Table of Contents


```
classdef Hypergraph
    %HYPERGRAPH Class to store a hypergraph, summarize it, make
    %   computations on it, and plot it.

    % This class treats rows of the incidence matrix as nodes and the
    % columns as hyperedges.
    properties
        IM (:,:) % incidence matrix
        ES (:,:) % edge set for uniform hypergraphs
        edgeWeights
        nodeWeights
    end

    methods

        function obj = Hypergraph(nameValueArgs)
            %HYPERGRAPH Construct an instance of this class.
            %   Takes in any 2D array representing an incidence matrix and
            %   stores it in sparse format. Also can store the hyperedge
            %   set for uniform hypergraphs.
            arguments
                nameValueArgs.H = sparse(1);
                nameValueArgs.edgeSet = [];
                nameValueArgs.edgeWeights = 0;
                nameValueArgs.nodeWeights = 0;
            end
            obj.IM = sparse(nameValueArgs.H);
            obj.ES = nameValueArgs.edgeSet;
            if nameValueArgs.edgeWeights == 0
                nameValueArgs.edgeWeights = ones(size(obj.IM, 2), 1);
            end
            if nameValueArgs.nodeWeights== 0
                nameValueArgs.nodeWeights = ones(size(obj.IM, 2), 1);
            end
            obj.edgeWeights = nameValueArgs.edgeWeights;
            obj.nodeWeights = nameValueArgs.nodeWeights;
        end
```

Summarization

```
        function m = numRows(obj)
            %NUMROWS Get the number of rows in the incidence matrix.
```

```
    m = size(obj.IM, 1);
end

function n = numCols(obj)
    %NUMCOLS Get the number of columns in the incidence matrix.
    n = size(obj.IM, 2);
end

function d = density(obj)
    %DENSITY Gets the density of the underlying incidence matrix.
    %If this is less than (m*(n-1)-1)/2, then the matrix is so
    %dense that storing it in CSC format takes up more memory than
    %dense format.
    d = nnz(obj.IM)/(obj.numCols * obj.numRows);
end

function t = CSCThreshold(obj)
    %CSCTHRESHOLD Gets the density threshold over which it saves
    %space to store the matrix in dense format.
    m = obj.numRows;
    n = obj.numCols;
    t = (m*(n-1)-1)/(2*m*n);
end

function sz = edgeSizes(obj)
    %EDGESIZES Get the number of nodes in each edge.
    sz = sum(obj.IM, 1);
end

function dg = nodeDegrees(obj)
    %NODEDEGREES Get the degree of each node.
    dg = sum(obj.IM, 2);
end

% fun decls

% Returns the s-connected components of the hypergraph.
% s: the minimum connecting edge size. When s=1, this function
%   returns the connected components of the clique expansion.
% outputForm {"vector" (default), "cell"}: arg to MATLAB's
%   conncomp function. Specifies the output form of the connected
%   components.
[bins, binSize]  = sConnectedComponents(obj, s, outputForm)

r = sRadius(obj, s)

d = sDiameter(obj, s)
```

Translation

```
function D = getDense(obj)
    %GETDENSE Returns a densely-stored copy of the incidence
    %matrix.
```

```
            D = full(obj.IM);
        end
```

Computation

Visualization

```
    end
end
```

Published with MATLAB® R2021b