k-means++

Author and editors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Abstract

k-means++ is a randomized rule for the initial cluster centroids of Lloyd's algorithm: the first is a data point drawn uniformly at random, each further one a data point drawn with probability proportional to its squared Euclidean distance to the nearest cluster centroid already chosen. The initial cluster centroids are thereby spread across the dataset, so that Lloyd's algorithm rarely starts with several of them in the same cluster. The expected clustering error of the initial cluster centroids is at most $8(\ln k + 2)$ times the smallest clustering error attainable with k cluster centroids, for every dataset and before a single iteration of Lloyd's algorithm; uniform seeding has no such guarantee. Since Lloyd's algorithm never increases the clustering error, the bound also holds for its output. The seeding changes neither the fixed-point iteration of Lloyd's algorithm nor its fixed points; it selects the starting point and thereby which fixed point is reached. On the patches of a photograph with three clusters, Lloyd's algorithm reaches the best clustering found from 64% of the k-means++ seeds and from 48% of the uniform seeds, and it needs fewer iterations.

Definition

Lloyd's algorithm for k-means needs k initial cluster centroids before its first iteration, and the clustering it delivers depends on them. Consider a photograph of the Oetscher massif cut into 972 square patches, each a data point with a feature vector $\mathbf{x} \in \mathbb{R}^2$ that holds its average greenness and blueness. Vegetation, mountain and haze, and sky form three clusters of patches, and the best cluster centroids found for $k = 3$ have the clustering error 0.018. Started from three patches drawn uniformly at random, Lloyd's algorithm ends in 516 of 1000 runs at a different fixed point with clustering error 0.025: two cluster centroids share the vegetation, and the mountain patches are divided between the sky cluster centroid and one of the vegetation cluster centroids. k-means++ is a randomized rule for the initial cluster centroids that spreads them across the dataset (Arthur and Vassilvitskii, 2007). Started from its seeds, Lloyd's algorithm ends at the worse fixed point in 359 of 1000 runs.

The clustering error is not a convex function of the cluster centroids. Lloyd's algorithm never increases it and stops at a fixed point, which can be a local minimum rather than a global one; which fixed point is reached depends on the initial

cluster centroids (Bishop, 2006, Sect. 9.1; Hastie et al., 2009, Sect. 14.3.6). The usual remedy is to run Lloyd's algorithm from several initial choices and keep the run with the smallest clustering error (Hastie et al., 2009, Sect. 14.3.6), the simplest initial choice being k data points drawn uniformly at random (Bishop, 2006, Sect. 9.1). Uniform seeding tends to place several initial cluster centroids in the largest cluster: 566 of the 972 patches are vegetation, so all three uniform seeds are vegetation patches with probability 0.20. k-means++ draws the seeds one after the other and lets the data points far from the seeds chosen so far be the likely next ones (Algorithm 1). The first seed $\mathbf{w}^{(1)}$ is a data point drawn uniformly at random from the dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$. For $c = 2, \ldots, k$, the seed $\mathbf{w}^{(c)}$ is a data point drawn with probability proportional to its squared Euclidean distance to the nearest seed already chosen: it equals the data point $\mathbf{x}^{(r)}$, for $r = 1, \ldots, m$, with probability

$$\mathbb{P}\left(\mathbf{w}^{(c)} = \mathbf{x}^{(r)}\right) = \frac{\min_{c'=1,\ldots,c-1} \left\|\mathbf{x}^{(r)} - \mathbf{w}^{(c')}\right\|_2^2}{\sum_{r'=1}^{m} \min_{c'=1,\ldots,c-1} \left\|\mathbf{x}^{(r')} - \mathbf{w}^{(c')}\right\|_2^2}. \tag{1}$$

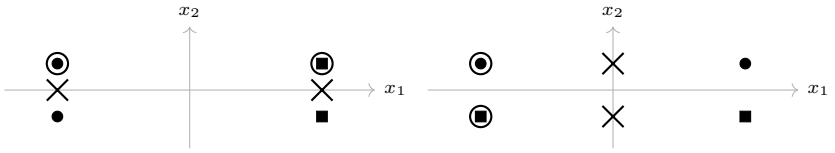
A data point that is already a seed has probability 0 under the seeding rule (1), and a data point twice as far from the seeds as another is twice as likely to be drawn. Given a vegetation patch as the first seed, the second seed is a sky patch with probability 0.74 under the seeding rule (1) and 0.21 under uniform seeding, the share of sky patches (Fig. 2(a)). It is another vegetation patch with probability 0.06 against 0.58. The seeds are the input of Lloyd's algorithm, and the combination of the seeding with Lloyd's algorithm is what Arthur and Vassilvitskii (2007) named k-means++. Ostrovsky et al. (2006) proposed the same seeding independently.

