k-Means Advantages and Disadvantages

Advantages of k-means

- **Relatively simple to implement.**
- **Scales to large data sets.**
- **Guarantees convergence.**
- **Can warm-start the positions of centroids.**
- **Easily adapts to new examples.**
- **Generalizes to clusters of different shapes and sizes, such as elliptical clusters.**

k-means Generalization

What happens when clusters are of different densities and sizes? Look at Figure 1. Compare the intuitive clusters on the left side with the clusters actually found by k-means on the right side. The comparison shows how k-means can stumble on certain datasets.

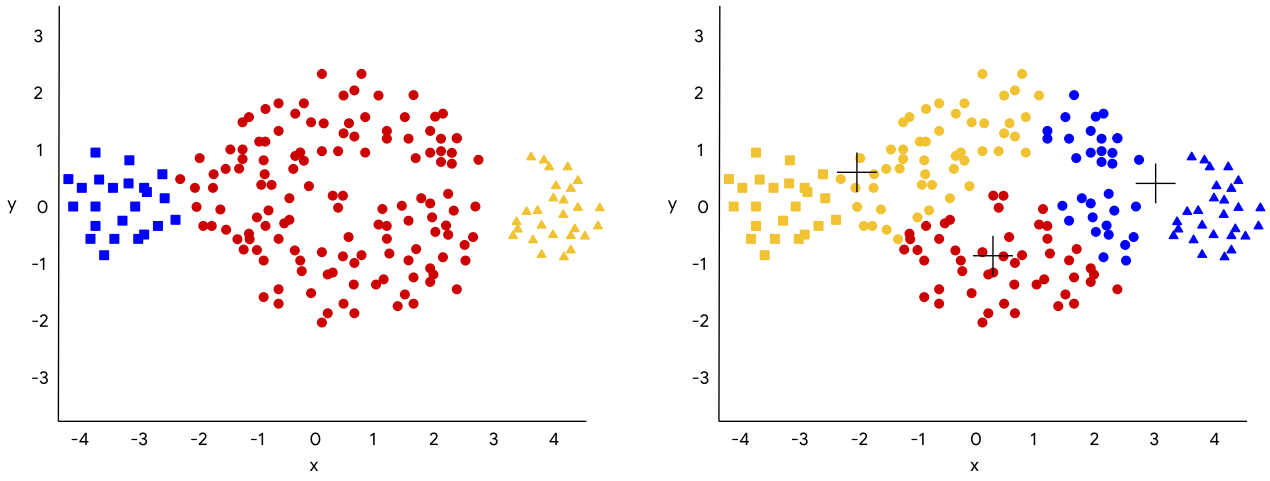

Figure 1: Ungeneralized k-means example.

To cluster naturally imbalanced clusters like the ones shown in Figure 1, you can adapt (generalize) k-means. In Figure 2, the lines show the cluster boundaries after generalizing k-means as:

- Left plot: No generalization, resulting in a non-intuitive cluster boundary.

- Center plot: Allow different cluster widths, resulting in more intuitive clusters of different sizes.
- Right plot: Besides different cluster widths, allow different widths per dimension, resulting in elliptical instead of spherical clusters, improving the result.

