Hyper-Flow Diffusion

Kimon Fountoulakis\textsuperscript{1}, Pan Li\textsuperscript{2}, Shenghao Yang\textsuperscript{1}

\textsuperscript{1}University of Waterloo \textsuperscript{2}Purdue University
Diffusion on graphs

Diffusion on a graph is the process of spreading a given initial mass from some seed node(s) to neighbor nodes using the edges of the graph.

Applications include recommendation systems, node ranking, community detection, social and biological network analysis, etc.
Diffusion on hypergraphs
Diffusion on hypergraphs

But how to diffuse mass within a hyperedge?
Flow of mass within a hyperedge

Flow on a graph edge

\[ v_1 \] sends 2 units of mass to \( v_2 \)
\( v_2 \) receives 2 units of mass from \( v_1 \)

Flow on a hyperedge

\( \{v_1\} \) sends 1 unit of mass to \( \{v_2, v_3, v_4\} \)
\( \{v_2\} \) receives 6 units of mass from \( \{v_1, v_3, v_4\} \)
\( \{v_1, v_3\} \) sends 4 units of mass to \( \{v_2, v_4\} \)
\( \{v_1, v_2\} \) receives 5 units of mass from \( \{v_3, v_4\} \)

...
Diffusion on hypergraphs

Hyper-Flow Diffusion diffuses mass according to hyperedge flows
Modelling higher-order relations

We want hyperedge flows to reflect nontrivial higher-order relations …

Primal-dual relations enable us to look at hyperedge cuts:

How do we treat differently from?
Distinct ways to cut a hyperedge may have different penalties.

$w_e : 2^{\{v_1,v_2,v_3,v_4\}} \to \mathbb{R}_+$ is a submodular set function, where $w_e(S)$ specifies the cost of splitting $e$ into $S$ and $e \setminus S$. 
Modelling higher-order relations

Different cut penalties lead to different flow dynamics.

\[ w_e(\{v_1, v_2\}) = 0 \]

\( w_e : 2^{\{v_1, v_2, v_3, v_4\}} \rightarrow \mathbb{R}_+ \) is a submodular set function, where \( w_e(S) \) specifies the cost of splitting \( e \) into \( S \) and \( e \setminus S \).

No flow of mass is allowed between \( \{v_1, v_2\} \) and \( \{v_3, v_4\} \).
Application

Diffusion leaves excess mass on nodes. This induces an ordering of nodes.

We use the ordering for **node ranking** and **local clustering**.

It achieves the first edge-size-independent Cheeger-type approximation guarantee for local hypergraph clustering.

Empirically, the running time depends only on the output size.
Empirical results

Node-ranking and local clustering results on a Florida Bay food network.

<table>
<thead>
<tr>
<th>Method</th>
<th>Query: Raptors</th>
<th>Query: Gray Snapper</th>
<th>Clustering F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-HFD</td>
<td>Epiphytic Gastropods, Detriti. Gastropods</td>
<td>Meiofauna, Epiphytic Gastropods</td>
<td>Prod. 0.69, Low 0.47, High 0.64</td>
</tr>
<tr>
<td>C-HFD</td>
<td>Epiphytic Gastropods, Detriti. Gastropods</td>
<td>Meiofauna, Epiphytic Gastropods</td>
<td>0.67 0.47 0.64</td>
</tr>
<tr>
<td>S-HFD</td>
<td>Gruiformes, Small Shorebirds</td>
<td>Snook, Mackerel</td>
<td><strong>0.69 0.62 0.84</strong></td>
</tr>
</tbody>
</table>

- S-HFD uses specialized submodular cut-cost shown on the left.
- The example shows that general submodular cut-cost can be necessary.
- HFD is the only local diffusion method that works with general submodular cut-costs.
Hyper-Flow Diffusion

For more details:

Please see our paper **Local Hyper-Flow Diffusion**, NeurIPS 2021

Thank you!