



WIKIPEDIA  
The Free Encyclopedia

[Main page](#)  
[Contents](#)  
[Featured content](#)  
[Current events](#)  
[Random article](#)  
[Donate to Wikipedia](#)  
[Wikipedia store](#)

Interaction

[Help](#)  
[About Wikipedia](#)  
[Community portal](#)  
[Recent changes](#)  
[Contact page](#)

Tools

[What links here](#)  
[Related changes](#)  
[Upload file](#)  
[Special pages](#)  
[Permanent link](#)  
[Page information](#)  
[Wikidata item](#)  
[Cite this page](#)

Print/export

[Create a book](#)  
[Download as PDF](#)  
[Printable version](#)

Languages

[日本語](#)  
[Русский](#)  
[Српски / srpski](#)  
[Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

# K-means++

From Wikipedia, the free encyclopedia

In [data mining](#), ***k*-means++**<sup>[1][2]</sup> is an algorithm for choosing the initial values (or "seeds") for the *k*-means [clustering](#) algorithm. It was proposed in 2007 by David Arthur and Sergei Vassilvitskii, as an approximation algorithm for the [NP-hard](#) *k*-means problem—a way of avoiding the sometimes poor clusterings found by the standard *k*-means algorithm. It is similar to the first of three seeding methods proposed, in independent work, in 2006<sup>[3]</sup> by Rafail Ostrovsky, Yuval Rabani, [Leonard Schulman](#) and Chaitanya Swamy. (The distribution of the first seed is different.)

## Contents

[\[hide\]](#)

- [1 Background](#)
- [2 Initialization algorithm](#)
- [3 Example bad case](#)
- [4 Applications](#)
- [5 Software](#)
- [6 References](#)

## Background

[\[edit\]](#)

The *k*-means problem is to find cluster centers that minimize the intra-class variance, i.e. the sum of squared distances from each data point being clustered to its cluster center (the center that is closest to it). Although finding an exact solution to the *k*-means problem for arbitrary input is NP-hard,<sup>[4]</sup> the standard approach to finding an approximate solution (often called [Lloyd's algorithm](#) or the *k*-means algorithm) is used widely and frequently finds reasonable solutions quickly.

However, the *k*-means algorithm has at least two major theoretic shortcomings:

- First, it has been shown that the worst case running time of the algorithm is super-polynomial in the input size.<sup>[5]</sup>
- Second, the approximation found can be arbitrarily bad with respect to the objective function compared to the optimal clustering.

The *k*-means++ algorithm addresses the second of these obstacles by specifying a procedure to initialize the cluster centers before proceeding with the standard *k*-means optimization iterations. With the *k*-means++ initialization, the algorithm is guaranteed to find a solution that is  $O(\log k)$  competitive to the optimal *k*-means solution.

## Initialization algorithm

[\[edit\]](#)

The intuition behind this approach is that spreading out the *k* initial cluster centers is a good thing: the first cluster center is chosen uniformly at random from the data points that are being clustered, after which each subsequent cluster center is chosen from the remaining data points with probability proportional to its squared distance from the point's closest existing cluster center.

The exact algorithm is as follows:

- Choose one center uniformly at random from among the data points.
- For each data point *x*, compute  $D(x)$ , the distance between *x* and the nearest center that has already been chosen.
- Choose one new data point at random as a new center, using a weighted probability distribution where a point *x* is chosen with probability proportional to  $D(x)^2$ .
- Repeat Steps 2 and 3 until *k* centers have been chosen.
- Now that the initial centers have been chosen, proceed using standard [k-means clustering](#).

This seeding method yields considerable improvement in the final error of *k*-means. Although the initial selection in the algorithm takes extra time, the *k*-means part itself converges very quickly after this seeding and thus the algorithm actually lowers the computation time. The authors tested their method with real and synthetic datasets and obtained typically 2-fold improvements in speed, and for certain datasets, close to 1000-fold

improvements in error. In these simulations the new method almost always performed at least as well as [vanilla  \$k\$ -means](#) in both speed and error.

Additionally, the authors calculate an approximation ratio for their algorithm. The  $k$ -means++ algorithm guarantees an approximation ratio  $O(\log k)$  in expectation (over the randomness of the algorithm), where  $k$  is the number of clusters used. This is in contrast to vanilla  $k$ -means, which can generate clusterings arbitrarily worse than the optimum.<sup>[6]</sup>

## Example bad case [\[edit\]](#)

To illustrate the potential of the  $k$ -means algorithm to perform arbitrarily poorly with respect to the objective function of minimizing the sum of squared distances of cluster points to the centroid of their assigned clusters, consider the example of four points in  $\mathbf{R}^2$  that form an axis-aligned rectangle whose width is greater than its height.

