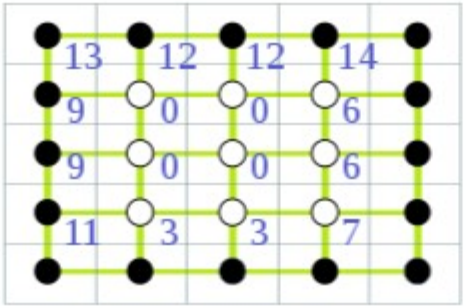
Threshold with iso-value

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

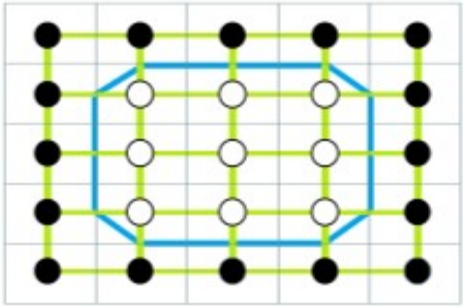
Binary image to cells



Give every cell a number based on which corners are true/false



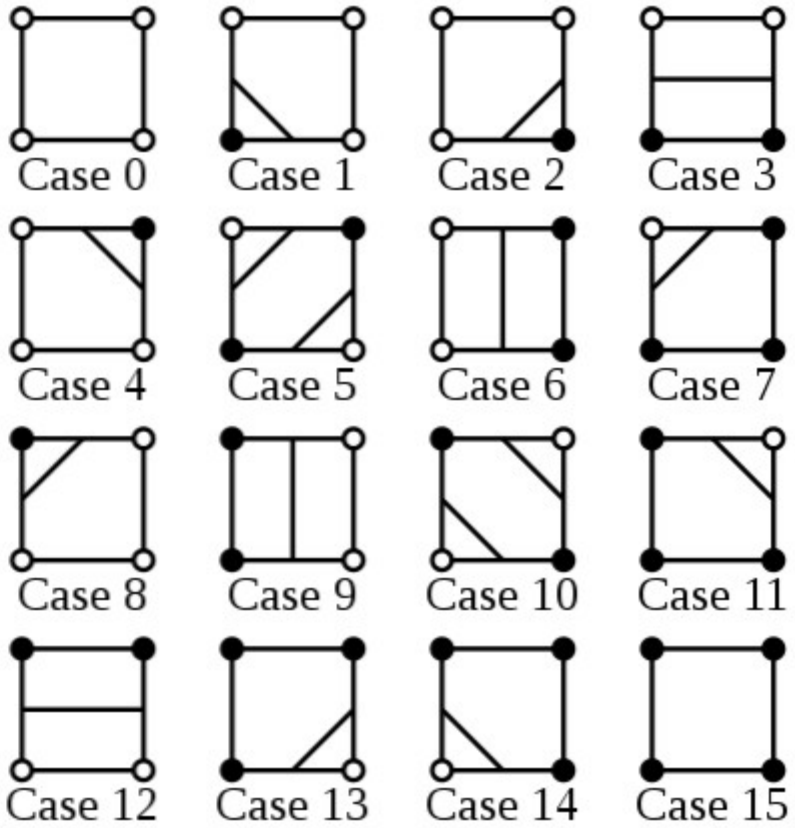
Look up the contour lines in the database and put them in the cells



Look at the original values and use linear interpolation to determine a more accurate position of all the line end-points

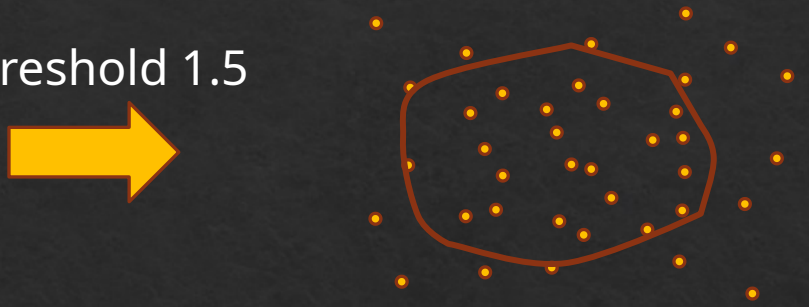
1	1	1	1	1
1	2	3	2	1
1	3	3	3	1
1	2	3	2	1
1	1	1	1	1

