Sensitivity Analysis of the Maximum Matching Problem

Yuichi Yoshida
Samson Zhou
Average Sensitivity for Graph Algorithms

- Measure of how much a discrete algorithm changes when the input changes
- For a deterministic graph algorithm $A$ on a graph $G$ with $n$ vertices, the average sensitivity is defined as

$$\mathbb{E}_{e \sim E(G)} [|A(G) \Delta A(G - e)|]$$

where $\Delta$ is the symmetric difference
Average Sensitivity for Graph Algorithms

- Example: Greedy algorithm for maximal matching on graph with 6 nodes
Average Sensitivity for Graph Algorithms

- Example: Greedy algorithm for maximal matching on graph with 6 nodes
Average Sensitivity for Graph Algorithms

- Example: Greedy algorithm for maximal matching on graph with 6 nodes
Average Sensitivity for Graph Algorithms

- Example: Greedy algorithm for maximal matching on graph with 6 nodes
Average Sensitivity for Graph Algorithms

❖ Example: Greedy algorithm for maximal matching on graph with 6 nodes
Average Sensitivity for Graph Algorithms

- Example: Greedy algorithm for maximal matching on graph with 6 nodes
Average Sensitivity for Graph Algorithms

- Example: Greedy algorithm for maximal matching on graph with 6 nodes
Average Sensitivity for Graph Algorithms

- Example: Greedy algorithm for maximal matching on graph with 6 nodes
Average Sensitivity for Graph Algorithms

❖ Example: Greedy algorithm for maximum matching on graph with 6 nodes
❖ $|A(G) \triangle A(G - e)| = 3$
Average Sensitivity for Graph Algorithms

❖ Measure of how much a discrete algorithm changes when the input changes
❖ For a randomized graph algorithm $A$ on a graph $G$ with $n$ vertices, the average sensitivity is defined as

$$\mathbb{E}_{e \sim E(G)}[d_{EMD}(A(G), A(G - e))]$$

where $d_{EMD}$ is the Earth Mover Distance
Average Sensitivity for Graph Algorithms

- Introduced by Murai and Yoshida [MY19] for centrality

- betweenness centrality
- harmonic centrality
- spanning tree centrality
- PageRank
- closeness centrality
Average Sensitivity for Graph Algorithms

- Introduced by Murai and Yoshida [MY19] for centrality
- Varma and Yoshida [VY21] for graph algorithms that output edge/vertex sets

<table>
<thead>
<tr>
<th>Problem</th>
<th>Output</th>
<th>Approximation Guarantee</th>
<th>Average Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Spanning Forest</td>
<td>Edge set</td>
<td>1</td>
<td>$O\left(\frac{n}{m}\right)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$&lt; \infty$</td>
<td>$\Omega\left(\frac{n}{m}\right)$</td>
</tr>
<tr>
<td>Global Minimum Cut</td>
<td>Vertex set</td>
<td>$2 + \epsilon$</td>
<td>$n^{O\left(\frac{1}{\epsilon \text{OPT}}\right)}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$&lt; \infty$</td>
<td>$\Omega(n)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\Omega\left(\frac{n^{\text{OPT}}}{\text{OPT}^2}\right)$</td>
</tr>
<tr>
<td>Minimum s-t Cut</td>
<td>Vertex set</td>
<td>additive $O(n^{2/3})$</td>
<td>$O\left(n^{2/3}\right)$</td>
</tr>
<tr>
<td>Maximum Matching</td>
<td>Edge set</td>
<td>$1/2$</td>
<td>$1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1 - \epsilon$</td>
<td>$\tilde{O}\left(\frac{\text{OPT}^{1/3}}{\epsilon^{2/3}}\right)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1$</td>
<td>$\Omega(n)$</td>
</tr>
<tr>
<td>Minimum Vertex Cover</td>
<td>Vertex set</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2-Coloring</td>
<td>Vertex set</td>
<td>—</td>
<td>$\Omega(n)$</td>
</tr>
</tbody>
</table>
Average Sensitivity for Graph Algorithms

