Technical Perspective of Efficient Directed Densest Subgraph Discovery

Yufei Tao
Chinese University of Hong Kong
taoyf@cse.cuhk.edu.hk

Consider a directed graph \( G = (V, E) \). Given two (possibly overlapping) subsets \( S, T \subseteq V \), denote by \( E(S, T) \) the set of edges \( (s, t) \in E \) with \( s \in S \) and \( t \in T \) and by \( G(S, T) \) the subgraph with \( S \cup T \) as the vertex set and \( E(S, T) \) as the edge set. The density of \( G(S, T) \) equals \( |E(S, T)|/\sqrt{|S||T|} \). The directed densest subgraph (DDS) problem is to return a pair \( (S, T) \) maximizing the density of \( G(S, T) \).

The problem is useful in graph mining because dense subgraphs often represent patterns deserving special attention. They could indicate, for example, an authoritative community in a social network, a building brick of more complex biology structures, or even a type of malicious behavior such as spamming. See [1, 3] and the references therein for an extensive discussion on the applications of DDS.

Previous research has led to non-trivial findings on DDS. The problem is solvable in polynomial time: the fastest algorithm known today has a time complexity \( O(|V|^3|E| \log |V|) \) [2]. One could attain better efficiency by settling for an approximate output. Specifically, if the optimal density achievable is \( \rho^* \), a pair \( (S, T) \) makes a \( \epsilon \)-approximate solution if \( G(S, T) \) has density at least \( \rho^*/\epsilon \). It was claimed in [2] that a 2-approximate solution could be found in \( O(|V| + |E|) \) time.

The 2-approximate result of [2], unfortunately, turned out to be incorrect. Ma et al. — the authors of the paper I am introducing — pointed out a loophole in the argument of [2], which (in my opinion) was rather difficult to discern even in retrospect. They went further to dispove the claim by constructing a class of counterexamples. It remains unclear whether the issue can be fixed with the O(|V| + |E|) complexity restored. Currently, the best fix available necessitates \( O(|V| \cdot (|V| + |E|)) \) time [3].

As a partial remedy, Ma et al. developed a 2-approximate algorithm with running time \( O(\sqrt{|E|} \cdot (|V| + |E|)) \), which improves \( O(|V| \cdot (|V| + |E|)) \) as long as \( |E| = o(|V|^2) \). They achieved the purpose by establishing a connection between DDS and the \([x, y]\)-core, a new concept that can be regarded as the directed counterpart of \( k\)-core. The connection then led to an elegantly simple algorithm.

Formally, call a subgraph \( G(S, T) \) an \([x, y]\)-core if every vertex in \( S \) has at least \( x \) outgoing edges and every vertex in \( T \) has at least \( y \) incoming edges. Define \( xy \) as the product of \( G(S, T) \). Denote by \( G(S^*, T^*) \) the subgraph of the maximum product. Ma et al. proved that the density of \( G(S^*, T^*) \) is at least \( \rho^*/2 \). In other words, \( G(S^*, T^*) \) serves as 2-approximate solution to the DDS problem.

Now, it remains to compute \( G(S^*, T^*) \). Suppose that \( G(S^*, T^*) \) is an \([x^*, y^*]\)-core where the chosen values of \( x^* \) and \( y^* \) maximize \( x^* \cdot y^* \). Assume, for the time being, \( x^* \leq y^* \). Two observations are immediate. First, as \( G(S^*, T^*) \) has at least \( x^* \cdot y^* \) edges, \( x^* \cdot y^* \leq |E| \) and, hence, \( x^* \leq \sqrt{|E|} \). Second, by the definition of \( G(S^*, T^*) \), no \([x, y]\)-cores can exist for any \( y > y^* \).

These observations suggest the following strategy for discovering \( G(S^*, T^*) \): Fix an integer \( x \in [1, \sqrt{|E|}] \) and find the maximum \( y \) such that an \([x, y]\)-core exists. We can accomplish this in \( O(|V| + |E|) \) time in a way reminiscent of computing an undirected graph’s core number:

\begin{enumerate}
  \item \( y_x = 0 \)
  \item \textbf{repeat}
  \item remove the vertices with out-degree less than \( x \)
  \item \( y_{min} = \) the smallest in-degree of the remaining vertices
  \item if \( y_{min} > y_x \) then \( y_x = y_{min} \)
  \item remove an arbitrary vertex with in-degree \( y_{min} \)
  \item until no vertices are left
\end{enumerate}

Besides \( y_x \), one can also return the corresponding \([x, y_x]\)-core by modifying the pseudocode slightly. Executing the code for all \( x \in [1, \sqrt{|E|}] \) produces (at most) \( \sqrt{|E|} \) subgraphs, among which the one with the maximum product is \( G(S^*, T^*) \). The assumption \( x^* \leq y^* \) can be removed by considering \( y^* \leq x^* \) in a symmetric manner.

The paper of Ma et al. makes further contributions by (i) showing how to use the \([x, y]\)-core technique to accelerate the state-of-the-art exact DDS algorithm [2], and (ii) demonstrating the performance of proposed algorithms through an extensive experimental evaluation. This is a beautiful paper that fuses theory and practice very nicely.

1. REFERENCES