Look-up table contour lines



Marching Square

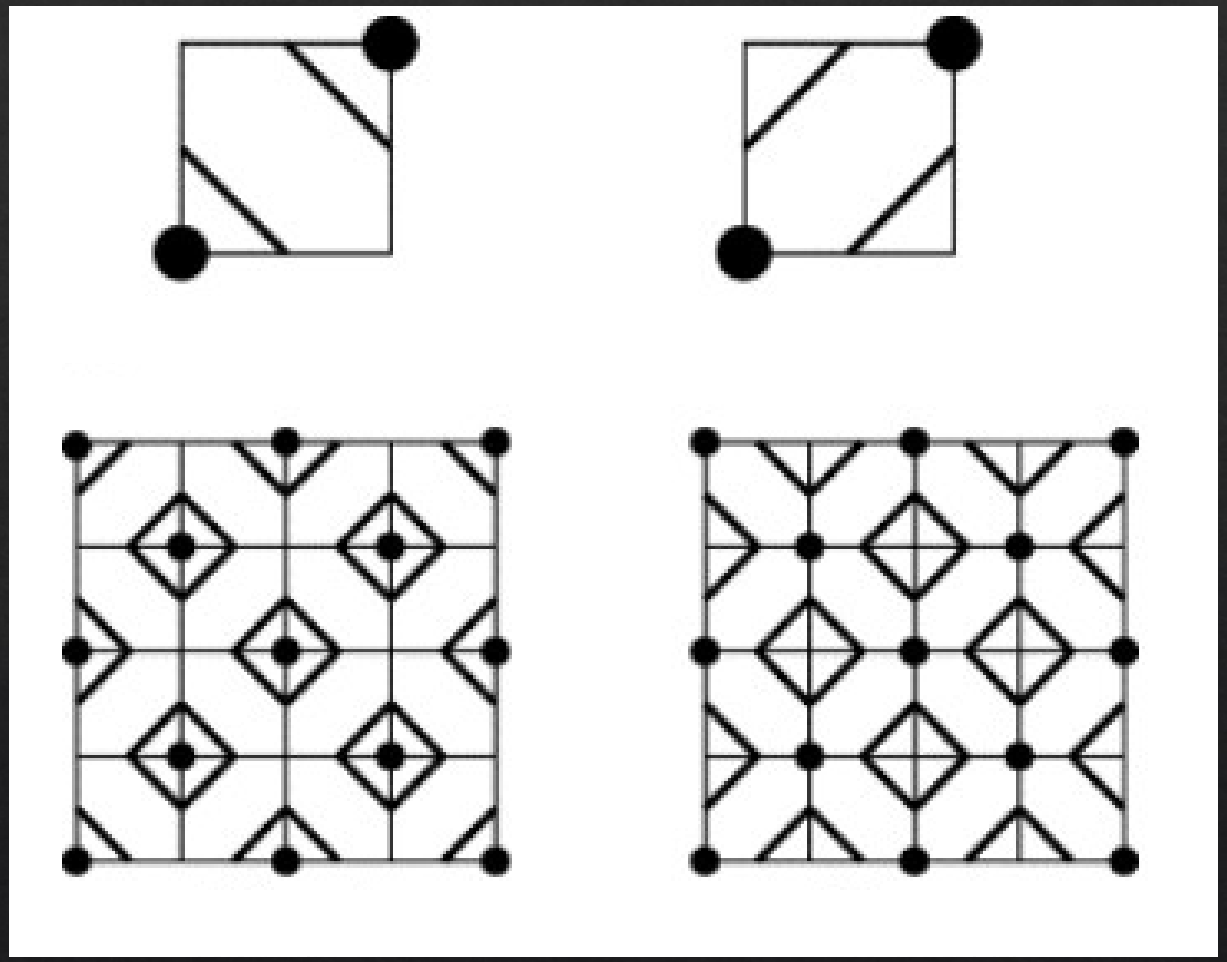
Threshold 1.5



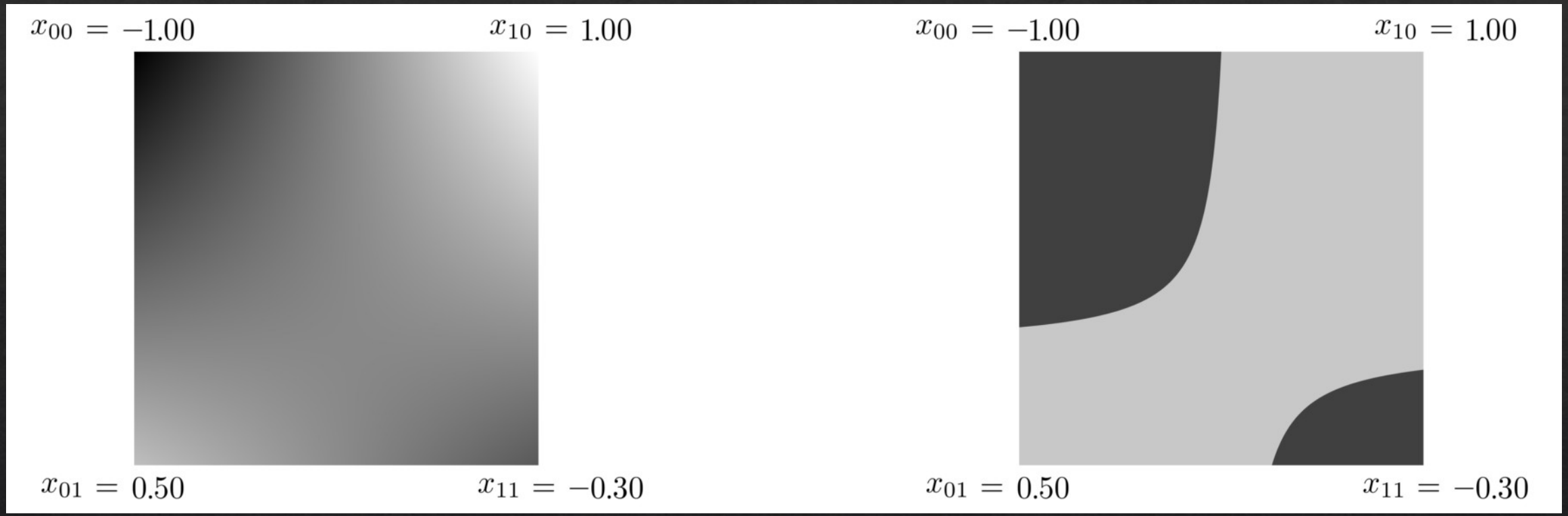
