K-Density-Based-Means (KDB-Means)

Grace Bai, Aneesh Pabolu
Dr. Yilmaz Period 4 ML
January 2025

Table of contents

01

Algorithm

A deep dive into KDB-Means

The problem…

K-Means

An unsupervised machine learning algorithm that partitions data into k number of distinct clusters by assigning points to the nearest centroid and updating centroids periodically based on distances between data points in their respective cluster.

Its Benefits:

- Scalable and Easy to implement
- Guaranteed convergence
- Interpretable clustering results

Its Drawbacks:

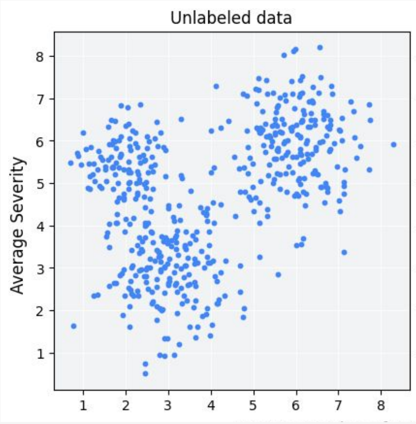
- Highly sensitive to centroid initialization
- Random initialization → unstable clustering results
- Poor initialization increases iterations and runtime

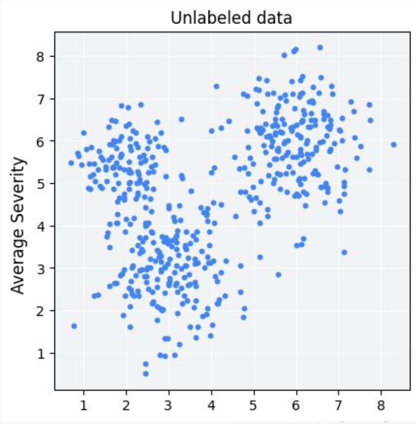
The Solution

KDB-Means

Aiming to solve the inherent issues of the normal K-Means algorithm, we present KDB-Means, a deterministic extension of K-Means that initializes centroids in high-density regions while enforcing separation. Combining local density estimates with distance-based selection to calculate centroids allows for faster convergence and improved clustering on unevenly distributed data.

Here is a quick example:





[4]

Algorithm and Pseudocode

Pseudocode:

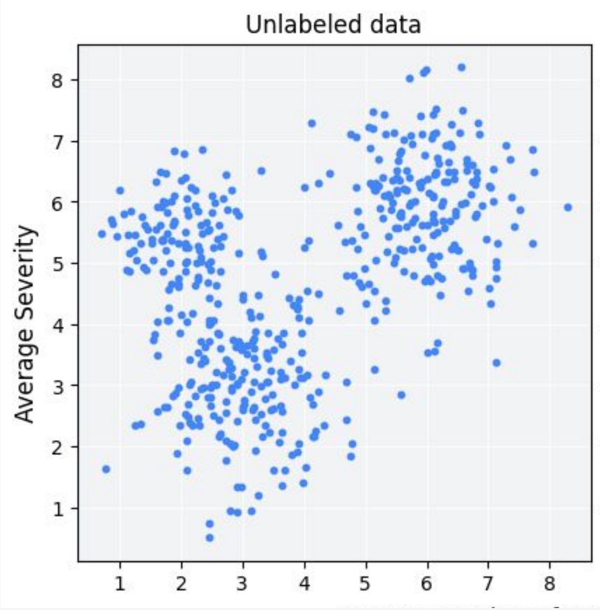
1. Normalize features of the dataset
2. Computer measure of spread of the entire dataset with the average distance
3. Define density radius r_o based on the spread of the dataset; average distance * an alpha value (0.2 through trial-and-error)
4. For each instance x_i in dataset X:

    Look in neighborhood around data point in the radius r_o: count how many points fall inside that neighborhood, record as density value

5. Select the point with highest density as first centroid
6. Until k centroids are picked:

    Choose next centroid by calculating the distance of every node to its closest centroid, select the node with the highest density * distance^2 value

7. Record centroids and use in SciKit's K-Means algorithm



[16]

02

Related Work

K-Means++ and DBSCAN

K-Means++

A modification of the K-means algorithm that improves standard K-Means by carefully initializing centroids so they are well separated, reducing poor initializations and speeding up convergence.

