Streaming Algorithms

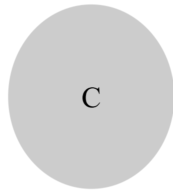
Meng-Tsung Tsai

05/08/2018

2VC sparse certificate

<u>Claim</u>. Let G be any undirected graph, and let $B_i(G)$ be any BFS forest of $G \setminus (B_1(G) \cup \dots \cup B_{i-1}(G))$. Then a node v is an articulation point in $B_1(G) \cup B_2(G)$ if and only if v is an articulation point in G.

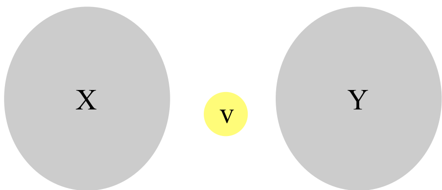
A connected component C in $B_1(G) \cup B_2(G)$ that contains v.



2VC sparse certificate

<u>Claim</u>. Let G be any undirected graph, and let $B_i(G)$ be any BFS forest of $G \setminus (B_1(G) \cup \dots \cup B_{i-1}(G))$. Then a node v is an articulation point in $B_1(G) \cup B_2(G)$ if and only if v is an articulation point in G.

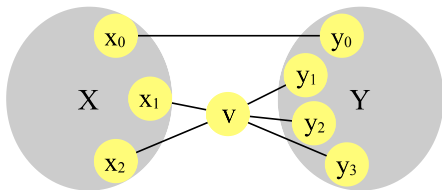
Partition C into X and Y by removing v.



2VC sparse certificate

<u>Claim</u>. Let G be any undirected graph, and let $B_i(G)$ be any BFS forest of $G \setminus (B_1(G) \cup \dots \cup B_{i-1}(G))$. Then a node v is an articulation point in $B_1(G) \cup B_2(G)$ if and only if v is an articulation point in G.

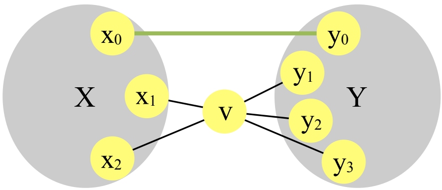
The induced subgraph of C in G.



2VC sparse certificate

Claim. Let G be any undirected graph, and let $B_i(G)$ be any BFS forest of $G \setminus (B_1(G) \cup \ldots \cup B_{i-1}(G))$. Then a node v is an articulation point in $B_1(G) \cup B_2(G)$ if and only if v is an articulation point in G.

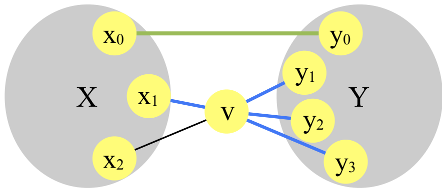
Any crossing edge from X to Y is not contained in $B_1(G) \cup B_2(G)$.



2VC sparse certificate

Claim. Let G be any undirected graph, and let $B_i(G)$ be any BFS forest of $G \setminus (B_1(G) \cup \ldots \cup B_{i-1}(G))$. Then a node v is an articulation point in $B_1(G) \cup B_2(G)$ if and only if v is an articulation point in G.

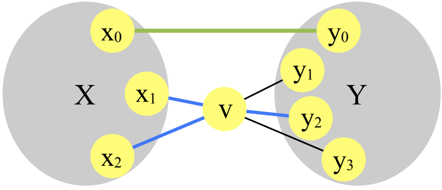
Any $B_1(G)$ looks like either:



2VC sparse certificate

Claim. Let G be any undirected graph, and let $B_i(G)$ be any BFS forest of $G \setminus (B_1(G) \cup \ldots \cup B_{i-1}(G))$. Then a node v is an articulation point in $B_1(G) \cup B_2(G)$ if and only if v is an articulation point in G.

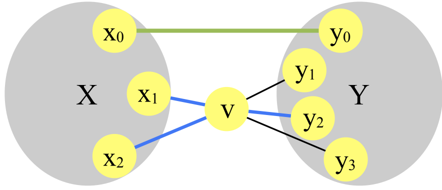
Any $B_1(G)$ looks like or:



2VC sparse certificate

Claim. Let G be any undirected graph, and let $B_i(G)$ be any BFS forest of $G \setminus (B_1(G) \cup \ldots \cup B_{i-1}(G))$. Then a node v is an articulation point in $B_1(G) \cup B_2(G)$ if and only if v is an articulation point in G.

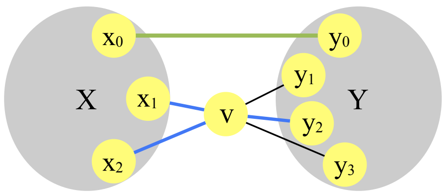
Therefore, X and Y are disconnected in $B_2(G)$.



