

Scalable Data Science
Prof. Anirban Dasgupta
Department of Computer Science and Engineering
Indian Institute of Technology, Gandhinagar

Lecture - 12
Locality Sensitive Hashing

Hello welcome to today's lecture of Scalable Data Science. Today we are continuing on a theme of finding nearest neighbours. Today's lecture is going to be on Locality Sensitive Hashing.

(Refer Slide Time: 00:28)

Finding Near Neighbors

Given a set of data points and a query

Can we find what is the nearest datapoint to the query?

- K-nearest neighbors
- $d(p, \text{query}) < r$

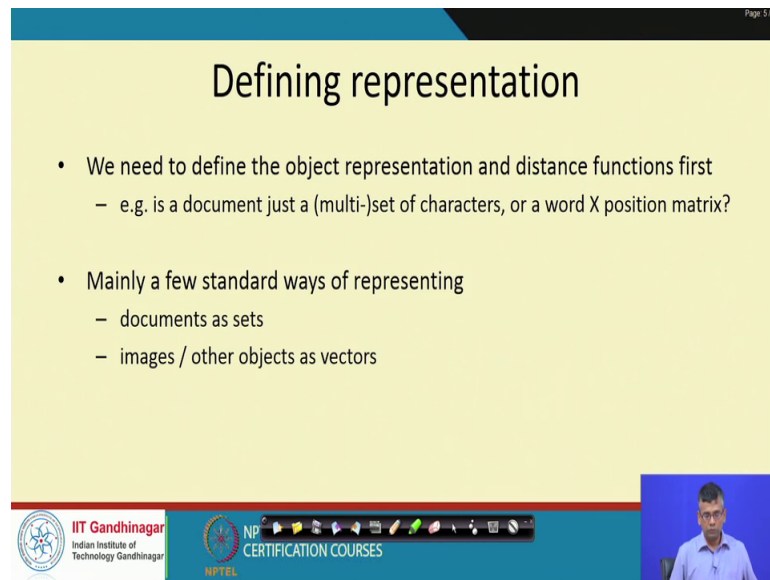
IIT Gandhinagar
Indian Institute of Technology Gandhinagar

NP
CERTIFICATION COURSES

NPTEL

So, here was a question of finding near neighbours that we are looking at. We are given by a we were given a set of data points right and we were feed if pre process this data points as you wanted. Given at query time we would be given a query one query point and we would be asked to return the nearest data point to the query, well the nearest or maybe some version of it maybe the k nearest or maybe we are given the query point along with the target radius, and we are being ask to return all the points in the data set, that live within this target radius target distance to the query. These are all variance of the question that are important in one setting on the other.

(Refer Slide Time: 01:14)



Page 5/13

Defining representation

- We need to define the object representation and distance functions first
 - e.g. is a document just a (multi-)set of characters, or a word X position matrix?
- Mainly a few standard ways of representing
 - documents as sets
 - images / other objects as vectors

IIT Gandhinagar
Indian Institute of
Technology Gandhinagar

NP
CERTIFICATION COURSES
NPTEL

So, before we delft moving to this, let us step back and talk a little bit about how we are representing an object, what is the distance and so on right. Because when we dealt with k d trees, we kind of implicitly assumed that maybe this a nice distance function like l 2 that is available, and because typically k d tree is work with some with nueclyden distance or versions of that right and a it means a metric space in which instead of putting in points. The locality sensitive hash functions also have some restrictions and we will see, we will sort of see what kind of what kind of distance functions are actually useful.