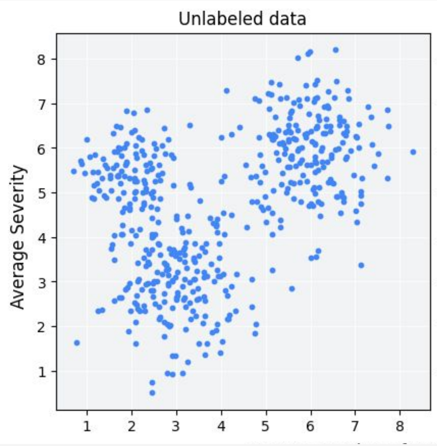
Its Benefits:

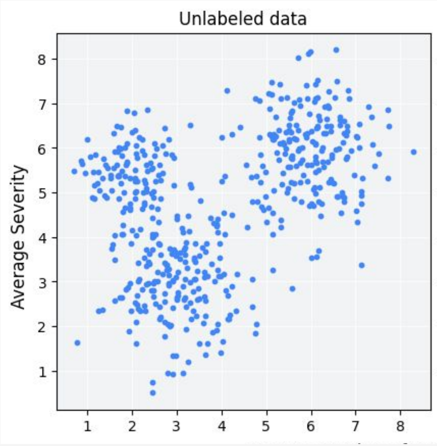
- Faster than regular K-Means
- Guaranteed convergence
- Interpretable clustering results

Its Drawbacks:

- Does not take valuable info such as local data density into account

Example:





[16]

DBSCAN

DBSCAN is clustering algorithm that clusters data by grouping dense regions of points and identifying sparse points as noise, without requiring the number of clusters in advance.

Benefits:

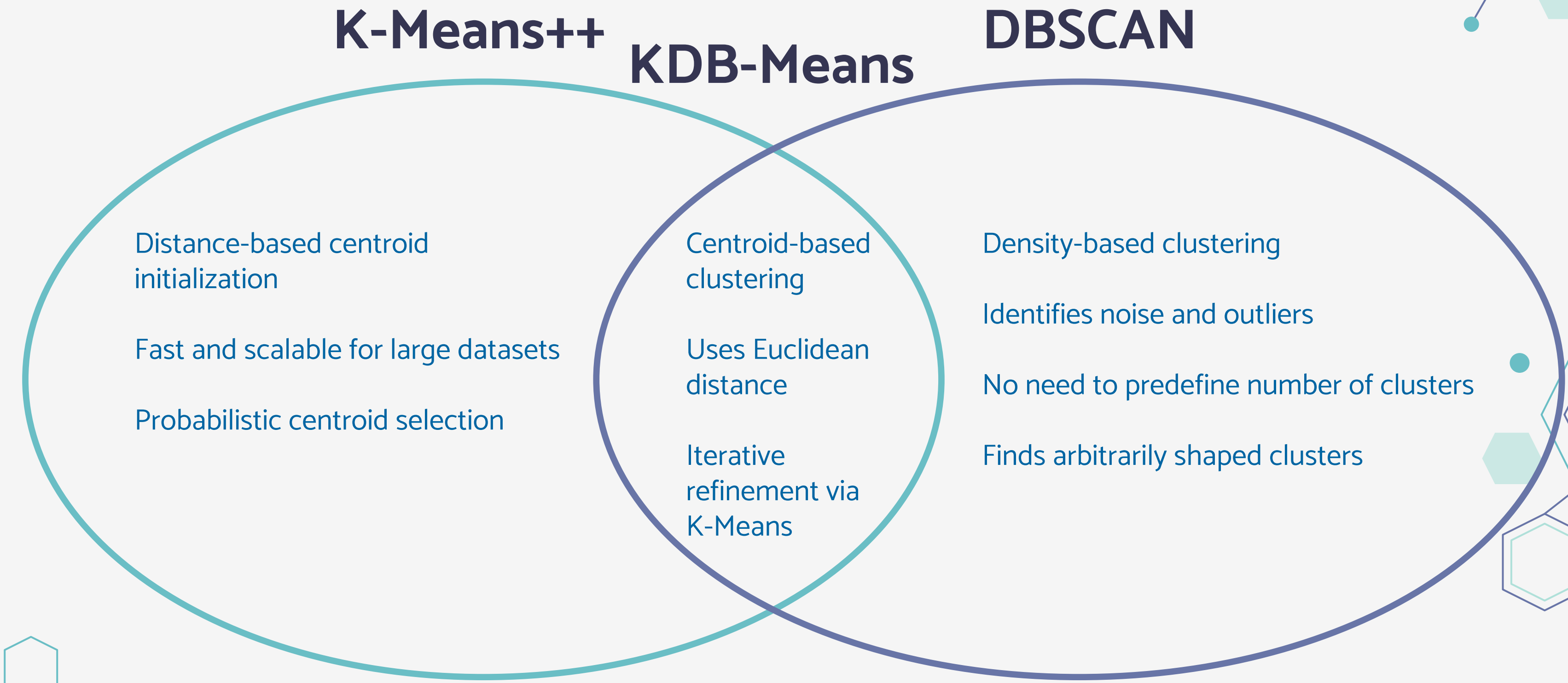
- No need to predefined number of clusters
- Can identify arbitrarily shaped clusters

