Hypergraph Partitioning on GPUs and Image Classification

Hyunsu Cho '15
Faculty Advisors: Peter A. Yoon, Department of Computer Science
Lin Cheng, Department of Engineering

Summary

- Goal: classify pictures into two categories
- **Hypergraph partitioning** is a good algorithm for image classification
- Hypergraphs capture similarities among neighboring pictures
- Use GPUs to accelerate hypergraph partitioning
- GPUs have lots of small workers and thus require a different programming model (many-core architecture)

Image Classification

- Want to **classify** a set of pictures into two groups
- Similar pictures should belong to same category
- **Hypergraph**: compare multiple pictures at a time
group nearest neighbors into **hyper-edges**

Image Classification Pipeline

1. Read pictures
2. Extract **local features**
3. Pool local features into summary vectors
4. Measure pairwise distances
5. Form hyper-edges by grouping nearest neighbors
6. **Partition the hypergraph** (most expensive)

Hypergraph partitioning

- Goal: **partition hypergraph to minimize edge cut**
- Some hyper-edges are stronger than others
- **Does not require prior training**

Multi-level Paradigm

- High computational cost to partition hypergraphs
- **Coarsening**: reduce # of nodes by fusing them
- Solve the reduced problem, and then estimate solution for the original problem
- Reduce recursively until # nodes become manageable

Classification Results

<table>
<thead>
<tr>
<th></th>
<th>Support Vector Machine (needs training)</th>
<th>Hypergraph partitioning (does not need training)</th>
</tr>
</thead>
<tbody>
<tr>
<td>98.03 %</td>
<td>99.24 %</td>
<td></td>
</tr>
<tr>
<td>91 %</td>
<td>80 %</td>
<td></td>
</tr>
</tbody>
</table>