
Decompositions.TensorDecomp.adjacencyTensor

Table of Contents

Syntax	1
Input	1
Output	1
Disclaimer	1
Code	1

Syntax

Input

Output

Disclaimer

Code

```
function adjTensor = adjacencyTensor(HG)
% H is the incidence matrix of a hypergraph
% Return the generalized adjacency tensor of the hypergraph
H = HG.IM;
eW = HG.edgeWeights;

numNodes = size(H, 1);
order = max(sum(H, 1));
adjTensor = zeros(numNodes^order, 1);

for i = 1:size(H, 2)
    hyperEdge = find(H(:, i)>0);

    if length(hyperEdge) == 1
        dummyEdges = nmultichoosek(hyperEdge, order);
        coefficient = 1;
    elseif length(hyperEdge) == order
        dummyEdges = hyperEdge';
        coefficient = 1/factorial(order-1);
    else
        dummyEdges = nmultichoosek(hyperEdge, order);
        dummyEdges = dummyEdges(arrayfun(@(x) all(ismember(hyperEdge',
dummyEdges(x, :))), 1:size(dummyEdges, 1))', :));
        dummyEdgesCell = mat2cell(dummyEdges, ones(1, size(dummyEdges, 1)),
size(dummyEdges, 2));
        numCounts = cellfun(@(x) histcounts(x), dummyEdgesCell, 'Uni', 0);
```

```
        numCounts = cell2mat(numCounts);
        coefficient = length(hyperEdge)/sum(factorial(order)./
prod(factorial(numCounts), 2));
    end

    coefficient=coefficient*eW(i);

    for m = 1:size(dummyEdges, 1)
        p = unique(perms(dummyEdges(m, :)), 'rows');
        for t = 1:size(p, 1)
            adjTensor(indexMapping(ones(1, order)*numNodes, p(t, :))) =
coefficient;
        end
    end
end
adjTensor = reshape(adjTensor, ones(1, order)*numNodes);
end

function combs = nmultichoosek(values, k)
n = numel(values);
combs = bsxfun(@minus, nchoosek(1:n+k-1,k), 0:k-1);
combs = reshape(values(combs),[],k);
end

function linearIdx = indexMapping(sz, subscript)
firsrtSub = subscript(1);
subscript = subscript(2:end)-1;
szProd = zeros(1, length(sz)-1);
for i = 1:length(sz)-1
    szProd(i) = prod(sz(1:i));
end
linearIdx = firsrtSub+sum(subscript.*szProd);
end
```

Published with MATLAB® R2021b