UNIVERSITÀ DEGLI STUDI DI PARMA

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

L11 - Image Segmentation

ANNO ACCADEMICO 2016/2017

K-Means Clustering

- Initialization:
  - choose *k* cluster centers
- Repeat:
  - **assignment step:**
    - For every point find its closest center
  - **update step:**
    - Update every center as the mean of its points
- Until:
  - The maximum number of iterations is reached, or
  - No changes during the assignment step, or
  - The average distortion per point drops very little

[Lloyd, 1957]

VISLAB.IT Extend Your Vision

K-Means Clustering

- **Input:** N examples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ ($\mathbf{x}_n \in \mathbb{R}^D$); the number of partitions K
- **Initialize:** K cluster centers $\mu_1, \ldots, \mu_K$. Several initialization options:
  - Randomly initialized anywhere in $\mathbb{R}^D$
  - Choose any K examples as the cluster centers
- **Iterate:**
  - Assign each of example $\mathbf{x}_n$ to its closest cluster center

$$\mathcal{C}_k = \{n : \quad k = \arg\min_k ||\mathbf{x}_n - \mu_k||^2\}$$

  ($\mathcal{C}_k$ is the set of examples closest to μ_k)
  - Recompute the new cluster centers μ_k (mean/centroid of the set $\mathcal{C}_k$)

$$\mu_k = \frac{1}{|\mathcal{C}_k|} \sum_{n \in \mathcal{C}_k} \mathbf{x}_n$$

  - Repeat while not converged

K-Means Clustering

The K-means objective function

- Let $\mu_1, \ldots, \mu_K$ be the K cluster centroids (means)
- Let $r_{nk} \in \{0, 1\}$ be indicator denoting whether point $\mathbf{x}_n$ belongs to cluster k
- K-means objective minimizes the total distortion (sum of distances of points from their cluster centers)

$$J(\mu, r) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \mu_k||^2$$

- Note: Exact optimization of the K-means objective is NP-hard
- The K-means algorithm is a heuristic that converges to a local optimum [1]

[1] L. Bottou and Y. Bengio. Convergence properties of the kmeans algorithm. NIPS, 1995.

Esercizio #1

- Implementare K-Means utilizzando come feature il **tono di grigio** dell'immagine [0-255]

- Scelta iniziale dei centri:
  - Equidistribuiti su 0-255
  - Random su 0-255

Esercizio #2

- Scelta dei centri in base all'istogramma:
  - Calcolare l'istogramma dei toni di grigio
  - Prendere un centro cluster per ogni **massimo locale** dell'istogramma

Esercizio #3

- Implementare K-Means utilizzando come feature i **3 canali di colore RGB**
  - Modificare oppurtunamente funzione distanza e costo

Esercizio #4

- Implementare K-Means utilizzando come feature il **tono di grigio** più la posizione riga-colonna:
  - (L,u,v,) -> [0 - 255][0 - (rows-1)][0 - (cols-1)]
  - Modificare opportunamente le funzioni distanza e costo

Superpixel

SLIC Superpixels:

- Feature space → intensity + position
  - L.a.b. color space
  - *limited region* (window 2*S)

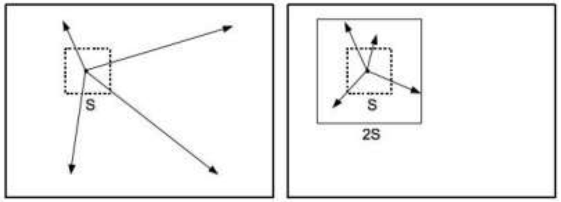
$$[l_k, a_k, b_k, x_k, y_k]$$

- Distance metric:

$$
\begin{aligned}
d_c &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\
d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\
D' &= \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2}.
\end{aligned}
$$

- Initialization:
  - Spatial grid (grid step = S)
- Iterate over centers and not points

Fig. 1: Images segmented using SLIC into superpixels of size 64, 256, and 1024 pixels (approximately).



(a) standard k-means searches the entire image

(b) SLIC searches a limited region

Achanta *et al.*, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, PAMI 2012.