CS-E4600 – Lecture notes #4
Exercise session 1

Han Xiao and Suhas Thejaswi

September 22, 2018

1 Distances in graphs

Given an *u*ndirected graph $G = (V, E, w)$ where V is the set of vertices, E is a set of edges and a function $w : E \rightarrow (0, 1]$. Let $d(u, v)$ denote the shortest path between any two vertices $u, v \in V$. The *s*hortest-distance between $u, v \in V$ is defined as the length of a shortest path from u to v in G.

In order to prove that a distance function is a metric, it should satisfy the following four properties: (i) non-negativity: for any $u, v \in V$ the distance $d(u, v)$ should be grater than or equal to 0. (ii) coincidence: for any $d(u, v) = 0$ if and only if $u = v$. (iii) symmetry: for any $u, v \in V$, $d(u, v) = d(v, u)$. (iv) triangle inequality: for any $u, v, z \in V$, $d(u, v) \leq d(u, z) + d(z, v)$. However, to prove that a distance function is not a metric it is sufficient to prove that at least one of these properties do not hold.

1.1 Prove or disprove that shortest path distance is a metric.

The shortest-path distance is a metric.

Non-negativity: For any edge $\{u, v\} \in E$, its edge weight are the range $0 < w(\{u, v\}) \leq 1$, it is easy to see that distance of all paths in G are non-negative.

Coincidence: Since all edge weights are non-negative there exists no path with distance $d(u, v) = 0$ between any two different vertices $u, v \in V$ and the distance from a vertex to itself is always 0 (assuming that graph is simple).

Symmetry: For any $u, v \in V$, let P be the shortest path with distance $d(u, v)$. Since G is undirected we can follow the path P' in the reverse order of P from v to u and the distances are equal. If P' is not minimum, it contradicts the minimality of P. So symmetry holds.

Triangle inequality: For any $u, v, z \in V$, let $P(u, v)$, $P(u, z)$, $P(z, v)$ be a shortest path between u to v, u to w and w to v, respectively. Let $d(u, v)$, $d(u, z)$, $d(z, v)$ be the distance of paths $P(u, v)$, $P(u, z)$ and $P(z, v)$, respectively. Let us assume that $d(u, v) > d(u, z) + d(z, v)$, we can replace the path from $P(u, v)$ by a path $P'(u,v) = P(u,z) + P(z,v)$ such that $d'(u,v) < d(u,v)$ which contradicts the minimality of $P(u,v)$.

1.2 What if the graph is directed?

Symmetry do not hold. Since the graph is directed the length of a shortest path from u to v need not be same as the length of a shortest path from v to u for any $u, v \in V$.

1.3 What if the edge weigths are drawn from the normal distribution $\mathcal{N}(0,1)$?

If a negative edge weight is drawn then the distance function is not a metric.

2 String edit distance

The $s$tring edit distance is defined as the minimum number of edit operations required to transform one string to another and the transformation operations include: (i) insert: add one character in one string at any position, (ii) delete: delete one character in one string at any position and (iii) insert: insert one character in one string at any position. Refer lecture notes (02-distances) for a detailed explanation.

2.1 Prove or disprove that string edit distance is a metric.

The string-edit distance is a metric. Let X and Y be two strings, we denote the string edit distance between X and Y as $d(X,Y)$.

Non-negativity: Since the distance is measured as the minimum number of edit operations required, it is trivial to see that the distance is non-negative.

Coincidence: If $X = Y$ then there is no transformation operation required and hence $d(X,Y) = 0$. If $X \neq Y$ then we need atleast one edit operation to transform X to Y and viceversa.

Symmetry: Let $T = (t_1, t_2, \ldots, t_n)$ be a minimum set of transformations from X to Y with distance $|T|$, then we can use $T' = (t_n, t_{n-1}, \ldots, t_1)$ to transform Y to X and $|T| = |T'|$.

Triangle inequality: Let X, Y and Z be three strings. Let $T_{X,Y}$, $T_{X,Z}$ and $T_{X,Y}$ be a minimum set of transformations from X to Y, X to Z and Z to Y. We claim that $|T_{X,Y}| \leq |T_{X,Z}| + |T_{Z,Y}|$. For the sake of contradiction, let us assume that $|T_{X,Z}| + |T_{Z,Y}| > |T_{X,Y}|$, then we can replace $T_{X,Y}$ by $T'_{X,Y} = T_{X,Z} \cup T_{Z,Y}$ such that $T'_{X,Y} < T_{X,Y}$ which contradicts the minimality of $T_{X,Y}$.

3 Dynamic time warping (DTW)

Refer to the lecture notes for the definition.

3.1 Prove or disprove that DTW distance between time-series is a metric.

Recall that in order prove a distance function is not a metric it is often sufficient to show that it violates at least one of the four properties. Here we will show that DTW distance voilates coincidence.