- Introduced by Murai and Yoshida [MY19] for centrality
- Varma and Yoshida [VY21] for graph algorithms that output edge/vertex sets
- Peng and Yoshida [PY20] for spectral clustering
From Average Sensitivity to Worst-Case Sensitivity

- We introduce the definition of worst-case sensitivity
- For a randomized graph algorithm $A$ on a graph $G$ with $n$ vertices, the average sensitivity is defined as

\[
\mathbb{E}_{e \sim E(G)} \left[ d_{EMD}(A(G), A(G - e)) \right]
\]

- For a randomized graph algorithm $A$ on a graph $G$ with $n$ vertices, the worst-case sensitivity is defined as

\[
\max_{e \sim E(G)} \left[ d_{EMD}(A(G), A(G - e)) \right]
\]
Our Results (1)

❖ Theorem: \((1 + \epsilon)\)-approximation algorithm to the maximum matching problem with worst-case edge sensitivity \(O_\epsilon(1)\).

❖ Theorem: Deterministic algorithm for maximal matching with worst-case sensitivity \(\Delta^x\), where \(x = O(6^\Delta + \log^* n)\), for graphs with vertex degree bounded by \(\Delta\).
Our Results (1)

❖ Theorem: \((1 + \epsilon)\)-approximation algorithm to the maximum matching problem with worst-case edge sensitivity \(O_\epsilon(1)\).

❖ Theorem: Deterministic algorithm for maximal matching with worst-case sensitivity \(\Delta^x\), where \(x = O(6^\Delta + \log^* n)\), for graphs with vertex degree bounded by \(\Delta\).
Maximum Matching

- \((1 + \epsilon)\)-approximation algorithm for maximum matching with average sensitivity \(O\left(\left(\frac{OPT}{\epsilon^3}\right)^{1/(1+\Omega(\epsilon^2))}\right)\) \([VY21]\)

- What about worst-case sensitivity? Previous example shows greedy algorithm can have sensitivity \(\Omega(n)\)
Maximal Matching

❖ First idea: Randomized greedy for maximal matching
❖ The “expected” deleted/added edge appears halfway through the ordering of edges
❖ Intuition: Either most of the edges in the matching occur early in the ordering (dense graph) OR the deleted/added edge does not affect edges in the matching (sparse graph)
Randomized greedy for maximal matching

- Let $\pi(e)$ be the rank of $e$ in a (fixed) random permutation
- $I_{\pi}(e):= \text{Neighbors of } e \text{ with smaller rank than } e$
- $e$ is in the maximal matching $M$ iff no edges of $I_{\pi}(e)$ are in $M$
- Consider the set $S$ of edges that must be modified to maintain the invariant if $e$ is moved to the beginning of $\pi$
- Unless $\pi(e) = \min \pi(S)$, then no edges need to be modified
- $\pi(e) = \min \pi(S)$ with probability $\frac{1}{|S|}$
- Expected number of modified edges $O(1)$
Randomized greedy for maximal matching

- Let $\pi(e)$ be the rank of $e$ in a (fixed) random permutation $\pi$.
- $I_{\pi}(e)$: Neighbors of $e$ with smaller rank than $e$.
- $e$ is in the maximal matching $M$ iff no edges of $I_{\pi}(e)$ are in $M$.
- Consider the set $S$ of edges that must be modified to maintain the invariant if $e$ is moved to the beginning of $\pi$.
- Unless $\pi(e) = \min \pi(S)$, then no edges need to be modified.
- $\pi(e) = \min \pi(S)$ with probability $\frac{1}{|S|}$.
- Expected number of modified edges $O(1)$.