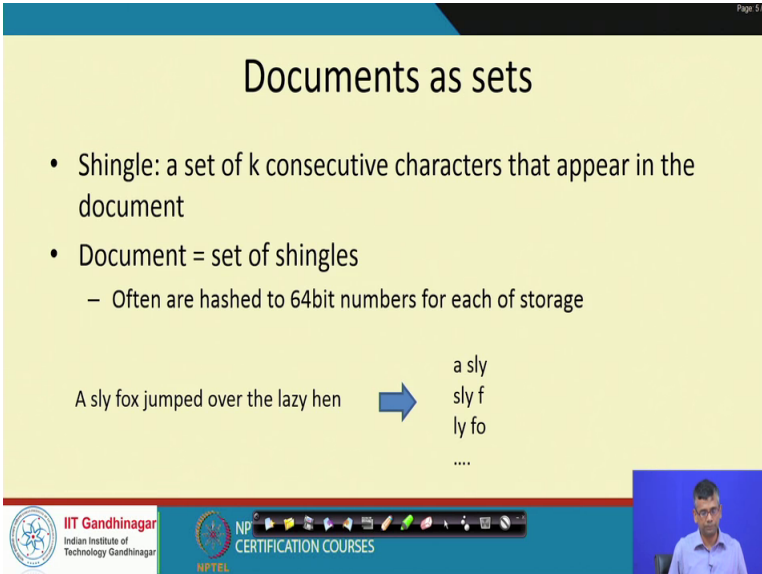
So, before we talk about distance functions. So, we need to talk about that what is a point what is a point right. I mean there is a obvious representation of a of a point as a vector right I mean we represent images as a vectors; you can represent documents as vectors, but there are also other representations. You could also think about I mean a data point as really a set or a multi set right; for instance given a particular document it a it so, a priory obvious to you if we have not thought about it that, what is the best representation for it?

Should it be a word by position matrix right when the when one axis is was the rows or was the columns are may the other columns are I mean the positions and then and then you create this matrix, is at a useful representation. Should I represent it as a multi set of characters right and it turns out that none of these represent none of these representations are actually good representation right.

It turns out that the nice representations for a document is in terms of that are few nice representations, one nice representation is as a set of words in the document right I mean what is the word? Well it is easy to defined what about this if you writing in English, but if you writing in a language in which there is in which is not using spaces separator is now natural separator, it is not easy to define what about is right and then we have to go to some other things that we see today. There are I mean and, but once we do this, we will represent document as a set right.

And these will be basically the two standard representations that we use, either a set or a vector right. An either the vector will be normalized or it will be a it would not be, and the and the and the distance function that we used might different these two cases might depend on the whether the vector is normalize or not.

(Refer Slide Time: 03:51)



The slide is titled "Documents as sets" and contains the following content:

- Shingle: a set of k consecutive characters that appear in the document
- Document = set of shingles
 - Often are hashed to 64bit numbers for each of storage

An example is provided: "A sly fox jumped over the lazy hen" with a blue arrow pointing to a list of shingles: "a sly", "sly f", "ly fo", and "....".

The slide footer includes logos for IIT Gandhinagar, NPTEL, and NP Certification Courses, along with a small video inset of a speaker.

So, let us go to a representing documents as sets. So, as I said that I mean in most cases I will, in a lot of cases you might be in a setting, when you contrary rely on a natural separator for the documents right. And the one particularly useful technique that is used in that is used in let us say analysing I mean any documents or in the web companies analysing external pages is in using shingles.

So, what is a shingle? Shingle is nothing, but a set of k consecutive characters that have appear in the document right. So, it does not makes sense to take the each individual character because then every document would basically have the same representation

right. Is the frequency of the characters is more or less are same in any large enough document. So, what you do is it a k consecutive characters right. And then that substring is a shingle. So, rather than store the substring the list of the substring on the multi set of the substring. So, what we do is that, we hash each of these substrings into a 64 bit number right using something like MD 5 or SHA in using something in MD 5 mer three any fast hash function would do. And then and then you store the multi set or the set of these of the shingles.

For instance here is a particular sentence right as sly fox jumped over the lazy hen. And even create as shingle by sliding a window, suppose yeah when you create a shingles of size 5 right. So, you want to size I mean k equal to 5. So, slide a window of size 5 over the document right which is the only one sentence. So, first shingle is a s l y, maybe the next shingle is s l y f maybe the next shingle is l y f o space f o and you keep on creating the shingles, and then you hash them and then you should the keep rest of the shingles. The idea is that that keeping k consecutive characters preserves some amount of semantic semantics of the document.

