K

K-Means Clustering

Xin Jin[1] and Jiawei Han[2]
[1]PayPal Inc., San Jose, CA, USA
[2]University of Illinois at Urbana-Champaign, Urbana, IL, USA

Abstract

K-Means Clustering is a popular clustering algorithm with local optimization. In order to improve its performance, researchers have proposed methods for better initialization and faster computation.

Synonyms

Cluster initialization; Iterative computation; K-means clustering

Definition

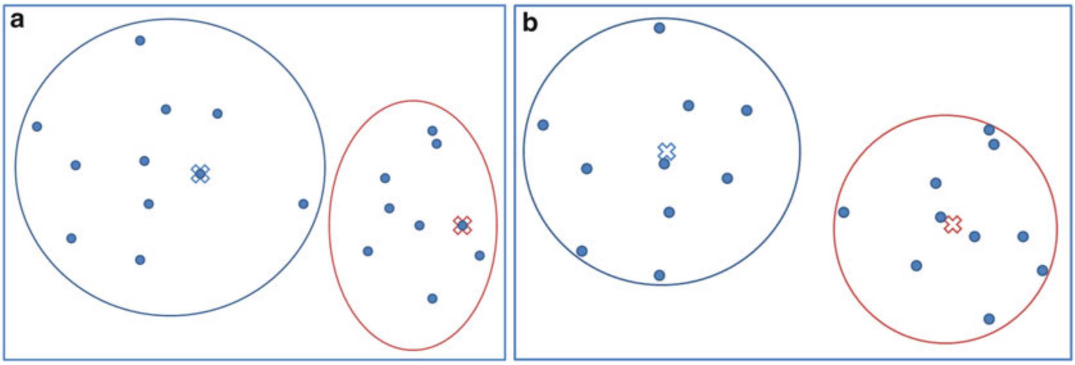
K-means (Lloyd 1957; MacQueen 1967) is a popular data clustering method, widely used in many applications. Algorithm 1 shows the procedure of K-means clustering. The basic idea of the K-means clustering is that given an initial but not optimal clustering, relocate each point to its new nearest center, update the clustering centers by calculating the mean of the member points, and repeat the relocating-and-updating process until converge criteria (such as predefined number of iterations, difference on the value of the distortion function) are satisfied.

The task of initialization is to form the initial K clusters. Many initializing techniques have been proposed, from simple methods, such as choosing the first K data points, Forgy initialization (randomly choosing K data points in the dataset), and random partitions (dividing the data points randomly into K subsets), to more sophisticated methods, such as density-based initialization, intelligent initialization, furthest-first initialization (FF for short, it works by picking first center point randomly and then adding more center points which are furthest from existing ones), and subset furthest-first (SFF) initialization. For more details, refer to paper Steinley and Brusco (2007) which provides a survey and comparison of over 12 initialization methods.

Algorithm 1 *K*-means clustering algorithm

```
Input: $K$, number of clusters; $D$, a data set of $N$ points
Output: A set of $K$ clusters
Initialization.
repeat
    for each point $p$ in $D$ do
        find the nearest center and assign $p$ to the corresponding cluster.
    end for
    update clusters by calculating new centers using mean of the members.
until stop-iteration criteria satisfied
return clustering result.
```

C. Sammut, G.I. Webb (eds.), *Encyclopedia of Machine Learning and Data Mining*,
DOI 10.1007/978-1-4899-7502-7_431-1



K-Means Clustering, Fig. 1 K-means clustering example ($K = 2$). The center of each cluster is marked by "x." (**a**) Initialization. (**b**) Re-assignment

Figure 1 shows an example of K-means clustering on a set of points, with $K = 2$. The clusters are initialized by randomly selecting two points as centers.

Complexity analysis. Let N be the number of points, D the number of dimensions, and K the number of centers. Suppose the algorithm runs I iterations to converge. The space complexity of K-means clustering algorithm is $O(N(D + K))$. Based on the number of distance calculations, the time complexity of K-means is $O(NKI)$.

Fast Computation for Large-Scale Data

For large-scale data clustering, K-means algorithm spends the majority of the time on the numerous distance calculations between the points and the centers. Many algorithms have been proposed to handle this problem, such as PDS (Bei and Gray 1985), TIE (Chen and Hsieh 1991), Elkan (Beil et al. 2003), MPS (Ra and Kim 1993), kd-tree K-means (Pelleg and Moore 1999), HKM (Nister and Stewenius 2006), GT (Kaukoranta et al. 2000), CGAUDC (Lai et al. 2008), and GAD (Jin et al. 2011).