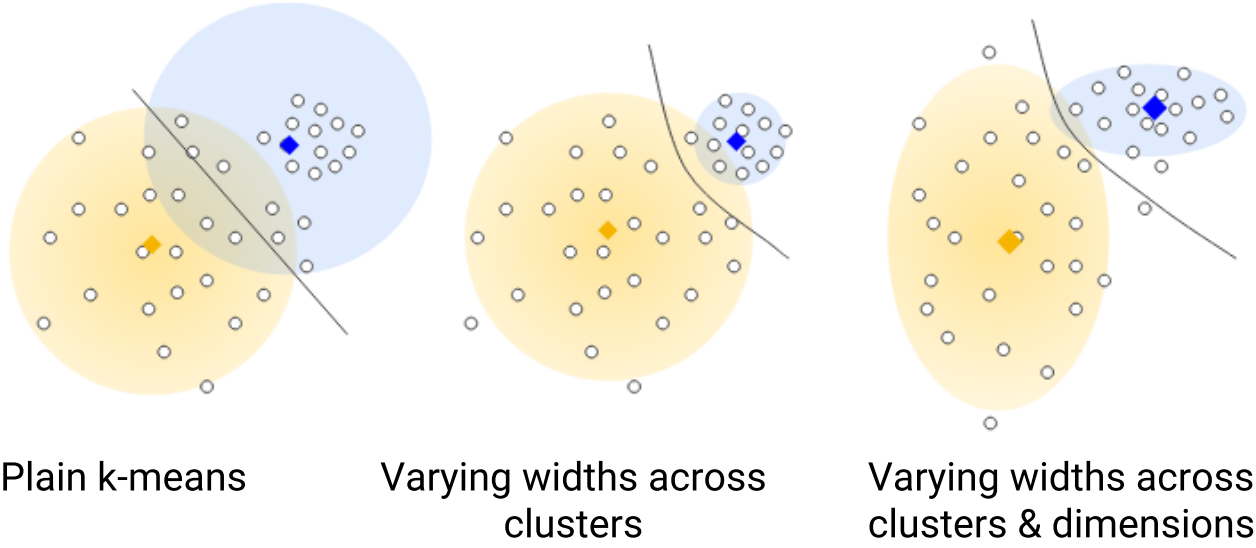

Figure 2: A spherical cluster example and a non-spherical cluster example.

While this course doesn't dive into how to generalize k-means, remember that the ease of modifying k-means is another reason why it's powerful. For information on generalizing k-means, see Clustering – K-means Gaussian mixture models (http://www.cs.cmu.edu/%7Eguestrin/Class/10701-S07/Slides/clustering.pdf) by Carlos Guestrin from Carnegie Mellon University.

Disadvantages of k-means

Choosing k manually.

Use the "Loss vs. Clusters" plot to find the optimal (k), as discussed in Interpret Results (/machine-learning/clustering/interpret).

Being dependent on initial values.

For a low k, you can mitigate this dependence by running k-means several times with different initial values and picking the best result. As k increases, you need advanced versions of k-means to pick better values of the initial centroids (called **k-means seeding**). For a full discussion of k- means seeding see, *A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm* (https://arxiv.org/abs/1209.1960) by M. Emre Celebi, Hassan A. Kingravi, Patricio A. Vela.

Clustering data of varying sizes and density.

k-means has trouble clustering data where clusters are of varying sizes and density. To cluster such data, you need to generalize k-means as described in the Advantages (/machine-learning/clustering/algorithm/advantages-disadvantages#advantages_of_k-means) section.

Clustering outliers.

Centroids can be dragged by outliers, or outliers might get their own cluster instead of being ignored. Consider removing or clipping outliers before clustering.

Scaling with number of dimensions.

As the number of dimensions increases, a distance-based similarity measure converges to a constant value between any given examples. Reduce dimensionality either by using **PCA** (https://wikipedia.org/wiki/Principal_component_analysis) on the feature data, or by using "spectral clustering" to modify the clustering algorithm as explained below.

Curse of Dimensionality and Spectral Clustering

These plots show how the ratio of the standard deviation to the mean of distance between examples decreases as the number of dimensions increases. This convergence means k-means becomes less effective at distinguishing between examples. This negative consequence of high-dimensional data is called the curse of dimensionality.