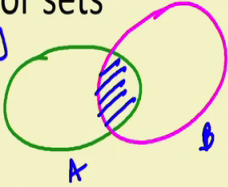
And two documents that are similar according to some similarity measure that you can define on these two sets also have been shown to be semantically similar in terms of human judgement right. So, whatever we try to we trying to capture the semantic similarity right in terms of this some mathematical notion of similarity, that will define on the sets of shingles. And shingles have turned out to be particularly useful representation in terms of doing this.

(Refer Slide Time: 06:23)

Page 5/13

Distance function for sets

- Jaccard similarity $JS(A, B) = \frac{|A \cap B|}{|A \cup B|}$ [0,1]
- Distance $JD(A, B) = 1 - JS(A, B)$



The slide features a Venn diagram with two overlapping circles, A (green) and B (pink). The intersection of A and B is shaded with blue diagonal lines. The labels 'A' and 'B' are written in blue below the circles. The slide also contains the title 'Distance function for sets' and two bullet points: 'Jaccard similarity JS(A, B) = |A ∩ B| / |A ∪ B| [0,1]' and 'Distance JD(A, B) = 1 - JS(A, B)'. The bottom of the slide shows the IIT Gandhinagar logo and NP Certification Courses branding.

So, now we have represented the document as a set. So, what do we do? So, have a two natural. So, is a natural similarity function over sets right its call a Jaccard similarity you must have heard about this what is this? He just says that if we have if we have two sets A and B, if we have two sets A and B and you want to and you want to calculate the and you want to calculate the Jaccard similarity of these two sets what you do is that, you take the you take the region you take the number of elements in A intersection B and then you divide by the cardinality of A union B right. It is easy to see the these the other this always gives you a number between 0 and 1 right. So, this is a notion of similarity.

So, Jaccard similarity of A with A could be 1 right and if the two sets are completely disjoint, Jaccard similarity would be 0. So, corresponding to this we can define a notion of distance functions, Jaccard distance this is not hash commonly referred to, but is basically the same thing and Jaccard distance is one minus Jaccard similarity right. So, again I mean if you are talking about the metric, Jaccard distance obeys all the properties of a metric that is Jaccard distance of A with itself would be 0; Jaccard distance of I mean of A with B is the same as a Jaccard distance of B with A and more importantly he doubles triangle in equality. That is not so, obvious to see actually the Jaccard distance, but it does. So, and that is actually going to turned out to be fairly important. Suppose we have vectors right so, think about images.

(Refer Slide Time: 08:07)

Page 6/14

Vectors

- Images
 - Vectors over 64x64 or 128x128
- Documents
 - Sets → vectors, possibly with TF-IDF or other weighting
- Distance functions
 - $l_2(x, y) = (\sum_i (x_i - y_i)^2)^{1/2}$ ✓ l_1 l_p
 - ...
 - angle between vectors

IIT Gandhinagar
Indian Institute of
Technology Gandhinagar

NP
CERTIFICATION COURSES

So, images are often represented as vectors. So, fix a 64 by 64 image you can represent one coordinate of the vector for each of these positions. So, the size of the vector is 64 cross 64 right and then maybe you have the gray scale value of the image at that particular pixel as the as a value in the vector right. Documents are also of an represent the as sets, he could convert the I mean documents also of an represented as vectors, he could convert the set into a 0 one vector the set that you have constructed.

Or he could create a vector with some t f i d f or some other weighting right basically the position there is a word there is a position for each shingle or a position for each I mean k grammar a position for each word, and then and then you sort of you put in some kind of weighting for that shingle or for that for that k grammar for that word in that particular position right. And once you create, once you represent as vectors that are of course, natural distance functions is l_2 distance function, there is the some l_1 distance function other l_p distance functions as and if its if the vectors and normalised.

That is if the vectors if the vectors lie on; if the vectors lie on let us say the on the units here, then one other useful then distance function at sometimes use is the angle between them right. And because a vectors are normalised the angle between them is I mean also satisfies the also satisfies triangle inequality and other properties that you need of a distance function.

