Cover tree

From Wikipedia, the free encyclopedia

Jump to: navigation, search

The **cover tree** is a type of data structure in computer science that is specifically designed to facilitate the speed-up of a nearest neighbor search. It is a refinement of the Navigating Net data structure, and related to a variety of other data structures developed for indexing intrinsically low-dimensional data.[1]

The tree can be thought of as a hierarchy of levels with the top level containing the root point and the bottom level containing every point in the metric space. Each level *C* is associated with an integer value *i* that decrements by one as the tree is descended. Each level *C* in the cover tree has three important properties:

- **Nesting:**
- **Covering:** For every point , there exists a point such that the distance from to is less than or equal to and exactly one such is a parent of .
- **Separation:** For all points , the distance from to is greater than .

Contents

Complexity[edit]

Find[edit]

Like other metric trees the cover tree allows for nearest neighbor searches in where is a constant associated with the dimensionality of the dataset and n is the cardinality. To compare, a basic linear search requires , which is a much worse dependence on . However, in high-dimensional metric spaces the constant is non-trivial, which means it cannot be ignored in complexity analysis. Unlike other metric trees, the cover tree has a theoretical bound on its constant that is based on the dataset's expansion constant or doubling constant (in the case of approximate NN retrieval). The bound on search time is where is the expansion constant of the dataset.

Insert[edit]

Although cover trees provide faster searches than the naive approach, this advantage must be weighed with the additional cost of maintaining the data structure. In a naive approach adding a new point to the dataset is trivial because order does not need to be preserved, but in a cover tree it can take time. However, this is an upper-bound, and some techniques have been implemented that seem to improve the performance in practice.[2]

Space[edit]

The cover tree uses implicit representation to keep track of repeated points. Thus, it only requires O(n) space.

See also[edit]

- Nearest neighbor search
- kd-tree

References[edit]

Notes

1.