Shanon Fano Encoding ALgorithm (Binary)

Anubhav Rathore

Input: Probabilities of Symbols

Output: Encodings for each symbol, with higher probability symbol having shorter code length and low probabability symbols having longer code length

Contents

Defaults

```
clearvars;
clear all;
clc;
```

Inputs, Variables and Constants

```
probabs = [0.1 0.2 0.1 0.3 0.1 0.1 0.1]; % Probabilities of Symbols
N = length(probabs);
codes = repmat({''}, 1, N);
[SortedProbabs, order] = sort(probabs, "descend"); %Order keeps indexing of symbols as original
idx = order; %Original Indices for Encoding
```

FindBestSplit

```
function [bestK] = FindBestSplit(probabilityVector)
minDifference = inf;
bestK = 1;

    for k = 1:(length(probabilityVector)-1)
        leftSum = sum(probabilityVector(1:k));
        rightSum = sum(probabilityVector(k+1:end));
        diff = abs(leftSum - rightSum);
        if diff < minDifference
            minDifference = diff;
            bestK = k;
        end
    end
end
```

Binary Shanon-Fano Encoding

```
function codes = encoding(probabs, idx, codes)
    if isscalar(idx)
        return
    end

    if length(idx) == 2
        codes{idx(1)} = [codes{idx(1)} '0'];
        codes{idx(2)} = [codes{idx(2)} '1'];
        return
    end

    bestK = FindBestSplit(probabs);
    leftIdx  = idx(1:bestK);
    rightIdx = idx(bestK+1:end);

    for i = leftIdx
        codes{i} = [codes{i} '0'];
    end

    for i = rightIdx
        codes{i} = [codes{i} '1'];
    end

    codes = encoding(probabs(1:bestK), leftIdx, codes);
    codes = encoding(probabs(bestK+1:end), rightIdx, codes);
end
```

The following results follow two important self-built functions

```
codes = encoding(SortedProbabs, idx, codes);

Probabilities = probabs';
Codes   = string(codes)';

T = table(Probabilities, Codes);
disp(T)
```

Published with MATLAB® R2024a