(Refer Slide Time: 09:50)

Page 7/15

Hash Tables

- For exact search we used hashing
- Can we adapt hashing to search for “near”?
- Repurpose “collision”
– Instead of trying to avoid collisions, now we try to make collisions happen if the data points are nearby

IIT Gandhinagar
Indian Institute of
Technology Gandhinagar

NPTEL
CERTIFICATION COURSES

So, now, that we have a handle on the object representations as well as the distance function, let us talk about the possible solutions ok. Now we have seen in excellent candidate for a possible solution. Because if instead of near search right near duplicates we talking about exact duplicates then we knew what to do. You would have use the hash table right and the, and that is and that could have been perfectly fine because suppose here was a question, that given that that given a set of documents given a query here asking does this exactly same web page or the exactly same document appear in your data set. If you another answer exactly same, then you could have created the MD 5 hash of the entire document, and then you just index there is you just stored these M D 5 hashes in a in a hash table right and answering the query is very straightforward.

So, the question is now, can be adapt this hashing idea or the hash table idea to search for not exact match, but close enough match this is what will try to do here. So, and the clever idea by by Indyk Motwani this is really sort of pioneered by Indyk Motwani and there have been a lot of a huge line of literature and in this line, the clever idea is to repurpose the idea of collision right.

Because remember what was what we were doing when we were designing hash tables, we were trying to avoid collisions, because collisions would lead to increase in the query time. But now collisions are exactly what you are looking for right. What we want to do is that we want to repurpose collision so, that collisions now happened, remember in the

designing hash table collisions happened over x and y that are I mean arbitrary right I mean any x who have collided with any y .

So, now we repurpose collisions and say that can I make if x and y are similar, can I make them collide? Can I make them can I have a greater chance that they are that if the nearby they go to the same they go to the same hash bucket and if there for you go to different hash buckets. Basically this is really the idea, the fact that this is possible and the fact that this is possible efficiently, none of these are obvious questions right.

(Refer Slide Time: 12:22)

Hash Tables

- Want the following
 - Nearby points should fall in “same” bucket, points further away should fall in different buckets

The diagram illustrates a hash table with five buckets. Points x and y are close together and map to the same bucket (the second one from the left). Point z is further away and maps to a different bucket (the fifth one from the left).

So, here is what we want right. We want to say that supposing you have two point x and y that are close, but x is power from z and y is power from z . So, so maybe a one x and y to go to the same bucket, but you one z go to a different bucket right further of points should fall in different buckets and the close enough and points at a nearby to each other should fall in the same bucket, this is what we want.

(Refer Slide Time: 12:56)

Page: 9/17

Locality Sensitive Hashing

[Indyk Motwani]

- Hash family H is *locality sensitive* if
 - $\Pr[h(x) = h(y)]$ is high if x is close to y
 - $\Pr[h(x) = h(y)]$ is low if x is far from y
- Not clear such functions exist for all distance functions

IIT Gandhinagar
Indian Institute of Technology Gandhinagar

NP
CERTIFICATION COURSES

So, we need to define this a little more formally, and will sort of check through the definitions in sort of increasing order of formality formalism, but. So, here is one intuitive as to slightly more complete definition than before right I mean. So, how would be choosing the hash function? So, this cannot be done this cannot be done with deterministic hash functions. Instead we will again to go to this idea of choosing hash functions randomly and we will say that the algorithm will design a hash family right. That given a distance function given a distance function d or a similarity measure s right the algorithm will sort of design a hash family, and then a run time it will choose a single hash function from this family right. And we will call the family to be locality sensitive if right remember the probabilities of the choice the random choice of the hash function.

If the I mean over the choice of the hash function the probability that for any two x and y , the probability that h of x and h of y are the same right is high if x is close to y . That if close to y they have a higher probability of collision if there over the choice of the hash function, and if x and y are far apart the probability of collision is small again over the choice of the hash function right. See very important to note that here the probability here we not talking about any randomness in the input right, the input data is fixed specifically the x and y are fixed, and this has to work for all x and y over the random choice of the hash function right.

And then it is not even cleared that such functions exists right because how do I know. There are as I said there are many different similarity functions you already saw a few, a many different distance functions and how do I even know that it is possible to create such families right. For instance if I where choosing the hash family h to be the set of all functions right.

