Multimedia oral examination - Nearest Neighbor Search

1.Whis is Nearest Neighbor Search problem?

Nearest neighbor search (NNS) is the problem of finding the closest point in a given set to a given point.In k-NN search, we need to find the top-k closest points.In range search, we find nearest neighbors within a radius.

2.The challenge of Nearest Neighbour Search?

NNS is a challenging problem because the complexity grows exponentially with the number of dimensions, but in most applications, users require instant response. The complexity upper bound is O(D·N),we could expect O(D1/c·logN).

3.Distance measurement

(1)What is metric,metric space and their properties? A metric space is a set together with a metric on the set. The metric is a function that defines a concept of distance between two points. The metric should satisfy a few simple properties,which are non-negative,symmetric,triangle inequality.

(2)What is the norm and their properties? A norm is a function that assigns a positive size to each vector in a vector space. For example,L-p norm is defined as the p-th root of of the sum of the p-th power of the absolute values of each element.

Norm need to satisfy a few properties, which are scale invariance,triangle inequality and only the zero vector has zero length.

(3)Some distance functinon (3-1)Euclidean distance is the "ordinary" straight-line distance between two points in Euclidean space,which is the square root of the sum of squares.The associated norm is called the Euclidean norm or l2-norm. It is scale and translation invariant.

L1-norm is the sum of the absolute values of each element **(3-2)Cosine distance** measures the cosine of the angle between two vectors. **Relationship between Euclidean distance and Cosine distance**:If x and y are l2 normalized in advance, Cosine distance is equivalent to Euclidean distance. **(3-3)Mahalanobis distance** When the distribution of points of the center is non-spherical, for instance ellipsoidal, the distance will depend on the direction. Points in directions along short axis will be closer.Similarily further away from each other when the axis is long.

Mahalanobis distance estimate the ellipsoid by covariance matrix and the distance will be divided by the width of the ellipsoid in the direction of two point.

The Mahalanobis distance is thus unitless and scale-invariant, and takes into account the correlations of the data set. **Relationship between Euclidean distance and Mahalanobis distance** If each of these axes has unit variance, the covariance matrix is the identity matrix, the Mahalanobis distance will reduce to the Euclidean distance.

Doing PCA before applying l2 may achieve similar effect.It is used not as frequently as l2. **(3-4)Hamming Distance** measures how many substitutions it takes from one string change to another,used for binary numbers and strings.It is very efficient. **(3-5)Distance space** There are some other distance space expect Euclidean,such as Riemann and Lobachevsky space.There are no effective distance measure for these non-Euclidean spaces and no universal distance measure.

4.B-Tree

B-Tree is best solution for search for one dimensional data. Its leaf node keeps all the data items, non-leaf nodes are used for indexing.The time complexity is log(N),the space complexity is O(2^n).

5.KD-Tree

(5-1)What is KD-Tree? K-D Tree means Tree for K-dimensional data points.It is a binary tree.Every leaf node is a k-dimensional point. Every non-leaf node is a splitting hyperplane perpendicular to one of the k-dimensions that divides the space into two parts. Points to the left of this hyperplane are represented by the left subtree of that node and points to the right are represented by the right subtree.

(5-2)How to build a KD-Tree(construction procedure)? KD-Tree is build by recursively choosing one coordinate as basis to split all rest points into two parts until there is no two points in the same node.

A good spliting method should guarantee that the tree is balanced.

(5-3)How to do NNS base on a K-D Tree Searching for a nearest neighbour starts with the root node then moves down the tree left or right depending on whether the point is lesser than or greater than the current node in the split dimension.

Once reaches a node, it checks if the distance is closer and update the "current best".Keeps moving downand looking for closer points until search is complete.

(5-4)Time and space complexity The space complexity is O(N) The time complexity for online search is O(logN) if the tree is balanced, but O(N) in the worst case.

(5-5)Pros and Cons Pros: The time complexity is O(logN) if the tree is balanced and it supports range search and top-k search.

Cons: First,The retrieval cannot be as efficient as it is supposed to and may nearly traverse the whole tree in the worst case.And the performance will rapidly degrade in high dimension.

Second,the reference data is partitioned according to axis, while NN is measured by l1 or l2 norms.So NN is not necessarily located in the same branch.

6.FLANN

(6-1)What is FLANN? FLANN means Fast Library for Approximate Nearest Neighbors.It applies priority search on hierarchical k-means trees. **(6-2)How to build a hierarchical k-means tree?** The hierarchical k-means tree is constructed by using k-means splitting points, and do it recursively for each level until the number of points in a region is smaller than K. **(6-3)How to do NNS base on FLANN?** Query is initially compared to the first level of the hierarchical k-means tree,and build a priority queue of top-n closest candidates.