Algorithm 1 k-means++ seeding (Arthur and Vassilvitskii, 2007, Sect. 2.2)

Input: feature vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$, cluster count k
1: draw $\mathbf{w}^{(1)}$ uniformly at random from $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$
2: **for** $c = 2, \ldots, k$ **do**
3: draw $\mathbf{w}^{(c)}$ from $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$ with the probabilities (1)
4: **end for**
Output: initial cluster centroids $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)}$ for Lloyd's algorithm

Fig. 1 shows the effect of the seeds on four data points with feature vectors $(-5, 1)^\top$, $(-5, -1)^\top$, $(5, 1)^\top$, and $(5, -1)^\top$ and $k = 2$. Two seeds on opposite sides lead Lloyd's algorithm to the left/right split with cluster centroids $(\pm 5, 0)^\top$ and clustering error 1, the smallest possible. Two seeds on the same side assign each right-hand data point to the seed at the same height, so the first averaging step delivers the cluster centroids $(0, \pm 1)^\top$. These are a fixed point of Lloyd's algorithm, and the top/bottom split they define has clustering error 25. Uniform seeding draws two seeds on the same side with probability 1/3, two of the six pairs. Under k-means++, the second seed is the other data point on the side of

the first with probability $4/(4+100+104) = 1/52$ by the seeding rule (1). The expected clustering error after Lloyd's algorithm is therefore 9 under uniform seeding and 1.46 under k-means++.



(a) seeds on opposite sides: left/right split (b) seeds on the same side: top/bottom split

Figure 1: Four data points and two seeds (rings) for Lloyd's algorithm with $k = 2$; the crosses are the cluster centroids at the fixed point reached, and a data point is drawn as a filled circle or a filled square by the cluster it ends up in. (a) Seeds on opposite sides, drawn by k-means++ with probability 51/52, lead to the left/right split with clustering error 1. (b) Seeds on the same side, drawn by uniform seeding with probability 1/3, lead to the cluster centroids $(0, \pm 1)^\top$, a fixed point of Lloyd's algorithm whose top/bottom split has clustering error 25

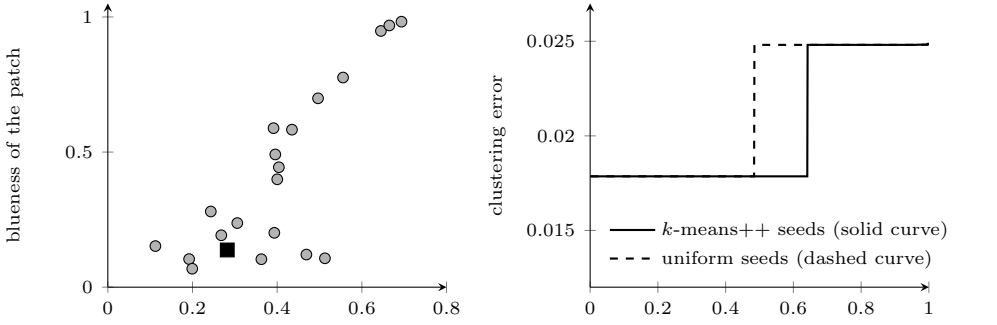
The seeds alone come with a guarantee. Let $E^\star$ denote the smallest clustering error that any k cluster centroids achieve on the dataset. The seeds $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)}$ of Algorithm 1 are random variables (RVs), and their expected clustering error satisfies, for every dataset and before a single iteration of Lloyd's algorithm (Arthur and Vassilvitskii, 2007, Thm. 1.1),

$$\mathbb{E}\left\{ \frac{1}{m} \sum_{r=1}^{m} \min_{c=1,\ldots,k} \left\| \mathbf{x}^{(r)} - \mathbf{w}^{(c)} \right\|_2^2 \right\} \leq 8\,(\ln k + 2)\, E^\star. \tag{2}$$