If  $k = 2$  and the two initial cluster centers lie at the midpoints of the top and bottom line segments of the rectangle formed by the four data points, the  $k$ -means algorithm converges immediately, without moving these cluster centers. Consequently, the two bottom data points are clustered together and the two data points forming the top of the rectangle are clustered together—a suboptimal clustering because the width of the rectangle is greater than its height.

Now, consider stretching the rectangle horizontally to an arbitrary width. The standard  $k$ -means algorithm will continue to cluster the points suboptimally, and by increasing the horizontal distance between the two data points in each cluster, we can make the algorithm perform arbitrarily poorly with respect to the  $k$ -means objective function.

## Applications [\[edit\]](#)

The  $k$ -means++ approach has been applied since its initial proposal. In a review by Shindler,<sup>[7]</sup> which includes many types of clustering algorithms, the method is said to successfully overcome some of the problems associated with other ways of defining initial cluster-centres for  $k$ -means clustering. Lee et al.<sup>[8]</sup> report an application of  $k$ -means++ to create geographical cluster of photographs based on the latitude and longitude information attached to the photos. An application to financial diversification is reported by Howard and Johansen.<sup>[9]</sup> Other support for the method and ongoing discussion is also available online.<sup>[10]</sup> Since the  $k$ -means++ initialization needs  $k$  passes over the data, it does not scale very well to large data sets. Bahman Bahmani et al. have proposed a scalable variant of  $k$ -means++ called  $k$ -means|| which provides the same theoretical guarantees and yet is highly scalable.<sup>[11]</sup>

## Software [\[edit\]](#)







- [Scikit-learn](#) has a K-Means implementation that uses  $k$ -means++ by default.
- [ELKI](#) data-mining framework contains multiple  $k$ -means variations, including  $k$ -means++ for seeding.
- [GNU R](#) includes  $k$ -means, and the "flexclust" package can do  $k$ -means++
- [OpenCV implementation](#) [↗](#)
- [Weka](#) contains  $k$ -means (with optional  $k$ -means++) and  $x$ -means clustering.
- [David Arthur's implementation](#) [↗](#)<sup>[[dead link](#)]</sup>
- [Apache Commons Math Java implementation](#) [↗](#)
- [CMU's GraphLab](#) [↗](#) [GraphLab](#) Efficient, open source clustering on multicore.



Wikibooks has a book on the topic of: ***K-Means++***

## References [\[edit\]](#)

- ↑ Arthur, D. and Vassilvitskii, S. (2007). "[k-means++: the advantages of careful seeding](#)" [↗](#) (PDF). *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035.
- ↑ <http://theory.stanford.edu/~sergei/slides/BATS-Means.pdf> [↗](#) Slides for presentation of method by Arthur, D. and Vassilvitskii, S.
- ↑ Ostrovsky, R., Rabani, Y., Schulman, L. J. and Swamy, C. (2006). "The Effectiveness of Lloyd-Type Methods for the  $k$ -Means Problem". *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE. pp. 165–174.
- ↑ Drineas, P. and Frieze, A. and Kannan, R. and Vempala, S. and Vinay, V. (2004). "Clustering Large Graphs via the Singular Value Decomposition". *Machine Learning* (Kluwer Academic Publishers Hingham, MA, USA) **56** (1–3): 9–33. doi:[10.1023/B:MACH.0000033113.59016.96](#) [↗](#).

5. <sup>^</sup> Arthur, D. and Vassilvitskii, S. (2006), "How slow is the  $k$ -means method?", *ACM New York, NY, USA*: 144–153
6. <sup>^</sup> T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, A. Wu "A Local Search Approximation Algorithm for  $k$ -Means Clustering"  2004 Computational Geometry: Theory and Applications.
7. <sup>^</sup> <http://web.archive.org/web/20110927100642/http://www.cs.ucla.edu/~shindler/shindler-kMedian-survey.pdf>  Approximation Algorithms for the Metric  $k$ -Median Problem
8. <sup>^</sup> <http://sir-lab.usc.edu/publications/2008-ICWSM2LEES.pdf>  Discovering Relationships among Tags and Geotags, 2007
9. <sup>^</sup> <http://www.cse.ohio-state.edu/~johanse/clusterings.pdf>  Clustering Techniques for Financial Diversification, March 2009
10. <sup>^</sup> <http://lingpipe-blog.com/2009/03/23/arthur-vassilvitskii-2007-kmeans-the-advantages-of-careful-seeding/>  Lingpipe Blog
11. <sup>^</sup> B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii "Scalable  $K$ -means++"  2012 Proceedings of the VLDB Endowment.

Categories: [Data clustering algorithms](#) | [Statistical algorithms](#)

This page was last modified on 10 February 2015, at 03:25.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