(Refer Slide Time: 15:20)

Page 10/12

Locality Sensitive Hashing

[Indyk Motwani]

- Hash family H is *locality sensitive* if
 - $\Pr[h(x) = h(y)]$ is high if x is close to y
 - $\Pr[h(x) = h(y)]$ is low if x is far from y
- Not clear such functions exist for all distance functions

Diagram: A blue arrow labeled u points to a blue-outlined rectangle representing a range from 0 to $m-1$.

IIT Gandhinagar
Indian Institute of Technology Gandhinagar

NP CERTIFICATION COURSES

Maybe may be the hash family so, here is the sample function right. May be may be the hash family is a set of all functions that go from u to 0 to m minus 1 right.

So, then this particular collision probability is independent of the distance. So, it does not certainly does not satisfy a locality sensitive property right. So, how do I even know that such hash families exists that is the first, that is the first thing that will see. That it is, that not only do such hash families exists, it is very easy to create. It is not very hard to create such hash families. So, that is the clever trick right.

(Refer Slide Time: 15:58)

The slide is titled "Locality sensitive hashing" and contains the following content:

- Originally defined in terms of a similarity function [C'02]
- Given universe U and a similarity $s: U \times U \rightarrow [0,1]$, does there exist a prob distribution over some hash family H such that

$$\Pr_{h \in H} [h(x) = h(y)] = s(x, y)$$

- $s(x, y) = 1 \rightarrow x = y$
- $s(x, y) = s(y, x)$

The slide footer includes the IIT Gandhinagar logo, the text "Indian Institute of Technology Gandhinagar", and "NP CERTIFICATION COURSES". A small video inset shows a man speaking.

So, again I mean before we see this, let us start we define this little more formally right and let us define this in terms of not the distance function, but in terms of similarity functions. All though there is a one to one, there will be a one to one link between the two.

So, and this is and this was the definition given by Charikar in 2002, given a universe U and a similarity function s right. So, what is the similarity function? The similarity function is nothing, but it takes a pair of elements, and it returns the similarity of this pair right. For instance the similar I mean and you can assume that is satisfy some simple properties for instance its now the similarity of x with y is 1 if and only if x equal to y right. And a similarity is a symmetric function similarity of x with y same the similarity of y with x right let us say similarity function is any function that satisfies the simple properties.

So, given universe U and a similarity function right does there exist the probability distribution, does there exist a hash family and such that we can choose from this hash family according to a probability distribution right, such that we satisfy this. That the probability of collision is exactly equal to the similarity between x and y ; the probability of collision of x and y is exactly equal to similarity between x and y ok. This is what we call to be let us the first definition of locality sensitive hashing according to similarity preserve a locality sensitive hashing ok.

Now that we ask this question, now that we are we have formalize it now, we can go back and ask that do such functions exists. Can I define we have already seen some natural notions of similarity let us say the Jaccard similarity, we will see couple of others to do functions exist for these natural notions of similarity.

(Refer Slide Time: 18:00)

Page 12/20

Hamming distance

- Points are bit strings of length d
- $H(x, y) = |\{i, x_i \neq y_i\}|$ $S_H(x, y) = 1 - \frac{H(x, y)}{d} \in [0, 1]$

- $x = \overset{1}{\underset{|}{1}}\overset{0}{\underset{|}{0}}\overset{1}{\underset{|}{1}}\overset{1}{\underset{|}{1}}\overset{0}{\underset{|}{0}}\overset{0}{\underset{|}{0}}\overset{0}{\underset{|}{1}}$, $y = \overset{0}{\underset{|}{0}}\overset{1}{\underset{|}{1}}\overset{1}{\underset{|}{1}}\overset{0}{\underset{|}{0}}\overset{1}{\underset{|}{1}}\overset{0}{\underset{|}{0}}\overset{1}{\underset{|}{1}}$
 - $H(x, y) = 3$ $S_H(x, y) = 1 - \frac{3}{10} = 0.7$

IIT Gandhinagar Indian Institute of Technology Gandhinagar NP CERTIFICATION COURSES

Let us take a simple one. The, suppose my points are bit strings of length d right. So, so one natural notion of similarity or distance here is known as the hamming distance, what is a hamming distance? Hamming distance is nothing, but if you a given two if I given two strings of length d , having distance between them is the number of positions in which they differ. For instance I have I have written two strings of length 10 here x and y , and the positions in which the differ r here, here because here is 1 there is a 0, there is a 0 there is a 1 all of this is the similar and I think at this point at this point right.