PDS (partial distortion search) (Bei and Gray 1985) cumulatively computes the distance between the point and a candidate center by summing up the differences at each dimension. TIE (triangular inequality elimination) (Chen and Hsieh 1991) prunes candidate centers based on the triangle inequality of metric distance. MPS (mean-distance-ordered partial search) (Ra and Kim 1993) uses sorting to initially guess the center whose mean value is closest to that of the current point and prune candidates via an inequality based on an Euclidean distance property.

In many large-scale applications, we need to perform large K clustering, and HKM (Nister and Stewenius 2006) and kd-tree K-means (Pelleg and Moore 1999) are fast algorithms which work for this *large cluster* problem because their time complexity on K is reduced from the original $O(K)$ in K-means to $O(\log(K))$.

HKM (Nister and Stewenius 2006) performs fast hierarchical K-means clustering. Instead of directly performing clustering on large clusters, HKM uses K-means for a small number of clusters at each node of a hierarchical tree.

The kd-tree K-means (Pelleg and Moore 1999) algorithm utilizes kd-tree to find the approximate nearest center in a way that is faster than brute force searching. Centers were split hierarchically from the root to the leaf nodes of the kd-tree; leaf nodes will contain similar centers. When searching for the nearest center, we only need to check leaf nodes which are most similar to the point.

Many fast algorithms are based on a strategy that filters out unnecessary distance calculations using metric properties and thus only work for metric distances. Another strategy called *activity* *detection* avoids the metric properties and works for both metric and non-metric distances. GT (Kaukoranta et al. 2000) utilizes point activity for fast clustering. CGAUDC (Lai et al. 2008) is an extension of GT and gets further improvement on performance. GAD (Jin et al. 2011) provides a general solution for utilizing activity detection for fast clustering.

Softwares

The following softwares have implementations of the K-means clustering algorithm:

- Weka. Open source data mining software in Java (Hall et al. 2009), from Machine Learning Group at the University of Waikato:
  http://www.cs.waikato.ac.nz/ml/weka/index.html
- Apache Mahout. Open source machine learning software in Java for use in Hadoop, with support on K-means, Fuzzy K-means, and streaming K-means:
  http://mahout.apache.org/users/clustering/k-means-clustering.html
- LNKnet Software. Written in C. A public domain software from MIT Lincoln Laboratory:
  http://www.ll.mit.edu/mission/communications/cyber/softwaretools/lnknet/lnknet.html
- R K-means. R package. It performs K-means clustering on a data matrix.
  http://stat.ethz.ch/R-manual/R-patched/library/stats/html/kmeans.html
- MLPack. A scalable C++ machine learning library.
  http://mlpack.org
- Scikit-Learn. An open source machine learning software written in Python.
  http://scikit-learn.org

Recommended Reading

Bei C-D, Gray RM (1985) An improvement of the minimum distortion encoding algorithm for vector quantization. IEEE Trans Commun 33:1132–1133

Beil F, Ester M, Xu X (2003) Using the triangle inequality to accelerate k-means. In: Twentieth international conference on machine learning (ICML'03), Washington, DC, pp 147–153

Chen S-H, Hsieh WM (1991) Fast algorithm for VQ codebook design. In: IEE Proceedings I-Communications, Speech and Vision, 138(5):357–362

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: an update. ACM SIGKDD Explor Newsl 11(1):10–18

Jin X, Kim S, Han J, Cao L, Yin Z (2011) A general framework for efficient clustering of large datasets based on activity detection. Stat Anal Data Min 4(1):11–29

Kaukoranta T, Franti P, Nevalainen O (2000) A fast exact gla based code vector activity detection. IEEE Trans Image Process 9(8):1337–1342

Lai JZC et al (2008) A fast VQ codebook generation algorithm using codeword displacement. Pattern Recognit 41(1):315–319

Lloyd SP (1957) Least squares quantization in pcm. Technical report RR-5497, Bell Lab, Sept 1957

MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Le Cam LM, Neyman J (eds) Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol 1. University of California Press, Berkeley, pp 281–297

Nister D, Stewenius H (2006) Scalable recognition with a vocabulary tree. In: CVPR06, New York

Pelleg D, Moore A (1999) Accelerating exact k-means algorithms with geometric reasoning. In: Proceedings of KDD'99, New York. ACM, pp 277–281

Ra S-W, Kim J-K (1993) A fast mean-distance-ordered partial codebook search algorithm for image vector quantization. IEEE Trans Circuits Syst 40:576–579

Steinley D, Brusco MJ (2007) Initializing k-means batch clustering: a critical evaluation of several techniques. J Classif 24(1):99–121