Redundancy-Aware Maximal Cliques

Jia Wang    James Cheng    Ada Wai-Chee Fu
Chinese University of Hong Kong
Maximal Cliques

• Input
  • Undirected graph $G = (V, E)$

• Maximal cliques
  • Clique: vertex set of a complete subgraph
  • Maximal: adding vertex makes it no clique
Classic problem

- MCE (Maximal Clique Enumeration)
  - exhaustive: finding set of ALL maximal cliques
Classic algorithm

• Algorithm: recursive search
  • Maintain *current clique* $C$ & *candidate set* $T$
  • Recursion:
    • select vertex in $T$, add to $C$ (a branch)
    • update $T$
Classic algorithm

• Example

![Diagram of a graph with nodes a, b, c, d, e, f, g, showing current clique and candidates.](image)
Problems of MCE

• **Usability**
  - overwhelmingly **large output**
  - cliques less useful due to **overlap**
  - full MCE no good or necessary
    - anomaly detection, exploration...

• **Speed**
  - exhaustive search of large space
  - can be *exponentially* many
Problems of MCE

• Instead we desire
  • I: compact representation – each result meaningful
  • II: preserved information – widely covering
  • I & II: a good **summary**, e.g.:
## Notations

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>$M$</strong></td>
<td>Set of all maximal cliques</td>
<td></td>
</tr>
<tr>
<td><strong>$S$</strong></td>
<td>a subset of $M$ (summary)</td>
<td></td>
</tr>
<tr>
<td><strong>$C/C'$</strong></td>
<td>current/last maximal clique</td>
<td></td>
</tr>
<tr>
<td><strong>$r$</strong></td>
<td>$\frac{</td>
<td>c \cap c'</td>
</tr>
</tbody>
</table>
A new notion

• Clique visibility
  • visibility of $C$ given $S$: max ratio $r$ of $C$ covered by any $C'$ in $S$
  • Denoted by $\text{vis}(C)$

• $\tau$-visible summary
  • A summary $S$ such that $\text{vis}(C) \geq \tau$ for each $C$ in $M$

• Problem: $\tau$-visible MCE
  • find a small $\tau$-visible summary $S$ of $M$

Have enabled redundancy reduction. Possibly faster too?

$\text{vis}(\{a, b, c, d, f\}) = 4/5$

$\text{vis}(\{b, d, f, g\}) = 3/4$

$S = \{\{a, b, d, e, f\}\}$
A naïve implementation

• In classic MCE
  • $S$: summary of cliques so far
  • $C$: compare to each maximal clique in
  • $\rightarrow$ add $C$ to $S$: if no redundancy
  • $\rightarrow$ discard $C$: if much overlap with any $C'$ in $S$

• Overhead
  • $O(T_{MCE} + |M| \times |S|)$
  • costly computation
Main idea

• Characterizing search process
  • nearby cliques $C$ and $C'$ (leafs) correlated
    • have common ancestors in search tree
  • $C \sim C'$ when close in search tree
For efficiency – first step

• Glancing at last one
  • discard most redundancy in one shot

generated sequence of cliques
For efficiency – first step

• Summary as a sample
  • retain with probability $s(r)$: decreases with $r$
  • cliques as data points, $r$ as slope
  • a perspective: analogy to importance sampling

[Diagram showing generated sequence of cliques with high and low $s(r)$]
For efficiency – first step

• Choice of $s(r)$
  • To meet visibility requirements
  • Choose: $s(r) = \frac{(1-r)(2-\tau)}{2-r-\tau}$
  • Claim: $E[\text{vis}(C)] \geq \tau$ for all $C$
For efficiency – a further step

• Detected redundancy when \textit{fully} grown
• Now: earlier with \textit{foresight}

• At inner node
  • lower bound $r$
  • prune whole branch with large $r$

foretell $r$ at least how much for any $C$ starting here?

At most $y$ vertices in $C'$ for $C$ (forming a clique)

$t$ more vertices to $C$

Then at least $y - t$ vertices in $C \cap C'$
For efficiency – a further step

• Sampling search branch
  • Want: guarantee still holds
    • for expected visibility
  • Need: maintain $\Pr[\text{final retaining prob.}] \geq s(r)$
  • How: set $\Pr[\text{sample a branch}] = \sqrt[l]{s(\tilde{r})}$
    • $l$: upper bound of branch depth
    • $\tilde{r}$: lower bound of $r$
Applying the summary

• Feed other computations
  • A succinct input
  • Example: top-$k$ results
    • Approx. ratio using $S$: $\tau(1 - 1/e)$

Set of all maximal cliques

$\tau$-visible summary

Applications

top-k retrieval
exploration
visualization
...

MCE

Summary
Applying the summary

• Discovering clique space
  • Proposal: explore interactively

- All maximal cliques, M
- summary of M, Top-k if too many
- Interesting region Z
- Cliques on Z and its neighbors, M’
- Summary of M’
On real world networks

• Datasets

<table>
<thead>
<tr>
<th></th>
<th>Blog</th>
<th>Skitter</th>
<th>Wiki</th>
<th>Patent</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>V</td>
<td>$</td>
<td>990K</td>
<td>1.7M</td>
</tr>
<tr>
<td>$</td>
<td>E</td>
<td>$</td>
<td>6.6M</td>
<td>11.1M</td>
</tr>
<tr>
<td>$</td>
<td>M</td>
<td>$</td>
<td>11.2M</td>
<td>18.3M</td>
</tr>
</tbody>
</table>

# of all maximal cliques
On real world networks

- Summary size
  - slimmed output
  - sharp drop from $\tau = 1$ to $\tau = 0.9$

$\sim50$ times smaller
On real world networks

- Running time
  - Reduced time
  - Especially from \( \tau = 1 \) to \( \tau = 0.9 \)

![Graphs showing time halved from \( \tau = 1 \) to \( \tau = 0.9 \).]
On real world networks

- Top-$k$ reporting
  - using full result or summary
  - setting: $k = 20$, $\tau = 0.7$
  - result: small quality loss, greatly faster

<table>
<thead>
<tr>
<th></th>
<th>Blog</th>
<th>Skitter</th>
<th>Wiki</th>
<th>Patent</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{samp}$</td>
<td>822</td>
<td>1205</td>
<td>462</td>
<td>173</td>
</tr>
<tr>
<td>$Q_{all}$</td>
<td>826</td>
<td>1214</td>
<td>464</td>
<td>174</td>
</tr>
<tr>
<td>$T_{samp}$</td>
<td>1.38</td>
<td>4.02</td>
<td>8.59</td>
<td>0.7</td>
</tr>
<tr>
<td>$T_{all}$</td>
<td>28.4</td>
<td>57.5</td>
<td>197</td>
<td>8.9</td>
</tr>
</tbody>
</table>

→ Quality by summary
→ Quality by all cliques
→ Time by summary
→ Time by all cliques
Wrapping up

• Tradeoff
  • completeness \(\rightarrow\) compactness & usability & time

• Approaches
  • notion of \(\tau\)-visible summary
  • fast redundancy detection
  • early pruning
  • summary as a sample

• Applications
  • exploration, top-\(k\), and more