Drawbacks:

- Sensitive to parameter selection
- Doesn’t create centroid-based clusters
- Less suitable for applications needing centroid refinement

KDB-Means vs. DBSCAN and K-Means++

“KDB-Means can be viewed as a hybrid that combines key ideas from K-Means++ and DBSCAN.”



03

Our Dataset

Dataset

Source: UCI Machine Learning Repository

Name: User Knowledge Modeling (UNS) [8]

Domain: Student knowledge assessment in *Electrical DC Machines*

Instances: 403 students

Features: 5 continuous attributes (values in [0,1][0,1][0,1])

Features

STG: Study time for target subject

SCG: Frequency of repetition of target material

STR: Study time for related subjects

LPR: Performance in related subject exams

PEG: Performance in elective exams

Class Label (UNS)

Ordinal categories: *Very Low, Low, Middle, High*

Used only for evaluation, not for clustering

Preprocessing

- **Combined training and testing sets**
  - Dataset is used in an **unsupervised** setting
  - All instances treated as unlabeled during clustering
- **Standardized class labels (UNS)**
  - Resolved capitalization inconsistency:
    - "very_low" and "Very Low"
  - Encoded labels as ordinal values *(0–3)*
  - **Labels were used only for evaluation**, not clustering
- **Feature normalization**
  - All continuous features normalized prior to clustering
  - Prevents features with larger variances from dominating:
    - Distance calculations
    - Density estimation
  - Essential for fair centroid initialization

Toy Datasets

What are they? Why did we use them?

In addition to our User Knowledge Modeling dataset, we created a number of toy (example) datasets for visualization purposes. Due to the higher-dimensional nature of the User Knowledge dataset, no visualizations were created.

The toy dataset visualizations provide much more understandable and communicable results.

Description of our Toy Datasets:

- Created with Scikit-Learn's make_blobs [11] function
- Each toy dataset has a random number of instances between 100 - 500 (inclusive)
- Each dataset has a random number of clusters between 3 and 10 (inclusive)

04

Results & Analysis

Results & Analysis

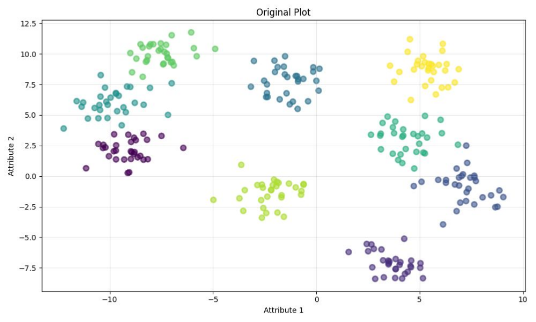


Fig. 1: Initial cluster formation in a 2D toy dataset with # nodes=271, k=9.