So, the hamming distance between them is 3 right; because they differ in exactly 3 positions. If an also define the similarity hamming similarity among them, that as 1 minus H of x y by d . So, this lies in 0 1 right and it satisfies all the property of a the similarity, of a similarity function. I mean the hamming similarity between these two is 1 minus 3 by 10 which is 0.7. Ok and if the hamming similarity if a write in x and x is 1 and; that means, that the two bit strings are exactly the same because a do not differ in any position.

So then has the, has a question then does this similarity have a corresponding locality sensitive hash function.


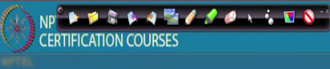

(Refer Slide Time: 19:39)

Page 14/22

Hamming distance

- Points are bit strings of length d
- $H(x, y) = |\{i, x_i \neq y_i\}|$ $S_H(x, y) = 1 - \frac{H(x, y)}{d}$
- Define a hash function h by sampling a set of positions
 - $x = 1011010001, y = 0111010101$
 - $S = \{1, 5, 7\}$
 - $h_5(x) = 100, h_5(y) = 100$

$\Pr_{h \in H} [h(x) = h(y)] = S_H(x, y)$
 $H = \{h_i(x) = x_i, i=1 \dots d\}$
 $|H| = d$

It turns out yes very interestingly right let us define this they hash family. What we say is that the hash family really consists of let us say we take all subsets of size s . So, there is a there is an element of the family, let us say that let us define it simply first. Let us say that there is an element of the family such that h_i of x is equal to x_i . So, the first family that I define is a very simple family of size d right what are the members of the family? There is exactly d hash functions and the d th hash function is nothing, but it something that picks up the i th coordinate right and something that picks up the i th coordinate right just 1 bit 0 or 1.

So, let us look at that first; let us look at that first then will do the analysis for the larger d . So, so what can we say? What we can say that if you pick h from H what is the probability that h of x ? Will equal h of y right, is somebody hard to see that picking H picking the hash function is really sort of deciding which coordinate we focus on? That we focus on one of the coordinates and we see that do x and y are x and y the same in this coordinate are not right and therefore, this probability is exactly equal to the similar the hamming similarity between x and y , because that is exactly the fraction of coordinates on which they are same. If they are same in 70 percent of the coordinates, then according to if I pick a hash function from this family the probability of them being

the same will be exactly 70 percent ok. And so, this particular family satisfies exactly the same, satisfies the I mean is actually locality sensitive hash family according to the definition that we give right.

What we can do, what we often by something a little different right what we do is to say that, let us pick instead of a single coordinate, let us pick k coordinates right. That supposing if we are picking three coordinates 1 5 7 right and we define a hash function out of these three coordinates.

For instance if I pick s to be 1 5 7, which is the position 1 2 3 4 5 position 5, position 7 right and then we look at for x we look at only the bits in positions 1 5 7 that gives us once 1 0 0. For y we can look at positions 1 5 and 7 that again gives me 1 0 0 right therefore, the hash function h s really is than the projection of the of the strings x and y, on to this on to this particular coordinates. Then you can ask the question that what is the probability that; what is the probability that h of x equals the h of y right?

(Refer Slide Time: 23:43)

Existence of LSH

- The above hash family is locality sensitive, $k = |S|$

$$\Pr[h(x) = h(y)] = \left(1 - \frac{H(x,y)}{d}\right)^k = \left(S_H(x,y)\right)^k$$

Page 13/23

IIT Gandhinagar Indian Institute of Technology Gandhinagar NP CERTIFICATION COURSES

And here now because we have used because we have used k coordinates, the probability is now s h x y to the power k right. Because we had choosing; because we have choosing k coordinates let us say we have choosing them with without we choosing a with replacement right and so, the probability and so, for a for any one coordinate right the probability that the they are saying is S H of x y. So, for k coordinates the probability that all of them are similar across x and y is S H x y to the power k.