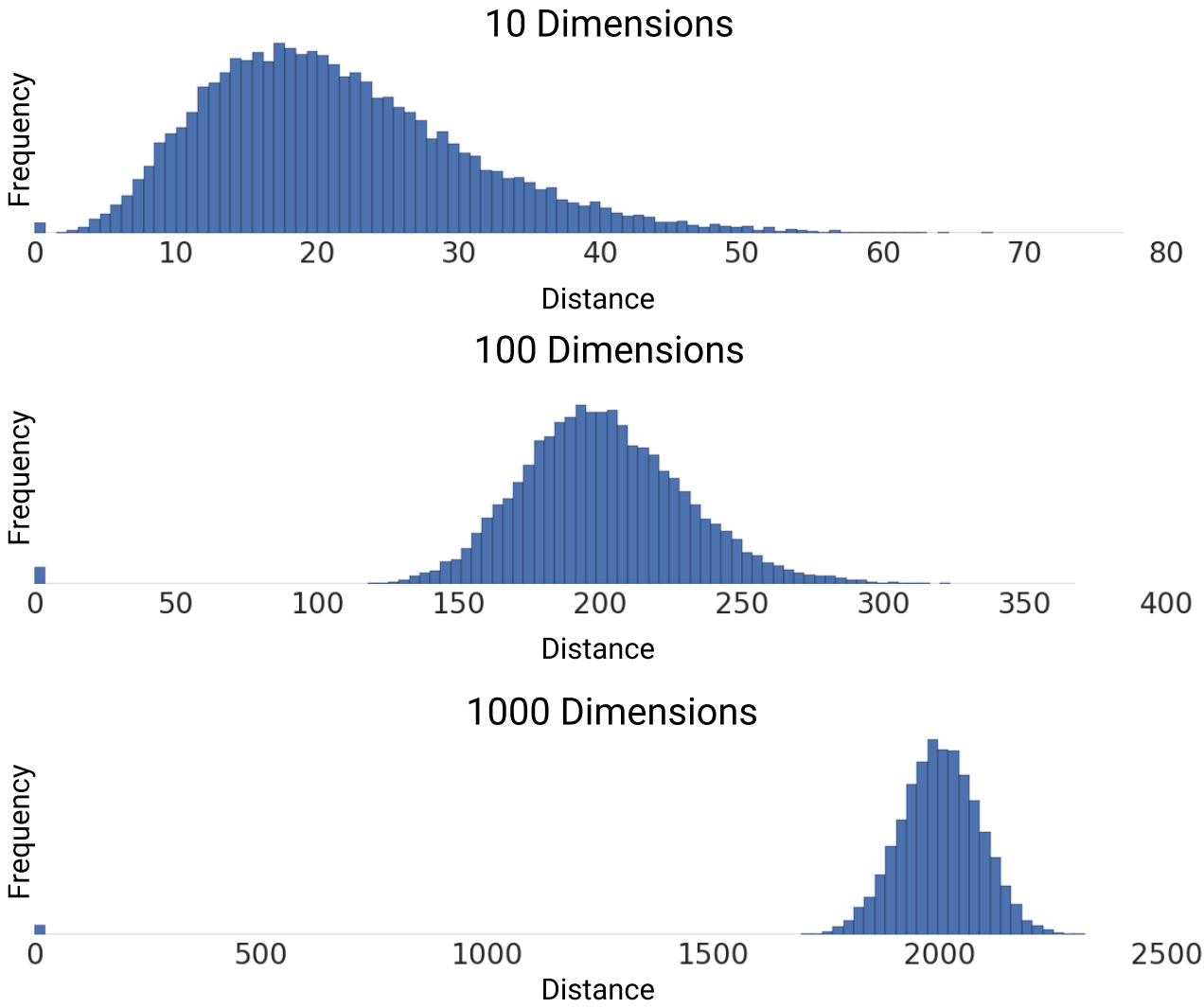

Figure 3: A demonstration of the curse of dimensionality. Each plot shows the pairwise distances between 200 random points.

Spectral clustering avoids the curse of dimensionality by adding a pre-clustering step to your algorithm:

1. Reduce the dimensionality of feature data by using PCA.
2. Project all data points into the lower-dimensional subspace.
3. Cluster the data in this subspace by using your chosen algorithm.

Therefore, spectral clustering is not a separate clustering algorithm but a pre- clustering step that you can use with any clustering algorithm. The details of spectral clustering are complicated. See *A Tutorial on Spectral Clustering* (https://github.com/petermartigny/Advanced-Machine-Learning/blob/master/DataLab2/Luxburg07_tutorial_4488%5B0%5D.pdf) by Ulrike von Luxburg.

ɪrms:

spectral clustering

Previous
← Interpret Results (/machine-learning/clustering/interpret)

Next
Implement k-Means (/machine-learning/clustering/implementation) →

Last updated 2021-01-13 UTC.