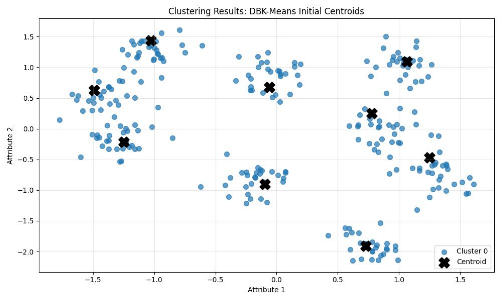


Fig. 2: DBK-Means initial centroid positions.

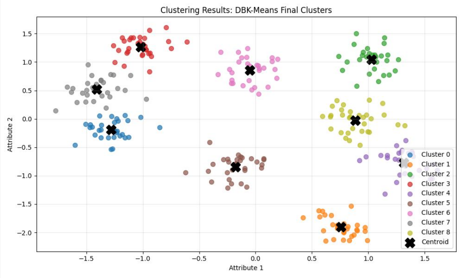


Fig. 3: DBK-Means final centroid positions with final clusters.

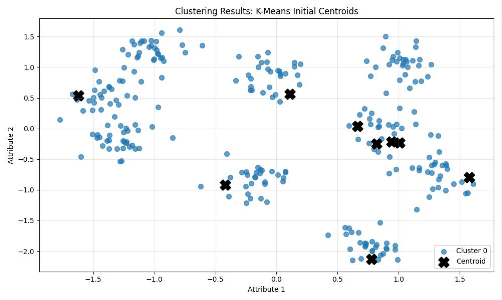


Fig. 4: K-Means initial centroid positions.

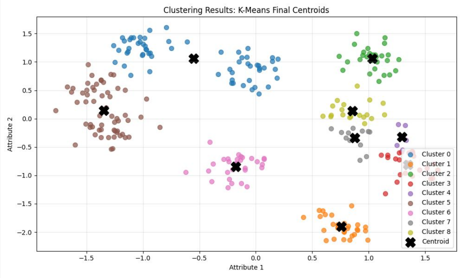


Fig. 5: K-Means final centroid positions with clusters.

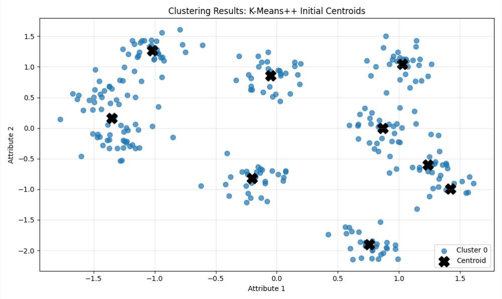


Fig. 6: K-Means++ initial centroid positions.

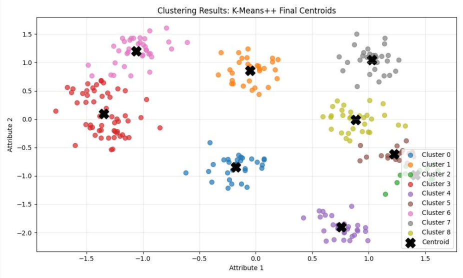


Fig. 7: K-Means++ final centroid positions with final clusters.