So, we see that at least for a very very simple distance or similarity function, we do have a notion of locality sensitive hashing let us go to a more complicated functions.

(Refer Slide Time: 24:46)

Page 16/24

LSH for angle distance

- x, y are unit norm vectors $x, y \in \mathbb{R}^d$
- $d(x, y) = \cos^{-1}(x \cdot y) = \theta$
- $S(x, y) = 1 - \theta/\pi \in [0, 1]$

So, here is one that we have just mentioned supposing x and y are unit norm vectors right which means that they lie on the 1 mean surface of the unit ball. So, they can live in \mathbb{R}^d . So, x and y belong to \mathbb{R}^d , but because we are looking at only two vectors x and y if we might as well draw the ball into dimensions. So, the distance between x and y is really the angle between them, the θ angle between them. So, the similarity between x and y could be defined as $1 - \theta/\pi$.

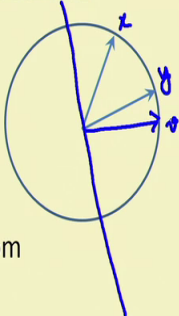
So, I am measuring that causing was always written of value it would be between 0 and π . So, therefore, this gives me a value between 0 and 1 ok. So, this is the similarity and this is the distance. So, here the distance is more natural to define than the similarity. So, let us now look at. So, for this particular similarity function right is there a corresponding notion of locality sensitive hashing? And it turns out there is and it is a very clever one It is a.

(Refer Slide Time: 26:08)

Page 17/25

LSH for angle distance

- x, y are unit norm vectors
- $d(x, y) = \cos^{-1}(x \cdot y) = \theta$
- $S(x, y) = 1 - \theta/\pi$
- Choose direction v uniformly at random
 - $h_v(x) = \text{sign}(v \cdot x)$ +1, -1, 0
 - $\Pr[h_v(x) = h_v(y)] = 1 - \theta/\pi$



IIT Gandhinagar
Indian Institute of
Technology Gandhinagar

NP
CERTIFICATION COURSES

17

What we do is this, we say that we choose let me draw this first and then will and explain, we choose a vector uniformly at random right we choose it we choose a vector uniformly at random, and then and then we say then we say that take the dot product of this vector with x right take the dot product of this vector with x , and then see the sign of that and take that sign as the hash as the hash value.

See the sign is plus 1 or minus 1 right or you can think of it a 0 and then yeah 1 and then 0, I mean if basically you just gives you a single bit. So, but what is the sign. So, you just gives just look at it geometrically. So, if I take this blue vector, if I take the sign of x along sign of beta text its really nothing, but you take the hyper plane that is orthogonal to the to this vector v right and then you see which side of the hyper plane the sign veta text tells you which sign site of the, of this particular hyper plane x falls n right. So, one side is the positive side the other side is the negative side.

So, now the question is that why does it satisfy I want to claim that is satisfy this locality sensitive property right, that is I want to say that using the definition of the similarity between x and y , I want to say that the probability that h_v of x equals h_v of y is nothing, but $S(x, y)$ which is equal to $1 - \theta/\pi$ why is that the case right? And you see see to sort of understand it from the geometry because on the geometry you see that when does the hyper plane see, we can think of the hyper plane and the vector v are in one to one correspondence.

So, I could as well think of choosing the hyper plane, instead of choosing the vector v randomly. So, why in does a hyper plane give you different signs for x and y , the hyper plane gives you exactly difference signs for x and y , when it passes through the when the hyper plane passes through the angle between x and y right. That means, that when the vector is chosen such that the corresponding hyper plane passes through this angle between x and y at that point, sign of $v \cdot x$ is different from sign of $v \cdot y$ and what is the probability of that right. The probability of that, I mean the probability of splitting is exactly θ by π because hyper plane will be chosen.