Similar idea used for maximal independent set in the dynamic distributed setting [CHK16]
Step 1: Maximal Matching

❖ First idea: Randomized greedy for maximal matching with worst-case sensitivity $O(1)$
❖ From maximal matching to $(1 + \epsilon)$-approximation algorithm for maximum matching?
❖ Modify the layered graph in the multi-pass streaming algorithm of McGregor [M05]
Step 2: Layered Graph

- Creates a graph with $\ell + 1$ layers to simultaneously sample a large number of augmenting paths of length $\ell$
Step 2: Layered Graph

- Creates a graph with \( \ell + 1 \) layers to simultaneously sample a large number of augmenting paths of length \( \ell \)

- Terminates in roughly \( \frac{1}{\varepsilon} \) iterations to achieve \((1 + \varepsilon)\)-approximation algorithm for maximum matching

- Samples roughly \( K = \left(\frac{1}{\varepsilon}\right)^{2^{O\left(\frac{1}{\varepsilon}\right)}} \) augmenting paths

- Worst-case sensitivity \( O(3^K) = O_{\varepsilon}(1) \)

- Runtime: \( O((n + m)K) \)
Our Results (1)

❖ Theorem: \((1 + \epsilon)\)-approximation algorithm to the maximum matching problem with worst-case edge sensitivity \(O_\epsilon(1)\).

❖ Theorem: Deterministic algorithm for maximal matching with worst-case sensitivity \(\Delta^x\), where \(x = O(6^\Delta + \log^* n)\), for graphs with vertex degree bounded by \(\Delta\).
Deterministic Algorithm

- Ingredient #1: Deterministic local computation algorithm (LCA) by Cole and Vishkin for $6^\Delta$-coloring of a graph with degree $\Delta$ using $O(\Delta \log^* n)$ queries [CV86]
- Partitions the graph into forests and assigns each node a color based on the LSB that differs between the node ID and its parent ID
- Fact #1: Deterministic LCA only queries vertices within distance $O(\log^* n)$
Deterministic Algorithm

- Ingredient #2: Framework by Parnas and Ron [PR07] that simulates local distributed algorithms using deterministic LCA using $\Delta^0(c)$ probes
- Iterate through colors and add each edge to maximal matching $M$ if no adjacent edge is already in $M$
- Fact #2: Framework only queries vertices within distance $O(6^\Delta)$
Deterministic Algorithm

❖ Fact #1: Deterministic LCA only queries vertices within distance $O(\log^* n)$
❖ Fact #2: Framework only queries vertices within distance $O(6^\Delta)$
❖ Sensitivity analysis: At most $O\left(\Delta 6^\Delta + \log^* n\right)$ queries in the LCA can be affected by a single deletion/addition
Our Results (1)

- Theorem: \((1 + \epsilon)\)-approximation algorithm to the maximum matching problem with worst-case edge sensitivity \(O_\epsilon(1)\).
- Theorem: Deterministic algorithm for maximal matching with worst-case sensitivity \(\Delta^x\), where \(x = O(6^\Delta + \log^* n)\), for graphs with vertex degree bounded by \(\Delta\).
Our Results (2)

- Theorem: Any deterministic constant-factor approximation algorithm for the maximum matching problem has worst-case edge sensitivity $\Omega(\log^* n)$.

- Theorem: For a graph with edge weights bounded by $W = \text{poly}(n)$ and a trade-off parameter $\alpha$, there exists an algorithm that outputs a $4\alpha$-approximation to the maximum weighted matching in $O(m \log_\alpha n)$ time. For $\alpha = 2$, the algorithm has weighted sensitivity $O(W \log n)$ and normalized weighted sensitivity $O(\log n)$. For $\alpha > 2$, the algorithm has weighted sensitivity $O(W)$ and normalized weighted sensitivity $O(1)$. 
Future Directions

❖ Worst-case analysis of monotone submodular maximization by McMeel and Yoshida [MY20]

❖ Worst-case (normalized) weighted sensitivity for other graph problems?

❖ Constant factor approximation algorithm for maximum weighted matching with “low” worst-case sensitivity?

❖ Can we get \((1 + \epsilon)\)-approximation algorithm for maximum weighted matching with “low” worst-case (normalized) weighted sensitivity?
thank you