Threshold 2.5

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

Ambiguity

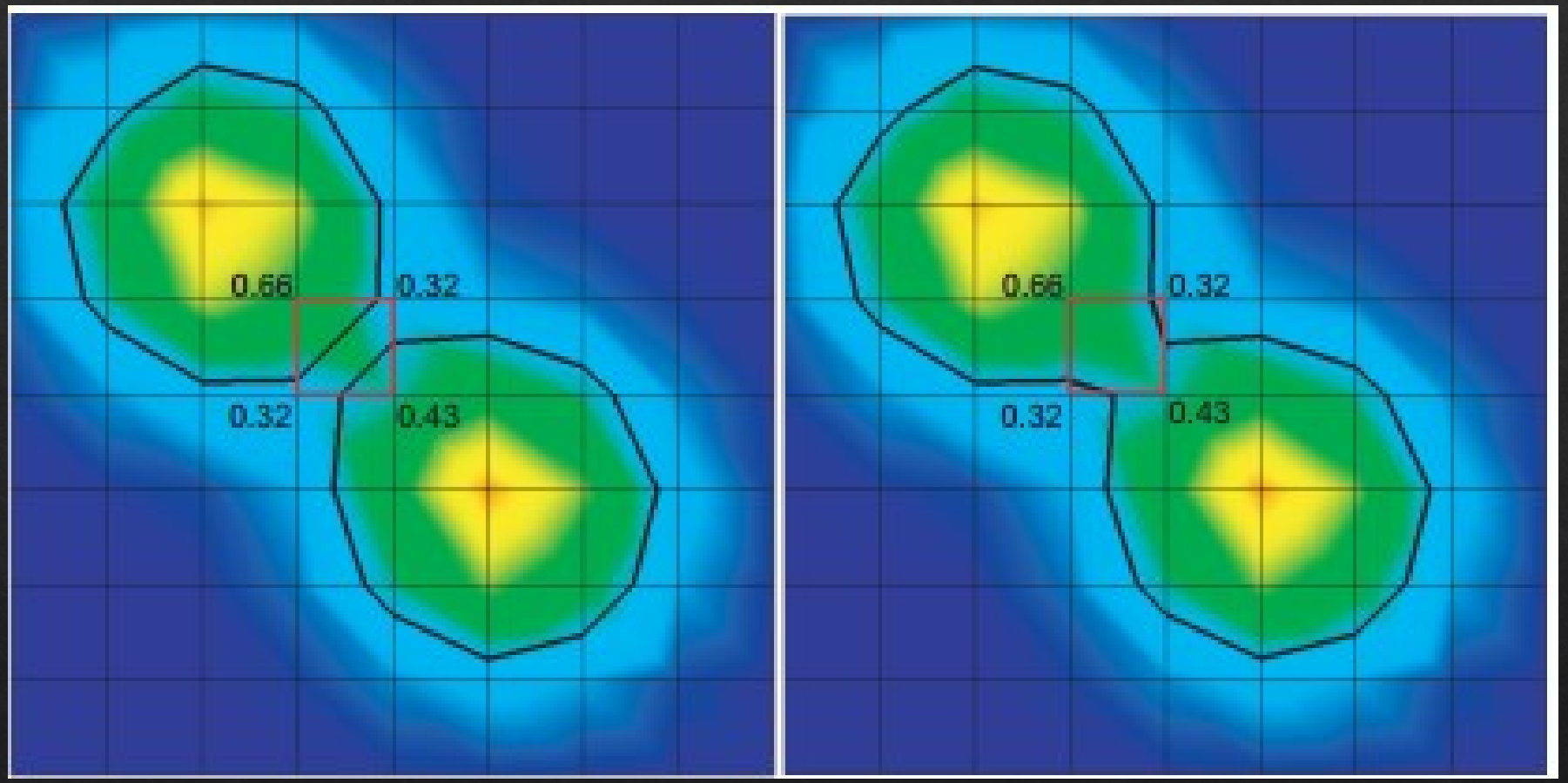


Resolving Ambiguity (interpolate)



Assume the isoline threshold is 0.
The bottom left and top right gets connected if we make the points above 0 to white.