Let $T_1 = (1, 1, 0)$ and $T_2 = (1, 0, 0)$ be two time series. The DTW distance between T_1 and T_2 is $d(T_1, T_2) = 0$. However, $T_1 \neq T_2$, which voilates the conincidence property. An illustration of the example is provided in Figure 1

4 Mean and median

4.1 Find a real number x that minimises $\sum_{i=1}^{n} |x - x_i|^2$.

The value of x that minimises $\sum_{i=1}^{n} |x_i - x|^2$ is the mean. Let us denote the mean of X by $\bar{x}$.

$$\sum_{i=1}^{n}(x_i-x)^2 = \sum_{i=1}^{n}((x-\bar{x})+(\bar{x}-x_i))^2 = \sum_{i=1}^{n}(x-\bar{x})^2+\sum_{i=1}^{n}(\bar{x}-x_i)^2+2(\bar{x}-x)\sum_{i=1}^{n}(\bar{x}-x_i)$$

Since,

$$\sum_{i=1}^{n}(\bar{x} - x_i) = \bar{x} - x_1 + \bar{x} - x_2 + \cdots + \bar{x} - x_n = n \cdot \frac{1}{n}\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i = 0$$

$$\sum_{i=1}^{n}(x_i - x)^2 = n(x - \bar{x})^2 + \sum_{i=1}^{n}(\bar{x} - x_i)^2$$

The sum is minimum if $x = \bar{x}$.

4.2 Find a real number x that minimises $\sum_{i=1}^{n} |x - x_i|$.

The value of x that minimises $\sum_{i=1}^{n} |x - x_i|$ is the median.

Let us assume that $x_1 \leq x_2 \leq \cdots \leq x_n$ if not we can reorder X in sorted order. Let us assume that $x_i \leq x \leq x_{i+1}$ for some $i \in [n]$. Then the sum is computed as,

$$\sum_{j \leq i}(x - x_j) + \sum_{j > i}(x_j - x)$$

This is a linear function restricted to the interval $[xi, x_{i+1}]$ with gradient equal to the number of j such that $j \leq i$ minus the the number of j such that $j > i$. The gradient is zero if it is less than a median, it is positive if greater than a median, and zero precisely when there are as many data points to the left and right.

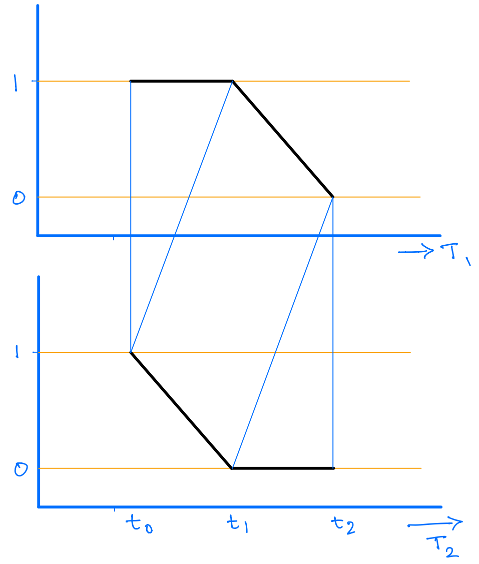

Figure 1: Illustration of DTW distance violating coincidence property.

5 Median string

There is an algorithm with $c = 2$ that does the following:

- Calculate the distances between each object in the set X and save the distances to an $n \times n$ table.
- Calculate the sum of distances by taking a sum of each row in the table.
- Find a row with the minimum sum of the calculated distances.
- Output the object with the index of the row found in step 3.

Proof that this algorithm gives an approximation of $c = 2$:
First denote x_0 as the object closest to x^* and $\bar{x}$ as the object in X that minimizes $\sum_x d(x, \bar{x})$
Then, we have

$$\begin{aligned}
\sum_x d(x, \bar{x}) &\leq \sum_x d(x, x_0) \\
&\leq \sum_x (d(x, x^*) + d(x^*, x_0)) \\
&\leq \sum_x d(x, x^*) + nd(x^*, x_0) \\
&\leq \sum_x d(x, x^*) + \sum_x d(x, x^*) \\
&= 2 \sum_x d(x, x^*)
\end{aligned}$$

6 VP-tree

6.1 How VP data structure works

Refer to page 42 of the course slides (04-simsearch.pdf)

6.2 Construction time complexity

Define S as the set of points and $n = |S|$. Then the depth of the tree is $O(\log n)$.

Distance computation At the first tree level, it takes $O(n)$ (constant D can be dropped) to compute the distance for all points, then nodes are split into two equal-sized groups and the recursion can be expressed as

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n),$$

where T represents the time complexity.

Applying the Master Theorem [1],

$$T(n) = aT\left(\frac{n}{b}\right) + O(n^c)$$

we get $a = 2, b = 2, c = 1$. Thus, $a = b^c$ and by using the result of Master Theorem,

$$T(n) = O(n \log n)$$

Finding median and partitioning At each tree node, we randomly select a vantage point and partition the points by finding the median distance.

To find the median in n numbers, it takes $O(n \log n)$ to sort and $O(1)$ to find median.[1] Thus, the time recursion can be expressed

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n \log n)$$

This is identical to distance computation, thus

$$T(n) = O(n \log^2 n)$$

Combined together, the total time complexity for tree construction is $O(n \log^2 n)$.

