WIKIPEDIA

k-medoids

The ***k*-medoids** or **partitioning around medoids** (**PAM**) algorithm is a clustering algorithm reminiscent of the k-means algorithm. Both the k-means and k-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the k-means algorithm, k-medoids chooses data points as centers (medoids or exemplars) and can be used with arbitrary distances, while in k-means the centre of a cluster is not necessarily one of the input data points (it is the average between the points in the cluster). The PAM method was proposed in 1987[1] for the work with l_1 norm and other distances.

k-medoid is a classical partitioning technique of clustering, which clusters the data set of n objects into k clusters, with the number k of clusters assumed known *a priori* (which implies that the programmer must specify k before the execution of the algorithm). The "goodness" of the given value of k can be assessed with methods such as the silhouette method.

It is more robust to noise and outliers as compared to k-means because it minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances.

A medoid can be defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal, that is, it is a most centrally located point in the cluster.

Algorithms

The most common realisation of k-medoid clustering is the partitioning around medoids (PAM) algorithm. PAM uses a greedy search which may not find the optimum solution, but it is faster than exhaustive search. It works as follows:

1. Initialize: greedily select k of the n data points as the medoids to minimize the cost
2. Associate each data point to the closest medoid.
3. While the cost of the configuration decreases:
    1. For each medoid m, and for each non-medoid data point o:
        1. Consider the swap of m and o, and compute the cost change
        2. If the cost change is the current best, remember this m and o combination
    2. Perform the best swap of m_{best} and o_{best}, if it decreases the cost function. Otherwise, the algorithm terminates.

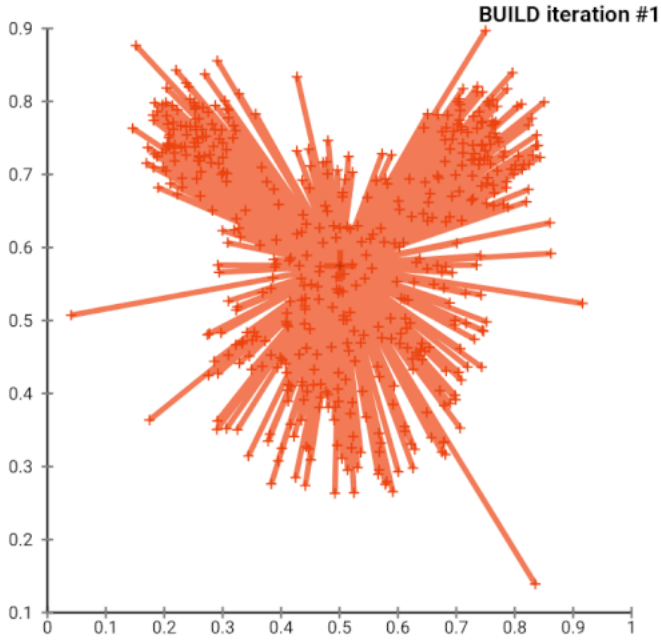
see page 4 for explanations

The runtime complexity of the original PAM algorithm per iteration of (3) is $O(k(n-k)^2)$, by only computing the change in cost. A naive implementation recomputing the entire cost function every time will be in $O(n^2k^2)$. This runtime can be further reduced to $O(n^2)$, by splitting the cost change into three parts such that computations can be shared or avoided.[2]

Algorithms other than PAM have also been suggested in the literature, including the following Voronoi iteration method:[3][4][5]

1. Select initial medoids randomly
2. Iterate while the cost decreases:
    1. In each cluster, make the point that minimizes the sum of distances within the cluster the medoid
    2. Reassign each point to the cluster defined by the closest medoid determined in the previous step.

However, *k*-means-style Voronoi iteration finds worse results, as it does not allow reassigning points to other clusters while changing means, and thus only explores a smaller search space.[2][6]



PAM choosing initial medoids, then iterating to convergence for k=3 clusters, visualized with ELKI.

The approximate algorithms CLARA and CLARANS trade optimality for runtime. CLARA applies PAM on multiple subsamples, keeping the best result. CLARANS works on the entire data set, but only explores a subset of the possible swaps of medoids and non-medoids using sampling.