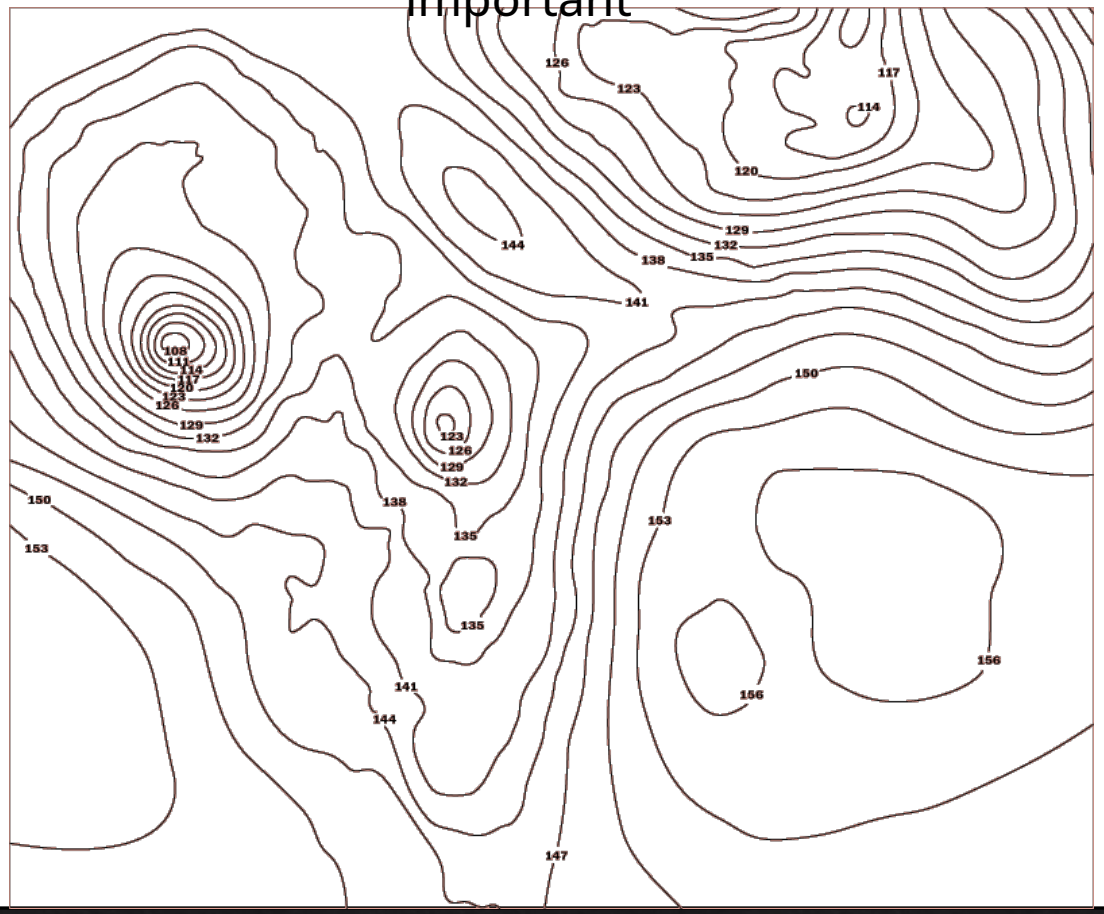
Ambiguity



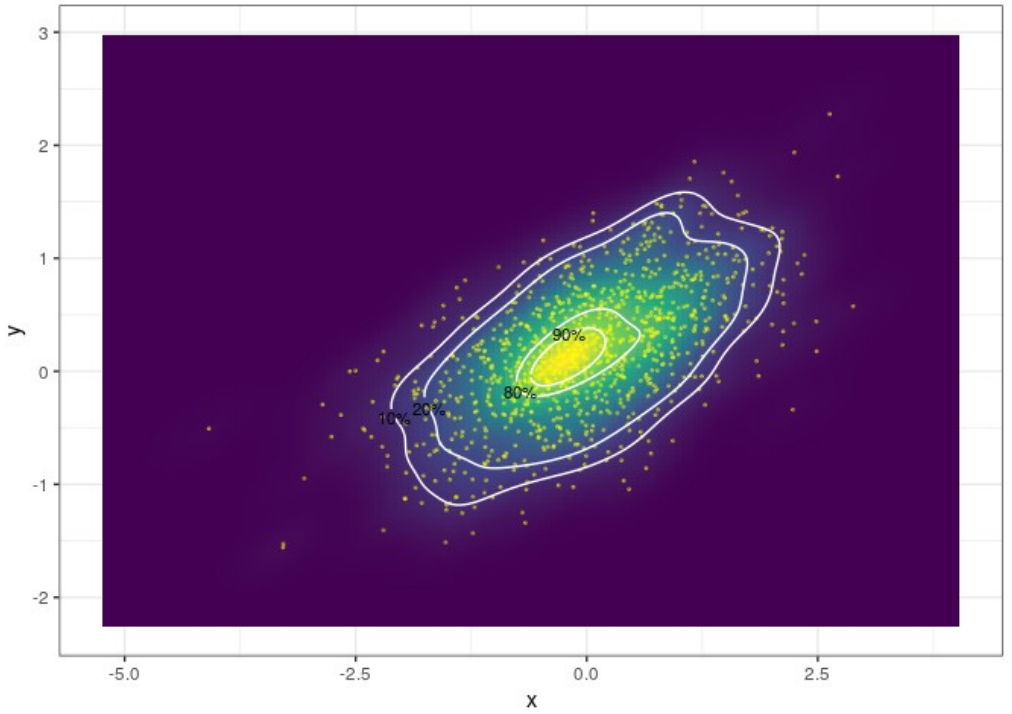
In practice, an implementation may choose either of the connection possibilities separately for each cell. If one has additional knowledge on the contour topology, e.g., that