Results & Analysis

	KDB-Means	K-Means	K-Means++
AMI	0.2930	0.2225	0.2217
ARI	0.2108	0.1579	0.1556
Silhouette	0.1755	0.1703	0.1697
Iterations	12.0000	18.8400	16.2000

06

Conclusion + Future Work

Key Findings

- Integrating density weighting improves clustering quality
- KDB-Means produces better formed clusters
- Achieves faster convergence

Strengths & Limitations

Strengths:

- Performs well on datasets with uneven cluster density
- Outperforms K-Means and K-Means++ in density-imbalanced data

Limitations:

- Performance degrades in very high dimensions due to density estimation within an n-dimensional radius r_o

Future Work

- Explore applications with Variants of K-Means and DBSCAN-inspired hybrids
- Tune density radius parameter α for higher-dimensional datasets
- Investigate better scalable density estimation methods for large feature spaces

References

[1] Henry, David, et al. "Clustering Methods with Qualitative Data: A Mixed-Methods Approach for Prevention Research with Small Samples." *Prevention Science*, vol. 16, no. 7, 7 May 2015, pp. 1007-16, https://doi.org/10.1007/s11121-015-0561-z.

[2] Shutaywi, Meshal, and Nezamoddin N. Kachouie. "Silhouette Analysis for Performance Evaluation in Machine Learning with Applications to Clustering." *Entropy*, vol. 23, no. 6, 16 June 2021, p. 759, https://doi.org/10.3390/e23060759.

[3] Hartigan, J. A., and M. A. Wong. "Algorithm as 136: A K-Means Clustering Algorithm." Applied Statistics, vol. 28, no. 1, 1979, p. 100, https://doi.org/10.2307/2346830.

[4] Zhao, YanPing, and XiaoLai Zhou. "K-Means Clustering Algorithm and Its Improvement Research." Journal of Physics: Conference Series, vol. 1873, no. 1, 1 Apr. 2021, p. 012074, https://doi.org/10.1088/1742-6596/1873/1/012074.

[5] Arthur, David, and Sergei Vassilvitskii. “k-means++: The Advantages of Careful Seeding.” Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA ’07), Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035, https://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf

[6] Liao, Jiyong, et al. "K-NNDP: K-Means Algorithm Based on Nearest Neighbor Density Peak Optimization and Outlier Removal." Knowledge-Based Systems, vol. 294, June 2024, p. 111742, https://doi.org/10.1016/j.knosys.2024.111742.

[7] Ester, Martin, et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.” Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD ’96), AAAI Press, 1996, pp. 226–231, https://ui.adsabs.harvard.edu/abs/1996kddm.conf..226E

References

[8] "User Knowledge Modeling." UC Irvine Machine Learning Repository, 13 July 2013, archive.ics.uci.edu/dataset/257/user+knowledge+modeling.
[9] Vinh, Nguyen Xuan, Julien Epps, and James Bailey. "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance." Journal of Machine Learning Research, vol. 11, no. 95, 2010, pp. 2837–2854, http://jmlr.org/papers/v11/vinh10a.html
[10] Rousseeuw, Peter J. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." Journal of Computational and Applied Mathematics, vol. 20, Nov. 1987, pp. 53-65, https://doi.org/10.1016/0377-0427(87)90125-7.
[11] Pedregosa, Fabian, et al. "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research, vol. 12, Oct. 2011, pp. 2825-30, jmlr.org/papers/v12/pedregosa11a.html
[12] Hunter, J. D. "Matplotlib: A 2D Graphics Environment." Computing in Science & Engineering, vol. 9, no. 3, 2007, pp. 90-95, https://doi.org/10.1109/MCSE.2007.55
[13] The pandas development team. pandas-dev/pandas: Pandas. Zenodo, Feb. 2020, https://doi.org/10.5281/zenodo.3509134
[14] Virtanen, Pauli, et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." Nature Methods, vol. 17, no. 3, 2020, pp. 261–72. Nature Methods. https://doi.org/10.1038/s41592-019-0686-2
[15] Harris, C. R., K. J. Millman, S. J. van der Walt, et al. "Array programming with NumPy." Nature, vol. 585, no. 7825, 2020, pp. 357–62. Nature, https://doi.org/10.1038/s41586-020-2649-2
[16] "What Is K-Means Clustering?" Google for Developers, 2024, developers.google.com/machine-learning/clustering/kmeans/overview.

Thank you for listening!

Questions?