I mean because the because angle will vary from 0 to π and therefore, the probability of choosing this angle right the probability of splitting x and y will be when the when the hyper plane passes through that, and because it is uniform the angle is chosen in the vector is chosen uniformly at random from the surface of the unit sphere, the probability of this is, probability of splitting is exactly θ by π .

(Refer Slide Time: 29:19)

Page 18/28

LSH for angle distance

- x, y are unit norm vectors
- $d(x, y) = \cos^{-1}(x \cdot y) = \theta$
- $S(x, y) = 1 - \theta/\pi$

- Choose direction v uniformly at random
 - $h_v(x) = \text{sign}(v \cdot x)$
 - $\Pr[h_v(x) = h_v(y)] = 1 - \theta/\pi = S(x, y)$

IIT Gandhinagar Indian Institute of Technology Gandhinagar NP CERTIFICATION COURSES

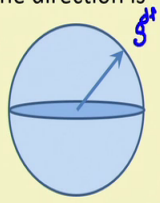
And therefore, the probability of not splitting that is a probability of collision is exactly 1 minus θ by π sees are.

So, then we also see that even for this angle distance there is a particular locality sensitive hash function right. So, we simply prepared a lucky in our choice of distance functions and so on.

(Refer Slide Time: 29:51)

Aside: picking a direction u.a.r.

- How to sample a vector $x \in R^d, |x|_2 = 1$ and the direction is uniform among all possible directions
- Generate $x = (x_1, \dots, x_d), x_i \sim N(0, 1)$ iid
- Normalize $\frac{x}{|x|_2}$
– By writing the pdf of the d-dimensional Gaussian in polar form, easy to see that this is uniform direction on unit sphere



IIT Gandhinagar
Indian Institute of
Technology Gandhinagar

NP
CERTIFICATION COURSES

18

So, the question then is a is this slag going to sustain. So, before we go there and that is what we will do a next class, we will let us just do a little bit of an a site for a minute about how to pick a vector, how to pick a direction uniformly at random right because that is one entirely clear. I mean I told you that a sample v as a uniform random direction and space, but how to we do this right.

And it is a say very clever trick. So, and that is why we want to see this. That suppose we want to choose in d dimensions, we want to choose I mean a vector x such that two norm of x is 1 and the direction of the vector is uniform among all possible directions in the in d dimensions, which means that it lies on a surface of the ball S^{d-1} .

This is known as the unit ball of radius t minus 1 of the unit ball and dimension d minus 1. So, so here is the way to do this right. What you do is that you take Gaussian distribution you take you take $N(0, 1)$ right and then you make d independent samples, from this call these x_1 to x_d these are the values of the initial values of the d coordinates ok. And then you normalize it with the length of the vector x .

So, recall the length of the vector is also random variable right, but you take x and then we divide by the you calculate the l_2 norm of the of x , the l_2 norm of x is as you know its summation x_i^2 square square root right and then you normalize it. So, now, it is certainly have a unit norm vector right. So, how do I know that it has a uniform direction in this in this fear?

So, for that what we need to do is that, you need to write the equation of the d dimension Gaussian in polar form, and what you will be able to see that that because we writing it in polar form, the part the r and the θ the r which sort of is the captures the length and the θ which captures the which captures the direction, they can be treated independently.

And because here we have effectively normalized by they as they can be treated as independent random variables right and here we are normalised it with the with the with the norm of the vectors. So, the R effect goes away not here left with is a uniformly chosen direction right. And this is a again a useful exercise to perform that we will possibly also look at in our in our exercises that we have to use.

So, with this a side let us let us end today, what we saw is that we saw the definition of we saw some initial definitions of locality sensitive hashing, we gave a slightly formal definition of it in terms of similarities. Then we were wondering whether such locality sensitive hash functions even exist, and at least for two of the similarity measures, two of the useful similarity measures the Jaccard similarity as well as the angle ways similarity, with its see that we are able to create using very clever very simple sampling methods are locality sensitive hash family.

Next we will extend this to other kinds of similarity measures, and we will see how to create a hash table out of this locality sensitive hash measures and that is it for today.

Thank you.