Next, extract closest point from the priority queue then restarts the tree traversal from that branch and updates to the priority queue until reaches leaf nodes.

(6-4)Time and space complexity

(6-5)Pros and Cons FLANN is fast but not precise.It only returns approximate NNs and lot of memories are required to maintain this indexing tree.

It is OK if we do not require precise NN search

7.LSH - Locality Sensitive Hashing

(7-1)What is Locality Sensitive Hashing?Why it works?

Similar items map to the same “buckets” with high probability after LSH hashes.

(7-2)How to build hash functions and hash codes? Hash functions are defined by drawing a random vector for L times. Hash code is produced by caculate hash code for each hash function and concatenate all generated hash code into a binary code. **(7-3)How to do NNS base on LSH?** Encode and compare query to all hash codes in reference set,keep the same one or similar ones as candidate.

Then compute real distance between query and these candidates,return the top-k ranked candidates. **(7-4)Time and space complexity** the width parameter k and the number of hash tables L. preprocessing time: O(nLkt) space:O(nL) query time: O(L(kt+dnP_2^k))

(7-5)trade-off between general and effective There are two types of errors,mismatch and false match.

Mismatch can be alleviated by multiple-probe LSH,which means the candidate set keeps not only points that share the same hash codes with the query,but also points that have similar hash codes.

False match can be alleviated by using more Hash functions.Actually it is a trade-off.Because the more the number of hash functions, the lower the recall rate and the higher the accuracy,the faster the speed.

(7-6)Pros and Cons LSH reduces the dimensionality of high-dimensional data,but do not support exact top k search. It is OK if you do not require precise NN search. Actually,In practices,this method has the worst accuracy and recall rate, and the efficiency is not the highest.

[ppt里的图？]

8.Product Quantizer（积量化）

(8-1)What is Product Quantizer? Why it works? Product Quantizer takes fewer prototypes (or “codes”) to represent the vectors and performs the query on compressed data. Product Quantizer is between scalar quantization and vector quantization. Vector quantization representes data points by the index of their closest centroid, it's a lossy data compression. For example,a 2-dimensional point will be quantized as a Voronoi cell ID which is 1D. scalar quantization represent 1D continuous signal with few digital numbers.For example,an RGB value from 0 to 255 can be digitize using 8 bits.

(8-2)How to do quantization? PQ works by dividing vectors into m subvectors, and running k-means on each of them to learn m vocabularies to represent all of the vectors. Each vector is quantized by finding the closest centroids for each subvector and concanate them.

(8-3)How to do NNS base on PQ.(how measure distance after quantization)? We measure the distance between the quantized query and quantized points in a precomputed set.

If we user symmetric product quantizer,distance between PQ vectors is the sum of the distances between the m centroids they are clustered.We can build a distance lookup table for each PQ vocabulary to simplify the calculation of distances by looking-ups and summations,which is very efficient.

Asymmetric product quantizer is a more precise but slower way which encode the reference side only,then the distance of each subvector-pair is the distance between the original subvector of query and the centroid vector of the reference subvector.

(8-4)Time and space complexity (8-5)pros and cons pros: The original data do not have to be loaded into memory For symmetric product quantizer,we can build a distance lookup table for each PQ vocabulary to simplify the calculation of distances by looking-ups and summations,which is very efficient.

cons: But it only return approximate results.And although PQ use compressed data representation to alleviate the burden of big data,but if the vocabulary is very large ,look-up is still time consuming.

NN-Descent

(9-1)What is NN-Descent? How to do NNS base on NND? NN-Descent is based on a nearest neighbor graph build offline. For a given query, NND performs hill-climbing starting from a random node and maintains a candidate list of the current top-k nearest visited points.Then repeat scaning and updating the candidate list until it cannot be updated during the entire scan.

(9-2)Why it works? It can be explained by small-world network.Most nodes are not neighbors of one another, but can be reached by a small number of hops.

In high-dimensional space, the distribution of points is not uniform. What we really care about is its intrinsic dimension, which is a sub-space.

Graph-based approch such as NND can use this informtion and climb in the sub-space ,which is much easier than exploring the whole.

(9-3)Time and space complexity The computational complexity of the naive construction of NNgraph is O(dn^2), the computational complexity is **(9-4)pros and cons**

10.summary

Based on the size of dimensions(high/low) and dataset (sparse/dense), the problem can be partitioned into 4 sub-problems. (1)For low and sparse data,such as Spatial-temporal data ,we can use KD-Tree and inverted file. (2)For low and dense data,such as X-Y, RGB and HSV,we can use KD-Tree. (3)For high and sparse data,such as text document and Bag-of-Visual Word ,we can inverted file. (4)For high and dense data,such as SIFT and VLAD ,we can use PQ and NND.