Algorithms for graph compression

Arindam Pal
IIT Delhi / Yahoo! Research

Problem description

- Weighted directed graph $G = (V,E)$.
- Weight of edge e is w_e.
- Nodes are numbered from 1 to n.
- Represented in the adjacency list format.
- Each element in the adjacency list requires $\log n$ bits to store.
- Goal: Minimize the number of bits needed to represent the graph.

Reference encoding

- Adjacency list of a node shares elements with the adjacency lists of other nodes.
- Shared nodes can be specified implicitly.
- Suppose x shares some neighbors with y.
- Create a bit-vector for y of length $d(y)$.
- Set the bit to 1 for those nodes in $N(y)$ which are shared with $N(x)$.

Illustrative example

Node	Adjacency list
1	2, 7, 13, 25
2	3, 4, 5, 7, 13, 20
3	2, 3, 5, 20, 25, 31
4	2, 3, 5, 7, 31
…	…

Example continued…

Node	Reference nodes	Shared lists	Exclusive list
1	0	EMPTY	2, 7, 13, 25
2	1	0110	3, 4, 5, 20
3	1, 2	1001, 101001	31
4	1, 2, 3	1000, 101100, 000001	EMPTY
…	…	…	…

Algorithm

- $c(x,y) = d(y) + (\,|N(x) – N(y)|\, + 1)\log n$.
- $c(x) = d(x)\log n$.
- Goal is to find an encoding of minimum cost.
- Construct another graph H having all nodes in G plus a new root node r.
- Cost of edge between x and y is $c(x,y)$.
- Cost of edge between x and r is $c(x)$.
- Find a directed MST T in H.
- Use the edges in T to encode x.

Approximate representation

- Construct a graph $G_\epsilon = (V,E_\epsilon)$ such that $|E_\epsilon|$ is minimized with the following property.

$$d_\epsilon(v) \geq (1 - \epsilon)d(v), \text{ for all } v \in V$$

- Equivalent to finding a set of edges $E' \subseteq E$ in G with $|E'|$ maximum such that in $G' = (V,E')$,

$$d'(v) \leq \epsilon d(v), \text{ for all } v \in V$$

- Can be modeled as an integer program.
- Using randomized rounding and Chernoff bounds, optimum can be found with high probability.

Integer program

$$\textbf{maximize} \sum_{e \in E} x_e$$

$$\textbf{such that} \sum_{e \in \delta(v)} x_e \leq \epsilon d(v) \quad \forall v \in V$$

$$x_e \in \{0, 1\} \quad \forall e \in E$$

Using multiple reference nodes

- Let $L(x)$ be the set of reference nodes used to cover the nodes in $N(x)$.

$$\textbf{minimize} \sum_{x \in V} c(x),$$

$$\textbf{where } c(x) = \left(|L(x)| + \left|N(x) \setminus \bigcup_{v \in L(x)} N(v)\right|\right) \log n$$

$$+ \sum_{v \in L(x)} d(v).$$

Open questions

- How to choose an ordering in which the nodes should be listed?
- Given an ordering, for each node x what nodes should be included in $L(x)$?
- Given x and $L(x)$, how to cover the nodes in $N(x)$ so that $c(x)$ is minimized?
- Can be modeled as a directed MST problem in a hypergraph.

Bounding number of dereferencing

- Dereferencing a node may lead to cascading dereferencing of other nodes.
- We may set a bound on the number of nodes we have to dereference.
- Equivalent to computing the directed MST with a depth bound, which is NP-hard to approximate.
- Even for depth 2, the facility location problem can be reduced to this problem.
- For undirected graphs, there is a randomized algorithm that computes a spanning tree of depth at most k, whose expected cost is $O(\log n)$ times the cost of the MST of depth at most k.

Questions?

Thank you!