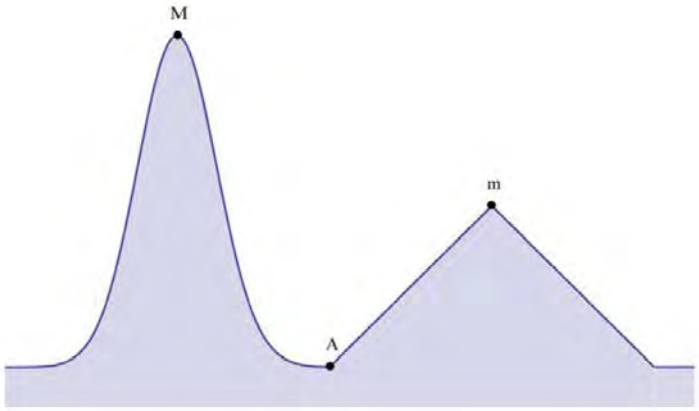
Software

- ELKI includes several *k*-medoid variants, including a Voronoi-iteration *k*-medoids, the original PAM algorithm, Reynolds' improvements, and the O(n²) FastPAM algorithm, CLARA, CLARANS, FastCLARA and FastCLARANS.
- Julia contains a *k*-medoid implementation of the k-means style algorithm (faster, but much worse result quality) in the JuliaStats/Clustering.jl (https://github.com/JuliaStats/Clustering.jl) package.
- KNIME includes a *k*-medoid implementation supporting a variety of efficient matrix distance measures, as well as a number of native (and integrated third-party) *k*-means implementations
- R contains PAM in the "cluster" package, including some of the FastPAM improvements via the pamonce=5 option.
- RapidMiner has an operator named KMedoids, but it does *not* implement the KMedoids algorithm correctly. Instead, it is a k-means variant, that substitutes the mean with the closest data point (which is not the medoid).
- MATLAB implements PAM, CLARA, and two other algorithms to solve the *k*-medoid clustering problem.

References

1. Kaufman, L. and Rousseeuw, P.J. (1987), Clustering by means of Medoids, in Statistical Data Analysis Based on the L_1–Norm and Related Methods, edited by Y. Dodge, North-Holland, 405–416.
2. Schubert, Erich; Rousseeuw, Peter J. (2019), Amato, Giuseppe; Gennaro, Claudio; Oria, Vincent; Radovanović, Miloš (eds.), "Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms", *Similarity Search and Applications*, Springer International Publishing, **11807**, pp. 171–187, arXiv:1810.05691 (https://arxiv.org/abs/1810.05691), doi:10.1007/978-3-030-32047-8_16 (https://doi.org/10.1007%2F978-3-030-32047-8_16), ISBN 9783030320461
3. Maranzana, F. E. (1963). "On the location of supply points to minimize transportation costs". *IBM Systems Journal*. **2** (2): 129–135. doi:10.1147/sj.22.0129 (https://doi.org/10.1147%2Fsj.22.0129).
4. T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning, Springer (2001), 468–469.
5. Park, Hae-Sang; Jun, Chi-Hyuck (2009). "A simple and fast algorithm for K-medoids clustering". *Expert Systems with Applications*. **36** (2): 3336–3341. doi:10.1016/j.eswa.2008.01.039 (https://doi.org/10.1016%2Fj.eswa.2008.01.039).
6. Teitz, Michael B.; Bart, Polly (1968-10-01). "Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph". *Operations Research*. **16** (5): 955–961. doi:10.1287/opre.16.5.955 (https://doi.org/10.1287%2Fopre.16.5.955). ISSN 0030-364X (https://www.worldcat.org/issn/0030-364X).

From Wikipedia, the free encyclopedia: A **greedy algorithm** is any algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage with the intent of finding a global optimum. Example below: Problem: Find the highest peak.
A greedy algorithm might be to move along the path with the highest slope. Following this greedy algorithm, you would stop at m instead of choosing the correct stopping point at M.



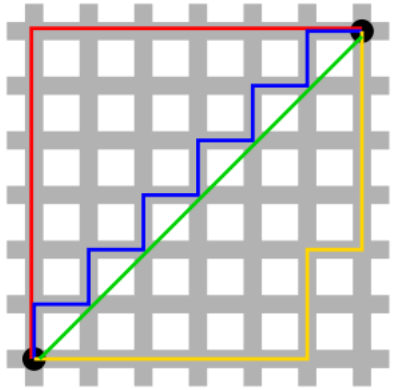
The cost parameter represents the cost of making errors. A large value severely penalizes errors and leads to a more complex classification boundary. There will be less misclassifications in the training sample, but over-fitting may result in poor predictive ability in new samples. Smaller values lead to a flatter classification boundary but may result in under-fitting. Cost is always positive. Examples: Sum(ei^2) Sum(|ei|).

WIKIPEDIA

Taxicab geometry

A **taxicab geometry** is a form of geometry in which the usual distance function or metric of Euclidean geometry is replaced by a new metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates. The **taxicab metric** is also known as **rectilinear distance**, L_1 **distance**, L^1 **distance** or ℓ_1 **norm** (see L^p space), **snake distance**, **city block distance**, **Manhattan distance** or **Manhattan length**, with corresponding variations in the name of the geometry.[1] The latter names allude to the grid layout of most streets on the island of Manhattan, which causes the shortest path a car could take between two intersections in the borough to have length equal to the intersections' distance in taxicab geometry.

The geometry has been used in regression analysis since the 18th century, and today is often referred to as LASSO. The geometric interpretation dates to non-Euclidean geometry of the 19th century and is due to Hermann Minkowski.



Taxicab geometry versus Euclidean distance: In taxicab geometry, the red, yellow, and blue paths all have the same shortest path length of 12. In Euclidean geometry, the green line has length $6\sqrt{2} \approx 8.49$ and is the unique shortest path.

Contents

If C = a diagonal matrix then Mahalanobis Distance = Euclidean Distance

Formal Definition

MAHALANOBIS DISTANCE

The Mahalanobis distance between two objects is defined (Varmuza & Filzmoser, 2016, p.46) as:

$d(\text{Mahalanobis}) = [(x_B - x_A)^T * C^{-1} * (x_B - x_A)]^{0.5}$

Where:

x_A and x_B is a pair of objects, and

C is the sample covariance matrix.

In our matrix notation
$C^{-1} = MSE(X'X)^{-1}$

Another version of the formula, which uses distances from each observation to the central mean:

$di = [(x_i - \bar{x})^t C^{-1}(x_i - \bar{x})]^{0.5}$

Where:

x_i = an object vector

$\bar{x}$ = arithmetic mean vector

Formal definition

The taxicab distance, d_1, between two vectors $\mathbf{p}, \mathbf{q}$ in an n-dimensional real vector space with fixed Cartesian coordinate system, is the sum of the lengths of the projections of the line segment between the points onto the coordinate axes. More formally,

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

where $(\mathbf{p}, \mathbf{q})$ are vectors

$$\mathbf{p} = (p_1, p_2, \ldots, p_n) \text{ and } \mathbf{q} = (q_1, q_2, \ldots, q_n)$$

For example, in the plane, the taxicab distance between (p_1, p_2) and (q_1, q_2) is $|p_1 - q_1| + |p_2 - q_2|$.

Properties

Taxicab distance depends on the rotation of the coordinate system, but does not depend on its reflection about a coordinate axis or its translation. Taxicab geometry satisfies all of Hilbert's axioms (a formalization of Euclidean geometry) except for the side-angle-side axiom, as two triangles with equally "long" two sides and an identical angle between them are typically not congruent unless the mentioned sides happen to be parallel.

Circles

A circle is a set of points with a fixed distance, called the *radius*, from a point called the *center*. In taxicab geometry, distance is determined by a different metric than in Euclidean geometry, and the shape of circles changes as well. Taxicab circles are squares with sides oriented at a 45° angle to the coordinate axes. The image to the right shows why this is true, by showing in red the set of all points with a fixed distance from a center, shown in blue. As the size of the city blocks diminishes, the points become more numerous and become a rotated square in a continuous taxicab geometry. While each side would have length $\sqrt{2}r$ using a Euclidean metric, where r is the circle's radius, its length in taxicab geometry is $2r$. Thus, a circle's circumference is $8r$. Thus, the value of a geometric analog to π is 4 in this geometry. The formula for the unit circle in taxicab geometry is $|x| + |y| = 1$ in Cartesian coordinates and

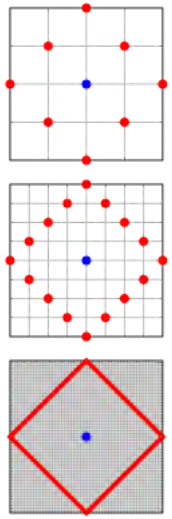
$$r = \frac{1}{|\sin\theta| + |\cos\theta|}$$

in polar coordinates.

A circle of radius 1 (using this distance) is the von Neumann neighborhood of its center.

A circle of radius r for the Chebyshev distance (L_∞ metric) on a plane is also a square with side length $2r$ parallel to the coordinate axes, so planar Chebyshev distance can be viewed as equivalent by rotation and scaling to planar taxicab distance. However, this equivalence between L_1 and L_∞ metrics does not generalize to higher dimensions.

Whenever each pair in a collection of these circles has a nonempty intersection, there exists an intersection point for the whole collection; therefore, the Manhattan distance forms an injective metric space.



Circles in discrete and continuous taxicab geometry

Applications

Measures of distances in chess

In chess, the distance between squares on the chessboard for rooks is measured in taxicab distance; kings and queens use Chebyshev distance, and bishops use the taxicab distance (between squares of the same color) on the chessboard rotated 45 degrees, i.e., with its diagonals as coordinate axes. To reach from one square to another, only kings require the number of moves equal to their respective distance; rooks, queens and bishops require one or two moves (on an empty board, and assuming that the move is possible at all in the bishop's case).

Compressed sensing

In solving an underdetermined system of linear equations, the regularization term for the parameter vector is expressed in terms of the ℓ_1-norm (taxicab geometry) of the vector.[2] This approach appears in the signal recovery framework called compressed sensing.

Differences of frequency distributions

Taxicab geometry can be used to assess the differences in discrete frequency distributions. For example, in RNA splicing positional distributions of hexamers, which plot the probability of each hexamer appearing at each given nucleotide near a splice site, can be compared with L1-distance. Each position distribution can be represented as a vector where each entry represents the likelihood of the hexamer starting at a certain nucleotide. A large L1-distance between the two vectors indicates a significant difference in the nature of the distributions while a small distance denotes similarly shaped distributions. This is equivalent to measuring the area between the two distribution curves because the area of each segment is the absolute difference between the two curves' likelihoods at that point. When summed together for all segments, it provides the same measure as L1-distance.[3]

History

The L^1 metric was used in regression analysis in 1757 by Roger Joseph Boscovich.[4] The geometric interpretation dates to the late 19th century and the development of non-Euclidean geometries, notably by Hermann Minkowski and his Minkowski inequality, of which this geometry is a special case, particularly used in the geometry of numbers, (Minkowski 1910) . The formalization of L^p spaces is credited to (Riesz 1910) .

See also

- Normed vector space
- Metric
- Orthogonal convex hull
- Hamming distance
- Fifteen puzzle
- Random walk
- Manhattan wiring

Notes

1. Black, Paul E. "Manhattan distance" (https://xlinux.nist.gov/dads/HTML/manhattanDistance.html). *Dictionary of Algorithms and Data Structures*. Retrieved October 6, 2019.
2. Donoho, David L. (March 23, 2006). "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution". *Communications on Pure and Applied Mathematics*. **59** (6): 797–829. doi:10.1002/cpa.20132 (https://doi.org/10.1002%2Fcpa.20132).
3. Lim, Kian Huat; Ferraris, Luciana; Filloux, Madeleine E.; Raphael, Benjamin J.; Fairbrother, William G. (July 5, 2011). "Using positional distribution to identify splicing elements and predict pre-mRNA processing defects in human genes" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3131313). *Proceedings of the National Academy of Sciences of the United States of America*. **108** (27): 11093–11098. Bibcode:2011PNAS..10811093H (https://ui.adsabs.harvard.edu/abs/2011PNAS..10811093H). doi:10.1073/pnas.1101135108 (https://doi.org/10.1073%2Fpnas.1101135108). PMC 3131313 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3131313). PMID 21685335 (https://pubmed.ncbi.nlm.nih.gov/21685335).
4. Stigler, Stephen M. (1986). *The History of Statistics: The Measurement of Uncertainty before 1900* (https://archive.org/details/historyofstatist00stig). Harvard University Press. ISBN 9780674403406. Retrieved October 6, 2019.

References

- Krause, Eugene F. (1987). *Taxicab Geometry* (https://archive.org/details/taxicabgeometrya0000krau). Dover. ISBN 978-0-486-25202-5.
- Minkowski, Hermann (1910). *Geometrie der Zahlen* (https://archive.org/details/geometriederzahl00minkrich) (in German). Leipzig and Berlin: R. G. Teubner. JFM 41.0239.03 (https://zbmath.org/?format=complete&q=an:41.0239.03). MR 0249269 (https://www.ams.org/mathscinet-getitem?mr=0249269). Retrieved October 6, 2019.
- Riesz, Frigyes (1910). "Untersuchungen über Systeme integrierbarer Funktionen". *Mathematische Annalen* (in German). **69** (4): 449–497. doi:10.1007/BF01457637 (https://doi.org/10.1007%2FBF01457637). hdl:10338.dmlcz/128558 (https://hdl.handle.net/10338.dmlcz%2F128558).

External links

- Weisstein, Eric W. "Taxicab Metric" (http://mathworld.wolfram.com/TaxicabMetric.html). *MathWorld*.
- Malkevitch, Joe (October 1, 2007). "Taxi!" (http://www.ams.org/publicoutreach/feature-column/fcarc-taxi). *American Mathematical Society*. Retrieved October 6, 2019.