Since Lloyd's algorithm never increases the clustering error, its output started from the seeds satisfies the bound (2) as well. The logarithm cannot be removed: there are datasets on which the expected clustering error of the seeds is at least a constant multiple of $\ln k$ times $E^\star$ (Arthur and Vassilvitskii, 2007, Sect. 4). Uniform seeding has no bound of this kind: on the four data points of Fig. 1, the expected clustering error of two uniform seeds is 18, against 2.92 for k-means++ and the bound $8\,(\ln 2 + 2) \cdot 1 = 21.55$ of the guarantee (2). Moving the two pairs apart, to $(\pm 50, \pm 1)^\top$, keeps $E^\star = 1$ but raises the uniform value to 1668, while the k-means++ value is 3.00. On the patches, the mean clustering error of 1000 seed sets is 0.038 for k-means++ and 0.086 for uniform seeding, 2.1 and 4.8 times the best value found; the worst k-means++ seed set has clustering error 0.149, the worst uniform seed set 0.551.

The seeding changes nothing about Lloyd's algorithm itself. With the cluster centroids collected in one vector $\mathbf{w} = (\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)})$, Lloyd's algorithm is the fixed-point iteration $\mathbf{w}^{(t+1)} = \mathcal{F}(\mathbf{w}^{(t)})$ for $t = 0, 1, \ldots$. The operator $\mathcal{F}$ applies one assignment step and one averaging step. Its fixed points are cluster centroids that each equal the mean of the data points assigned to them, and

its convergence follows from the monotone decrease of the clustering error. k-means++ only chooses the starting point $\mathbf{w}^{(0)}$, and with it the fixed point that is reached and the number of iterations to reach it: on the patches, Lloyd's algorithm needs 10.0 iterations on average from k-means++ seeds and 12.7 from uniform seeds. The seeding itself is not an iterative method but k draws, so no fixed-point view applies to it; its output is an RV, and the guarantee (2) is a statement about its expectation. Nor does the seeding exclude the worse fixed point; it makes it rarer (Fig. 2(b)). On the patches, 35.9% of the runs from k-means++ seeds and 51.6% of the runs from uniform seeds end at the clustering error 0.025, so with five restarts, keeping the run with the smallest clustering error, all five miss the best clustering with probability 0.6% against 3.7%. In terms of empirical risk minimization (ERM), k-means minimizes the training error of a hypothesis that maps a feature vector to its nearest cluster centroid; the seeding changes neither the hypothesis space nor the loss function, only the optimization method and thereby the local minimum of the training error that it finds.



(a) 20 of the 972 patches and the first seed

(b) clustering error after Lloyd's algorithm, sorted

Figure 2: (a) Every 49th of the 972 patches, 20 patches, drawn as circles whose area is proportional to the probability (1) of being the second seed when the first seed is the vegetation patch marked by the filled square; the three largest circles are sky patches. (b) The clustering error after Lloyd's algorithm in 1000 runs from k-means++ seeds (solid curve) and 1000 runs from uniform seeds (dashed curve), each sorted in increasing order against the fraction of the runs; the step from 0.018 to 0.025 sits at 64% of the k-means++ runs and at 48% of the uniform runs. Data generated by `pythondemos/kmeanspp.py`

Painting each patch of the photograph in the average color of its cluster is image segmentation, and the two fixed points give two different segmentations. The better one separates sky, mountain, and vegetation; the worse one paints the 204 sky patches and 91 mountain patches in one color and cuts the vegetation in two. k-means++ is the default initialization of k-means in scikit-learn (Pedregosa et al., 2011).

Cross References

k-means, Lloyd's algorithm, cluster centroid, clustering error, clustering, cluster, hard clustering, fixed point, fixed-point iteration, Euclidean distance, minimum, convex, ERM, image segmentation

References

1. Arthur and Vassilvitskii (2007). k-means++: the advantages of careful seeding. In: Proc. of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027–1035. Society for Industrial and Applied Mathematics Philadelphia.

2. Bishop (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media, New York, NY, USA.

3. Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12(85), pp. 2825–2830.

4. Ostrovsky et al. (2006). The effectiveness of Lloyd-type methods for the k-means problem. In: Proc. 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 165–176.

5. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.