Kmeans Clustering approach to the Container Selection Problem

Caleb Andrade

July 7, 2016

Here we explore a Kmeans Clustering approach to find an approximate solution to the continuous version of the Container Selection Problem (CSP) in two dimensions. A two-factor approximation in expectation is proven if we assume uniformity on the data distribution. We provide also an example to show that the approximation can be arbitrarily bad in some cases, if no restrictions on the data distribution are imposed. Nonetheless, the algorithm seems to work well in practice, as the experimental part shows. The advantages of this method are its ease of implementation and fast linear running time. It also generalizes to higher dimensions in a natural way.

Container Selection Problem (CSP)

Given a set of input points X in the plane and a budget k, the problem consists in finding a set S of k points in the plane so that the following objective function is minimized

$$\min_{S \in \mathbb{R}^2} \sum_{x \in X} \min_{\substack{s \in S \\ x \prec s}} \|s\|$$

where $x \prec s$ means s *dominates* x, that is, $x_1 \leq s_1$ and $x_2 \leq s_2$, given $x = (x_1, x_2)$ and $s = (s_1, s_2)$. Also, $\|s\|$ denotes the L_1-norm.

A very useful analogy to this problem is to think of X as being a set of *item* sizes, and S a set of *box* sizes. Each item is to be packed in a box, one item per box, and only those boxes whose size *dominates* the item's size can pack such item. The objective function represents the packing cost, the sum of all of the box sizes used. The goal is to find a set of k box sizes that would minimize this packing cost.

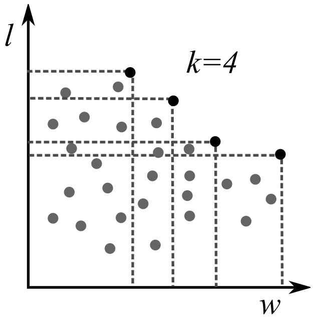

Figure 1. A data set X of item sizes (gray dots) dominated by a set S of $k = 4$ box sizes (black dots).

Intuitively, we want to *cluster* items according to their two-dimensional shape. Those that are closest in size will be grouped together and packed with the same box size, so Kmeans comes very naturally into play. We refer to Lloyd's Kmeans algorithm for more details *REFERENCE*.

CSP-Kmeans heuristic

Let CSP(X, k) be an instance of our problem, $X \subset \mathbb{R}^2$ being a set of n points and k the budget. Define $C(X, k, m, \mu)$ as the partitioning of X into k clusters produced by the Kmeans algorithm with the L_1-norm, m iterations, and $\mu \subset X$ as the initial set of k centroids. For brevity, denote this clustering as $\{C_i\}_{i=1}^{k}$.

It is a known fact that Kmeans minimizes the following function

$$\min_{\mu \subset \mathbb{R}^2} \sum_{x \in X} \min_{\mu_j \in \mu} d^2(x, \mu_j)$$

where $\mu = \{\mu_1, \ldots, \mu_k\}$ is a set of centroids and d is the euclidean distance. By minimizing the sum of squared distances this objective function captures the notion of closeness in the plane for groups of points (clusters), however, although this might seem to be a desirable property for our purposes, there are examples to show that this is not always the case. The question is then why Kmeans is a good approach. A first answer is that it induces a feasible solution in linear time, and still, we believe that the notion of close points in the same cluster is important for the most part.

Now, for each cluster C_i define its *corner* as $c_i = (\max x_1, \min x_2)$ over all $(x_1, x_2) \in C_i$ (this is the point that dominates all points in C_i with minimum L_1 norm), so the solution to our problem is the set of corners

$$S_c = \{c_i\}$$

with cost

$$\sum_{x \in X} \min_{\substack{c_i \in S_c \\ x \prec c_i}} \|c_i\|$$

Note that after applying Kmeans, it is possible that a point in some cluster C_i is closest to c_j than to c_i, so this point gets reassigned to cluster C_j. In this case we are breaking Kmeans local minima, but we are improving our CSP objective function, this means that, Kmeans optimal clustering is not equal to CSP optimal clustering in general.

Let $S = \{s_j\}$ be an optimal solution for CSP(X, k). Let us define

$$||X|| = \sum_{x \in X} ||x||$$

It should be straightforward to see that

$$\sum_{x \in X} \min_{x \prec s_j} ||s_j|| = \sum_{x \in X} \min_{x \prec s_j} d(x, s_j) + ||X||$$

This is graphically described in Figure 1.

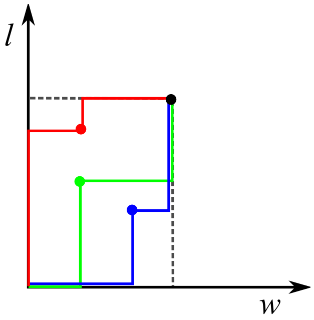

Figure 2. The objective function of CSP can be decomposed in two parts, the sum of the distances from every point to its closest dominant corner, plus the sum of all points' norms, as shown above for a single cluster.

This means that a lower bound for CSP(X, k) is $||X||$. We use this lower bound in the following observation

$$\sum_{i=1}^{k} |C_i| \cdot ||\mu_i|| = ||X||$$

which follows from the fact that

$$||\mu_i|| = \frac{x_{i_1} + x_{i_2} + \cdots + y_{i_1} + y_{i_1} + \ldots}{|C_i|}$$

Now, let y be the point that dominates a set of points Y with the smallest L_1-norm, and define its *container box* as $B(Y) = \{p \in \mathbb{R}^2 | p \prec y\}$. Now, suppose that the input points X are uniformly distributed in $B(X)$. For any cluster C_i we argue that

$$\frac{c_i}{2} \prec E[\mu_i]$$

which translates into the following inequality

$$||\frac{c_i}{2}|| \leq E[||\mu_i||]$$

The equality follows from the fact that we are assuming that the data is uniformly distributed in $B(X)$. Now, every cluster's container box will overlap with other container boxes on its SE corner, and because input points are assigned to the closest dominant NE corner, some container boxes will have some of its dominated input points being assigned to a different dominant corner, thus, pushing the centroid towards NE. This justifies the inequality for such cases. The only exception to this is the case when there is a container box strictly contained in all the other container boxes, but there can be only one such box, and for this case the equality still holds.

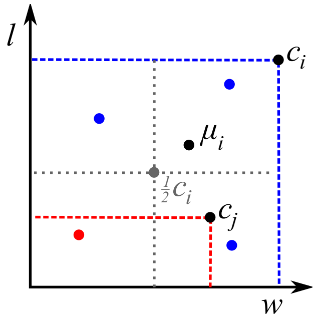

Figure 3. In the picture, a cluster C_i whose centroid is pushed towards its NE quadrant, because a smaller dominant containing box is a closest option to some of C_i's south-west points.

The advantage of this approach is its ease of implementation and fast running time $O(nkm)$, and the fact that the algorithm is naturally extended to any dimension.

In the current implementation, the k initial centroids are selected as those points in X with highest weight, the weight being the frequency of appearance of that point in the dataset. We think this is a good approach, because a high frequency point could potentially increase the objective's function value, so we want the dominant point of its cluster to be as closest as possible. Below, an example for an instance problem of 3108 points (with weights, depicted as the point's size) and $k = 15$, Figure 4. Figure 5 depicts the bad example.

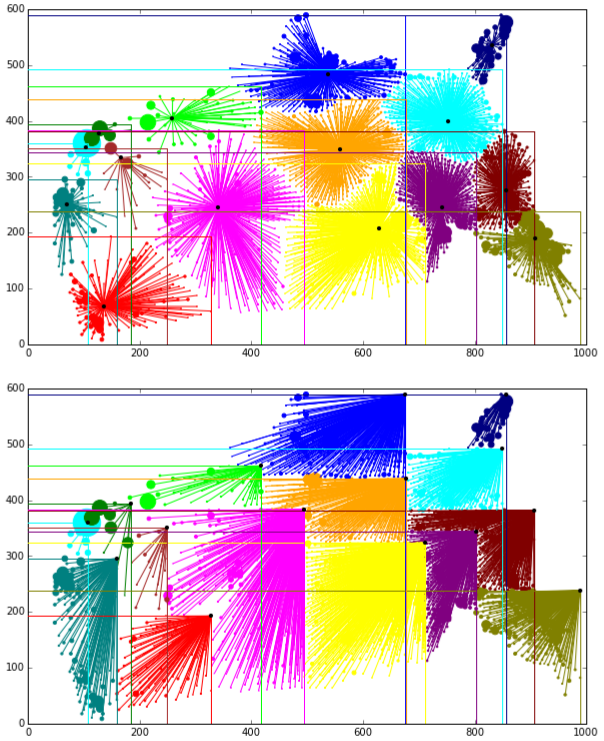

Figure 4. In the picture, a data set of United States' counties population is clustered according to the method described here. First apply Kmeans, then reassign points according to its closest dominant point.

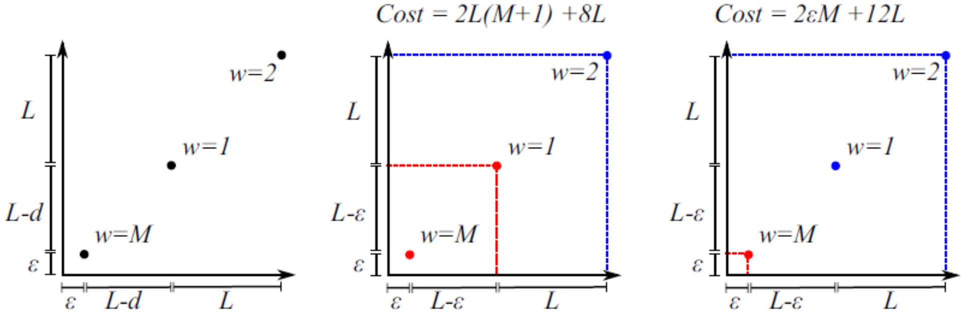

Figure 5. In the picture, If we set $\epsilon = 1/M$, the difference between kmeans and OPT is: $2LM + 10L - 12L - 2 = 2L(M-1) - 2$. If we take the ratio with respect to OPT we get $L(M-1) - 1/6L + 1$. This ratio grows arbitrarily as M grows.