it must have a single connected component, this information can be used to discriminate between the two connection possibilities.

What threshold to use?

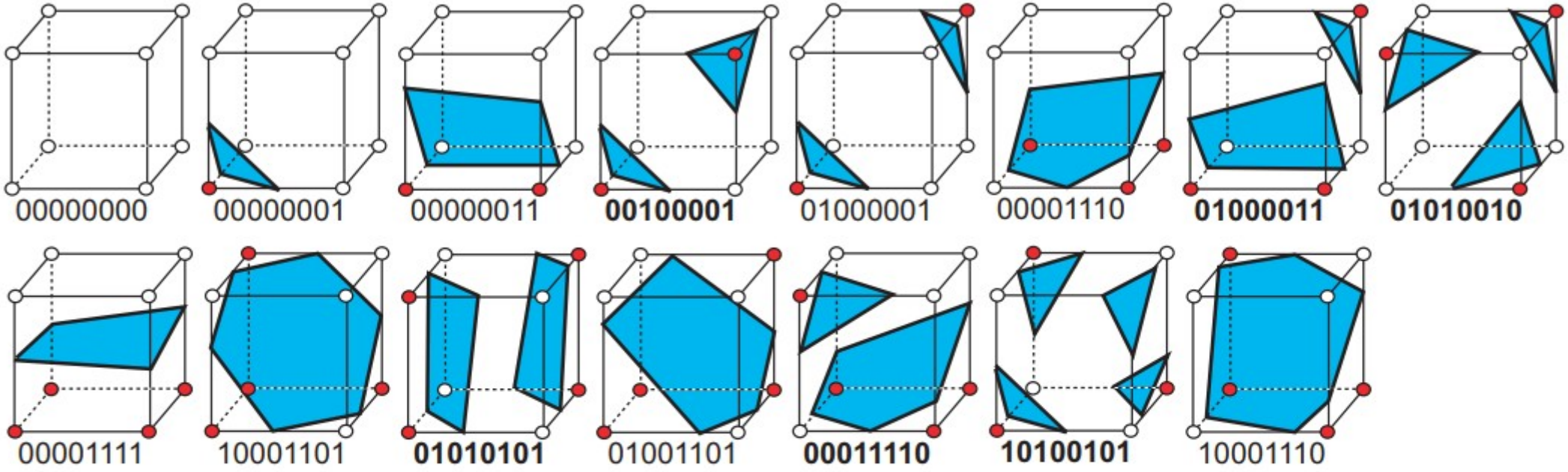
Use regular division when the gradient information is important