2VC sparse certificate

Claim. Let G be any undirected graph, and let $B_i(G)$ be any BFS forest of $G \setminus (B_1(G) \cup \ldots \cup B_{i-1}(G))$. Then a node v is an articulation point in $B_1(G) \cup B_2(G)$ if and only if v is an articulation point in G.

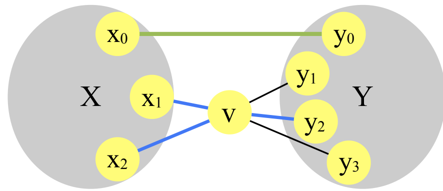
Since $B_2(G)$ is a BFS forest, there exists no crossing edge from X to Y.



2VC strong certificate

Claim. Let G and H be any undirected graph. Then a node v is an articulation point in $B_1(G) \cup B_2(G) \cup H$ if and only if v is an articulation point in $G \cup H$.

Induced graph of C in $G \cup H$.



No X-Y crossing edge in $B_1(G) \cup B_2(G) \cup H$.

X and Y are disconnected in $B_2(G)$.

If there is an X-Y crossing edge, it must be contained in $B_2(G)$.

Thus, v is an articulation point in $G \cup H$.

Exercise 1

Prove that $B_1(G) \cup B_2(G) \cup \ldots \cup B_{k+1}(G)$ is a strong k-VC certificate.

An Application to Turán's Theorem

If an n-node graph G has $> n^2/4$ edges, then G must have a triangle. Output any triangle in G using $O(n^2)$ time.

An Application to Turán's Theorem

Triangle-in-Dense-Graphs(G){
    H $\leftarrow$ the complement graph of G; // $e(H) < n^2/4 - n/2$
    Find an independent set in H that matches Turan's bound;
    // $\alpha(H) > n^2 / (n+2e(H)) = 2$
}

An Application to Turán's Theorem

Turan's-Independent-Set(H){
    I $\leftarrow \varnothing$;
    while(H $\neq \varnothing$){
        v $\leftarrow$ the minimum degree node in H;
        I $\leftarrow$ I $\cup$ {v};
        H $\leftarrow$ H \ {v} \ N(v);
    }
}

An Application to Turán's Theorem

Let $\beta(H) = t$ be the size of the independent set returned by the greedy algorithm. Let d_i be the degree of the i-th removal node. Clearly, d_i+1 nodes are removed in the i-th rounds.

$$\sum_{i\in[t]} d_i+1 = n.$$

Moreover, the edges removed in the i-th rounds is at least $d_i(d_i+1)/2$.

$$\sum_{i\in[t]} d_i(d_i+1)/2 \leq m.$$

Thus, $\sum_{i\in[t]} (d_i+1)^2 \leq 2m+n$.

By Cauchy-Schwarz Ineq, $\sum_{i\in[t]} (d_i+1)^2 \geq (\sum_{i\in[t]} (d_i+1))^2/t = n^2/t$.

Consequently, $t \geq n^2 / (n+2m)$.

References

• "Sparsification - A Technique for Speeding Up Dynamic Graph Algorithms," Eppstein et al. (1997)

• "Tight Bounds for Lp Samplers, Finding Duplicates in Streams, and Related Problems," Jowhari et al. (2010)

L_0-sampler

Input: a sequence of updates (c_i, Δ_i) where $c_i \in \{1, 2, ..., U\}$. Let $f_j = \sum_i \mathbf{1}[c_i = j]\Delta_i$.

Output: a random variate X so that if $f_k \neq 0$

$$\Pr[X = k] = 1/|F|$$

where $F = \{j \in U : f_j \neq 0\}$.

Goal: using O(poly log n) space and a single pass.

Applications to Dynamic Graph

Input: a sequence of edge insertions and deletions.

Output: a random edge sampled uniformly at random from the final graph.

Goal: using O(poly log n) space and a single pass.

We can sample T edges from the final graph using O(T poly log n) space.

Idea

Let $Q_k \subseteq [U]$ be a "random" subset of size 2^k.

Then there exists an integer k so that $|F \cap Q_k| \in [s/2, s]$ for some constant s w.h.p.

Let $h: U \to [s^2]$ be an s-wise independent hash function.

Then h(x) for all x in $F \cap Q_k$ are collision-free with some constant probability.

Algorithm

```
Update(){
   foreach (c, Δ){
      if (c in Q_k){
         A[h(c)] += Δ; // A[h(c)] is used to count f_c w.h.p.
         B[h(c)] += Δ*c;
      }
   }
}

Query(){
   Sample an index k uniformly from {x ∈ [1, c^2] : A(x) ≠ 0};
   Return B(k)/A(k);
}
```

Issues

1. How to determine k?

2. How to represent Q_k using small space?