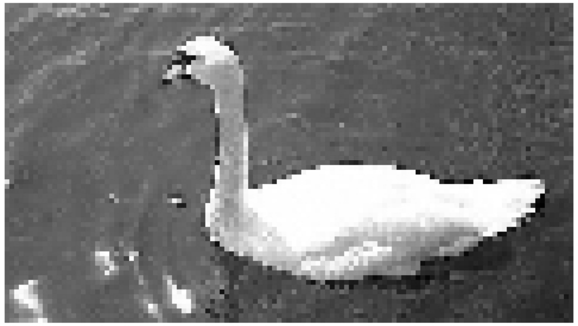
Clustering Color/Intensity



Group together pixels of similar color/intensity.

Agglomerative Clustering

- Cluster = connected pixels with similar color.
- Optimal decomposition may be hard.
  - For example, find k connected components of image with least color variation.
- Greedy algorithm to make this fast.

Clustering Algorithm

- Initialize: Each pixel is a region with color of that pixel and neighbors = neighboring pixels.
- Loop
  - Find adjacent two regions with most similar color.
  - Merge to form new region with:
    - all pixels of these regions
    - average color of these regions.
    - All neighbors of either region.
  - Stopping condition:
    - No regions similar
    - Find *k* regions.

Example

23	25	19	21	23
18	22	24	25	24
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

23	25	19	21	23
18	22	24	25	24
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

Example

23	25	19	21	23
18	22	24	25	24
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

23	25	19	21	23
18	22	24	25	24
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

Example

23	25	19	21	23
18	22	24	25	24
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

23	25	19	21	23
18	22	24.5	24.5	24
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

Example

23	25	19	21	23
18	22	24	25	24
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

23	25	19	21	23
18	22	24.33	24.33	24.33
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

Example

23	25	19	21	23
18	22	24	25	24
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

23	25	19	21	23
18	22	24.33	24.33	24.33
19.5	19.5	26	28	22
3	3	7	8	26
1	3	5	4	24

Example

23	25	19	21	23
18	22	24	25	24
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

23	25	19	21	23
18	22	24.33	24.33	24.33
19.5	19.5	26	28	22
3	3	7.5	7.5	26
1	3	5	4	24

Example

…

Example

23	25	19	21	23
18	22	24	25	24
20	19	26	28	22
3	3	7	8	26
1	3	5	4	24

22.9	22.9	22.9	22.9	22.9
22.9	22.9	22.9	22.9	22.9
22.9	22.9	22.9	22.9	22.9
4.25	4.25	4.25	4.25	22.9
4.25	4.25	4.25	4.25	22.9

Clustering complexity

- n pixels.
- Initializing:
  - O(n) time to compute regions.
- Loop:
  - O(n) time to find closest neighbors (could speed up).
  - O(n) time to update distance to all neighbors.
- At most n times through loop so O(n*n) time total.

Agglomerative Clustering: Discussion

- Start with definition of good clusters.
- Simple initialization.
- Greedy: take steps that seem to most improve clustering.
- This is a very general, reasonable strategy.
- Can be applied to almost any problem.
- But, not guaranteed to produce good quality answer.

Parametric Clustering

- Each cluster has a mean color/intensity, and a radius of possible colors.
- For intensity, this is just dividing histogram into regions.
- For color, like grouping 3D points into spheres.

K-means clustering

- Brute force difficult because many spheres, many pixels.
- Assume all spheres same radius; just need sphere centers.
- Iterative method.
  - If we knew centers, it would be easy to assign pixels to clusters.
  - If we knew which pixels in each cluster, it would be easy to find centers.
  - So guess centers, assign pixels to clusters, pick centers for clusters, assign pixels to clusters, ….
  - *matlab*

Why is this better?

- With a greedy algorithm, once we make a decision we cannot undo it.
- With an iterative algorithm, we can make changes.

K-means Algorithm

1. Initialize – Pick k random cluster centers
   – Pick centers *near* data. Heuristics: uniform distribution in range of data; randomly select data points.
2. Assign each point to nearest center.
3. Make each center average of pts assigned to it.
4. Go to step 2.

Let's consider a simple example. Suppose we want to cluster black and white intensities, and we have the intensities: 1 3 8 11. Suppose we start with centers c1 = 7 and c2=10. We assign 1, 3, 8 to c1, 11 to c2. Then we update c1 = (1+3+8)/3 = 4, c2 = 11. Then we assign 1,3 to c1 and 8 and 11 to c2. Then we update c1 = 2, c2 = 9 ½. Then the algorithm has converged. No assignments change, so the centers don't change.

K-means Properties

- We can think of this as trying to find the optimal solution to:
  - Given points p1… pn, find centers c1…ck
  - and find mapping f:{p1…pn}->{c1…ck}
  - that minimizes C = (p1-f(p1))^2 + …+ (pn-f(pn))^2.
- Every step reduces C.
  - The mean is the pt that minimizes sum of squared distance to a set of points. So changing the center to be the mean reduces this distance.
  - When we reassign a point to a closer center, we reduce its distance to its cluster center.
- Convergence: since there are only a finite set of possible assignments.

Local Minima

- However, algorithm might not find the best possible assignments and centers.
- Consider points 0, 20, 32.
  - K-means can converge to centers at 10, 32.
  - Or to centers at 0, 26.
- Heuristic solutions
  - Start with many random starting points and pick the best solution.

E-M

- Like K-means with soft assignment.
  - Assign point partly to all clusters based on probability it belongs to each.
  - Compute weighted averages (c_j) and variance (σ).

$$f_j(p_i) = \frac{e^{-\|p_i - c_j\|^2 / \sigma^2}}{\sum_j e^{-\|p_i - c_j\|^2 / \sigma^2}}$$

Cluster centers are c_j.

Example

- *Matlab: tutorial2*
- Fuzzy assignment allows cluster to creep towards nearby points and capture them.

E-M/K-Means domains

- Used color/intensity as example.
- But same methods can be applied whenever a group is described by parameters and distances.
- Lines (circles, ellipses); independent motions; textures (a little harder).