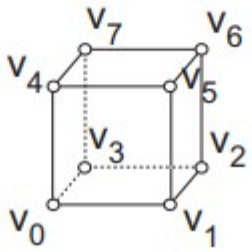


Use percentile when the frequency distribution over



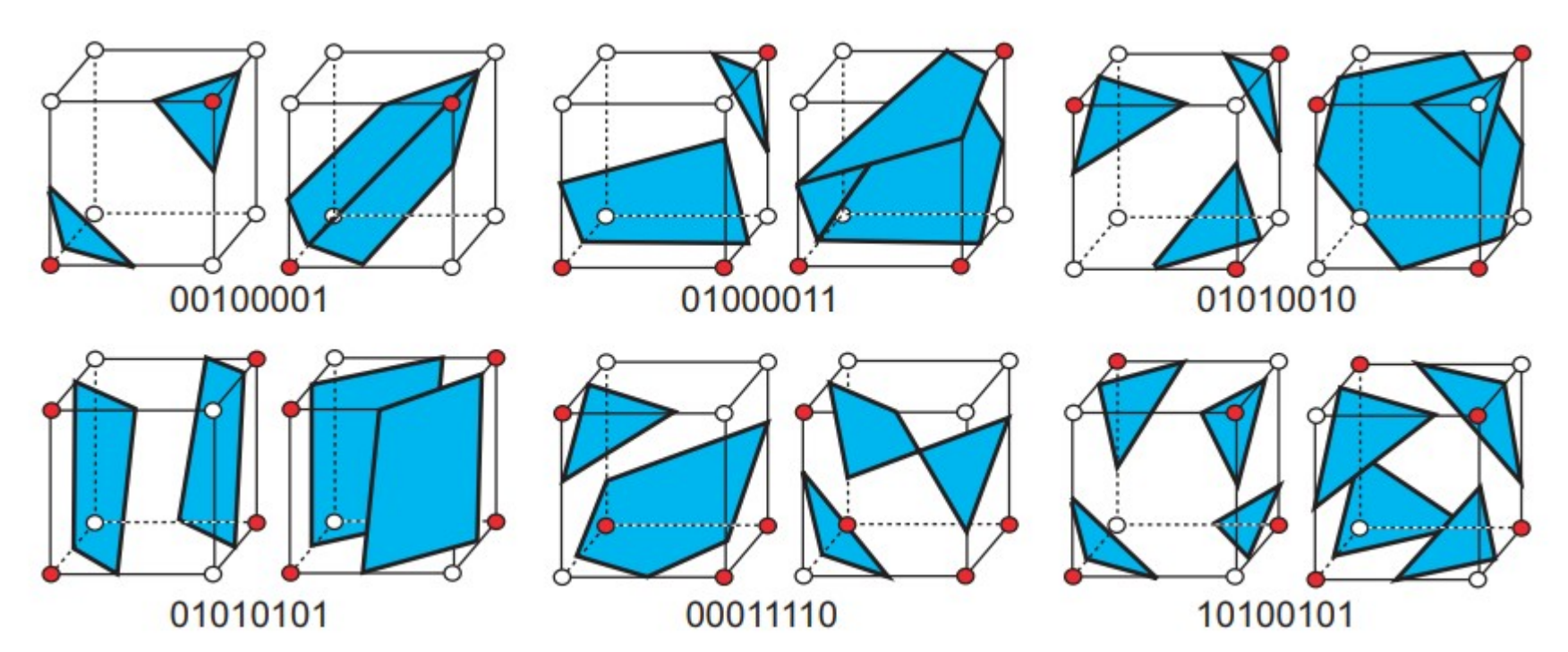
Marching Cube





8-bit code construction v_7 v_6 v_5 v_4 v_3 v_2 v_1 v_0

Ambiguity



Marching Cube in Practice

Two nested isosurfaces of
a tooth scan dataset