6.3 Space complexity

The space taken by VP-tree is $O(n)$. It is because at each level a node in the tree stores, only the vantage point, median distance (number), and 2 references to left and right subtrees if there are any (so in fact space is $O(4n)$, but the constant can be omitted).

6.4 Pruning rules

Let v be the vantage point, rank points by distance to v

$$x_1, \ldots, x_{N/2}, x_{N/2+1}, \ldots, x_N$$

such that

$$d(x_1, v) \leq \ldots \leq d(x_{N/2}, v) \leq d(x_{N/2+1}, v) \leq \ldots \leq d(x_N, v)$$

Then define $L(v) = \{x_1, \ldots, x_{N/2}\}$ and $R(v) = \{x_{N/2+1}, \ldots, x_N\}$.

Let d^* be the distance of query q to current NN and $R = d(x_{N/2}, v)$ (the radius)

The pruning rule considers two cases (illustrated in Figure2):

[1] Linear time median finding is possible [2].

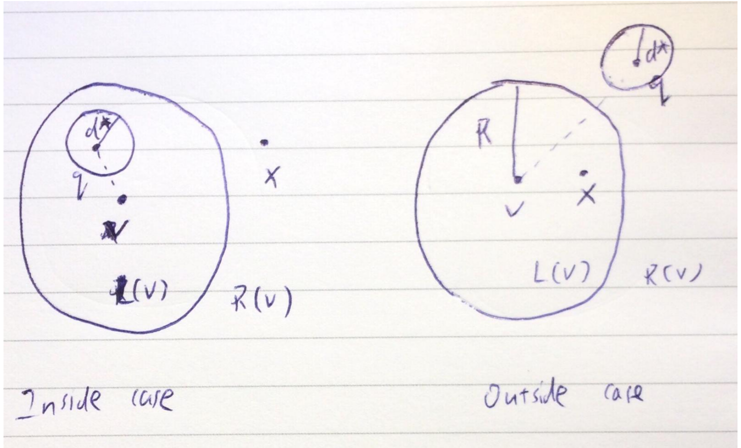

Figure 2: Illustration of pruning rules: “inside” case (left figure), “outside case (right figure)

1. “Inside” case: if $R \geq d(q, v) + d^*$ we can safely prune the right side of v, since we already found a current nearest neighbor n inside the area defined by R. Traversing along $L(v)$ may reveal points closer than the current nearest neighbor n.
2. “Outside” case: if $R \leq d(q, v) - d^*$ we can safely prune the left side of v, since we already found a current nearest neighbor n that is closer than any point contained in the inner area defined by R. Traversing along $R(v)$ of v may reveal points closer than the current nearest neighbor n.

6.5 Query algorithm

An query algorithm is given in Algorithm 1.

6.6 Correctness proof

Consider the pruning rule 1 and given a node x in $R(v)$, we know:

$$d(x, v) \geq R$$

$$R \geq d(q, v) + d^*$$

$$d(x, v) \leq d(x, q) \geq d(v, q) \text{ (triangle inequality)}$$

Combing them, we have

$$d(x, q) > d^*$$

Similarly, we can prove the correctness of rule 2.

Algorithm 1: Sequential search algorithm for Nearest-Neighbor Query given the root node of a VP-tree T and a query q; returns exact nearest neighbor in variable n. We denote by d^* the smallest distance seen so far from the current nearest neighbor n. P is defined as a list of nodes to visit. By R we denote the distance to the median dividing $L(v)$ and $R(v)$, with x be the node in $L(v)$ and $R(v)$.

```
Data: VP-tree root node $T$, query $q$
Result: Nearest Neighbor $n$
$d^* \leftarrow \infty$;
$P \leftarrow$ FIFO queue $[T]$;
while $|P| > 0$ do
    $v \leftarrow$ pop from $P$;
    $d_{q,v} \leftarrow d(q,v)$;
    if $d_{q,v} \leq d^*$ then
        $n \leftarrow v$;
        $d^* \leftarrow d_{q,v}$;
    if $d_{q,v} \leq R + d^*$ and $x \in L(v)$ exists then
        push $x$ into $P$;
    if $d_{q,v} \geq R - d^*$ and $x \in R(v)$ exists then
        push $x$ into $P$;
return $n$
```

6.7 Better vantage point selection

One way to do this is to select the VP that minimizes the median. We would expect that left subtrees are more likely to be pruned.

Similarly, we can select the VP that maximizes the median so that right sub trees are more likely to be pruned.

Another idea from [3] is to choose vantage points at the corner of the space. Computationally, point that has the maximum distance variance tends to be at the corner.

References

[1] Wikipedia. Master theorem — wikipedia, the free encyclopedia, 2016. [Online; accessed 6-October-2016].

[2] Wikipedia. Median of medians — wikipedia, the free encyclopedia, 2016. [Online; accessed 29-September-2016].

[3] Peter N Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces.