See `squareform` for information on how to calculate the index of this entry or to convert the condensed distance matrix to a redundant square matrix.

The following are common calling conventions.

1. `Y = pdist(X, 'euclidean')`

   Computes the distance between m points using Euclidean distance (2-norm) as the distance metric between the points. The points are arranged as m n-dimensional row vectors in the matrix X.

2. `Y = pdist(X, 'minkowski', p=2.)`

   Computes the distances using the Minkowski distance $\|u - v\|_p$ ($p$-norm) where $p > 0$ (note that this is only a quasi-metric if $0 < p < 1$).

3. `Y = pdist(X, 'cityblock')`

   Computes the city block or Manhattan distance between the points.

4. `Y = pdist(X, 'seuclidean', V=None)`

   Computes the standardized Euclidean distance. The standardized Euclidean distance between two n-vectors `u` and `v` is

   $$\sqrt{\sum (u_i - v_i)^2 / V[x_i]}$$

   V is the variance vector; V[i] is the variance computed over all the i'th components of the points. If not passed, it is automatically computed.

5. `Y = pdist(X, 'sqeuclidean')`

   Computes the squared Euclidean distance $\|u - v\|_2^2$ between the vectors.

6. `Y = pdist(X, 'cosine')`

   Computes the cosine distance between vectors u and v,

   $$1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2}$$

   where $\| * \|_2$ is the 2-norm of its argument `*`, and $u \cdot v$ is the dot product of `u` and `v`.

7. `Y = pdist(X, 'correlation')`

   Computes the correlation distance between vectors u and v. This is

   $$1 - \frac{(u - \bar{u}) \cdot (v - \bar{v})}{\|(u - \bar{u})\|_2 \|(v - \bar{v})\|_2}$$

   where $\bar{v}$ is the mean of the elements of vector v, and $x \cdot y$ is the dot product of $x$ and $y$.

8. `Y = pdist(X, 'hamming')`

   Computes the normalized Hamming distance, or the proportion of those vector elements between two n-vectors `u` and `v` which disagree. To save memory, the matrix `X` can be of type boolean.

9. `Y = pdist(X, 'jaccard')`

   Computes the Jaccard distance between the points. Given two vectors, `u` and `v`, the Jaccard distance is the proportion of those elements `u[i]` and `v[i]` that disagree.

10. `Y = pdist(X, 'jensenshannon')`

    Computes the Jensen-Shannon distance between two probability arrays. Given two probability vectors, $p$ and $q$, the Jensen-Shannon distance is

    $$\sqrt{\frac{D(p \parallel m) + D(q \parallel m)}{2}}$$

    where $m$ is the pointwise mean of $p$ and $q$ and $D$ is the Kullback-Leibler divergence.

11. `Y = pdist(X, 'chebyshev')`

    Computes the Chebyshev distance between the points. The Chebyshev distance between two n-vectors `u` and `v` is the maximum norm-1 distance between their respective elements. More precisely, the distance is given by

    $$d(u, v) = \max_i |u_i - v_i|$$

12. `Y = pdist(X, 'canberra')`

    Computes the Canberra distance between the points. The Canberra distance between two points `u` and `v` is

    $$d(u, v) = \sum_i \frac{|u_i - v_i|}{|u_i| + |v_i|}$$

13. `Y = pdist(X, 'braycurtis')`

    Computes the Bray-Curtis distance between the points. The Bray-Curtis distance between two points `u` and `v` is

    $$d(u, v) = \frac{\sum_i |u_i - v_i|}{\sum_i |u_i + v_i|}$$

14. `Y = pdist(X, 'mahalanobis', VI=None)`

    Computes the Mahalanobis distance between the points. The Mahalanobis distance between two points `u` and `v` is $\sqrt{(u - v)(1/V)(u - v)^T}$ where $(1/V)$ (the `VI` variable) is the inverse covariance. If `VI` is not None, `VI` will be used as the inverse covariance matrix.

15. `Y = pdist(X, 'yule')`

    Computes the Yule distance between each pair of boolean vectors. (see yule function documentation)

16. `Y = pdist(X, 'matching')`

    Synonym for 'hamming'.

17. `Y = pdist(X, 'dice')`

    Computes the Dice distance between each pair of boolean vectors. (see dice function documentation)

18. `Y = pdist(X, 'kulczynski1')`

    Computes the kulczynski1 distance between each pair of boolean vectors. (see kulczynski1 function documentation)

19. `Y = pdist(X, 'rogerstanimoto')`

    Computes the Rogers-Tanimoto distance between each pair of boolean vectors. (see rogerstanimoto function documentation)

20. `Y = pdist(X, 'russellrao')`

    Computes the Russell-Rao distance between each pair of boolean vectors. (see russellrao function documentation)

21. `Y = pdist(X, 'sokalmichener')`

    Computes the Sokal-Michener distance between each pair of boolean vectors. (see sokalmichener function documentation)

22. `Y = pdist(X, 'sokalsneath')`

    Computes the Sokal-Sneath distance between each pair of boolean vectors. (see sokalsneath function documentation)

23. `Y = pdist(X, 'kulczynski1')`

    Computes the Kulczynski 1 distance between each pair of boolean vectors. (see kulczynski1 function documentation)