

The knnFinder Package

February 16, 2008

Title Fast Near Neighbour Search

Version 1.0

Author Samuel E. Kemp

Description Finds the p number of near neighbours for every point in a given dataset in $O(M \log M)$ time.

Maintainer Samuel E. Kemp <sekemp@glam.ac.uk>

License GPL version 2

R topics documented:

nn	1
Index	3

nn	<i>Near Neighbour Search</i>
----	------------------------------

Description

Uses a kd-tree to find the p number of near neighbours for each point in an input/output dataset. The advantage of the kd-tree is that it runs in $O(M \log M)$ time.

Usage

```
nn(data, mask=seq(from=1, to=1, length=(length(data[,])-1)), p=10)
```

Arguments

data	An input-output dataset. THE OUTPUT MUST BE IN THE RIGHT MOST COLUMN OF A DATA FRAME OR MATRIX.
mask	A vector of 1's and 0's representing input inclusion/exclusion. The default mask is all 1's (i.e. include all inputs in the test).
p	The maximum number of near neighbours to compute. The default value is set to 10.

Details

The algorithm itself works by calculating the nearest neighbour distances in input space. This calculation is achieved in $O(M \log M)$ time, where M is the number of data points using Bentley's kd-tree. The `knnFinder` package utilizes the Approximate Near Neighbor (ANN) C++ library, which can give the exact near neighbours or (as the name suggests) approximate near neighbours to within a specified error bound. We use EXACT near neighbours in this package. For more information on the ANN library please visit <http://www.cs.umd.edu/~mount/ANN/>.

Value

`nn.idx` A MxP data.frame returning the near neighbour indexes.
`nn.dists` A MxP data.frame returning the near neighbour Euclidean distances.

Author(s)

Samuel E. Kemp. To report any bugs or suggestions please email: [⟨sekemp@glam.ac.uk⟩](mailto:sekemp@glam.ac.uk)

References

- Bentley J. L. (1975), Multidimensional binary search trees used for associative search. *Communication ACM*, 18:309-517.
- Arya S. and Mount D. M. (1993), Approximate nearest neighbor searching, *Proc. 4th Ann. ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, 271-280.
- Arya S., Mount D. M., Netanyahu N. S., Silverman R. and Wu A. Y (1998), An optimal algorithm for approximate nearest neighbor searching, *Journal of the ACM*, 45, 891-923.

Examples

```
# A noisy sine wave example
x1 <- runif(100, 0, 2*pi)
x2 <- runif(100, 0, 3)
e <- rnorm(100, sd=sqrt(0.075))
y <- sin(x1) + 2*cos(x2) + e
DATA <- data.frame(x1, x2, y)
nearest <- nn(DATA)
```

Index

*Topic **nonparametric**
nn, 1

nn, 1