ALGORITHMS FOR OPTIMIZATION AND INFERENCE – 2024
First Assignment

Giacomo Cirò

October 28, 2024

1 Theoretical Model

Given a subset of points in $\mathbb{R}^n$, we know the optimal centroid according to the sqaured L2-norm is the mean of the points in the subset. When choosing the optimal k centroids to cluster a given set of points into k subsets, we are then actually choosing among a finite number of possible optimal centroids, i.e., the means of all possible subsets of points.

With a similar optimality argument, we deduce that multiple occurrences of the same point must be assigned to the same centroid. Hence, we can further reduce to considering the means of all possible subsets of unique points. Nonetheless, these multiple occurrences are important as they skew the mean of the cluster they belong to and magnify the overall loss. Aware of this, we use the relative frequency of unique points as a weight when computing optimal centroids and the absolute frequency of unique points as a weight when computing the objective function.

Formally, we model the problem as follows.

Given a set of P points, denote with $i = 1, ..., N$ the unique points with coordinates $p_i \in \mathbb{R}^n$, clearly $N \leq P$. Denote with $j = 1, ..., 2^N - 1$ the possible centroids with coordinates c_j, obtained as follows:

$$c_j = \frac{1}{\sum_{i \in S_j} w_i} \sum_{i \in S_j} p_i \cdot w_i$$

Where S_j are all possible subsets of unique points and w_i their absolute frequency in the original set. We also define the total cost d_j of cluster S_j as:

$$d_j = \sum_{p_i \in S_j} w_i ||c_j - p_i||^2$$

And define the cluster matrix X as:

$$X_{ij} = \mathbb{1}_{S_j}(i)$$

Where $\mathbb{1}_{S_j}$ is the indicator function. We define a binary variable $y_j \in \{0, 1\}$, indicating whether or not we choose centroid j and the corresponding cluster S_j.

We can now model K-means as:

$$\min_{y_j} \sum_{j=1}^{2^N-1} y_j d_j$$

$$\text{s.t.}$$

$$\sum_{j=1}^{s^N-1} y_j X_{ij} = 1 \quad \forall i = 1, ..., N$$

$$\sum_{j=1}^{s^N-1} y_j = k$$

The constraints force to assign each point to exactly 1 centroid and to choose at most k centroids.

To optimize running time, we can perform the following semplification and avoid considering all the $2^N - 1$ possible clusters:

- discard all clusters such that the squared distance with respect to the generated centroid is greater than the objective function value found with K-means;
- discard all clusters with more than $N - k + 2$ elements. In fact, choosing one such cluster would lead to $k - 2$ points left, but only $k - 1$ clusters to be generated. It's now optimal to cluster $k - 2$ points on their own. This leaves one free cluster yet to be assigned. It's always optimal to remove one point from the first cluster and assign it to this last one.

2 Implementation

The folder `/submission/` contains all the code to compress an image using IP. Run the following commands:

1. `python3 recolor.py 20col.png 8col.png 8` to get the K-means compression using Lloyd algorithm;
2. `python3 generate_datafile.py` to generate the centroids and write the `.dat` file (either an integer is passed as argument or the program automatically uses the loss from the previously generated image as threshold when evaluating tentative clusters);
3. `glpsol --model main.mod --data main.dat > out.txt` to solve the IP and save the result;
4. `python3 visualize_compressions.py` to visualize the different compressions (IP-compressed image is generated by parsing `out.txt`).

Note: The folder `/additional_material/` contains files that were not strictly required by the submission, but that were generated in the process and I decided to include for completeness.
For example, `/additonal_material/all_compressions.png` is a visual comparison of the different types of compressions obtained, highlighting the loss and the size in KB.
Note that the folder `/additional_material/` can be safely deleted without affecting the runtime of the other files.