Clustering of an N-Cube

MTB Challenge - Summer Term 2018/19

Miloslav Čapek, Michal Mašek

March 12, 2019

1 Motivation

Imagine an N-dimensional cube with the unitary length of all the edges (also called as the Hamming graph $H(N,2)$). Imagine further you can select just $M \ll N$ vertices. What particular vertices do you pick to minimize the distance (counted in number of edges to pass through) to reach arbitrary (n-th) vertex.

The entire problem is alternatively defined as follows: Having a code word of N bit length and considering M distinct code words, what particular vertices should be chosen as pivots to minimize the maximum Hamming distance

$$h(\boldsymbol{g}_a, \boldsymbol{g}_b) = \|\boldsymbol{g}_a \oplus \boldsymbol{g}_b\| \in \mathbb{N}_0 \tag{1}$$

between an arbitrarily picked code word (out of the preselected pool of M code words) and any other word which can be generated by N bits? Explicitly, find a number $C(M,N)$, so that

$$C(M,N) = \min_{p \in \mathcal{P}} \max_{i \in \{1,\dots,2^N\}} \left\{ h\left(\boldsymbol{g}_p, \boldsymbol{g}_i\right) \right\}, \quad |\mathcal{P}| = M, \tag{2}$$

and determine associated pivots $\boldsymbol{g}_{p \in \mathcal{P}}$.

2 Task

Propose an algorithm and implement it effectively in MATLAB in order to solve this task for arbitrarily large N (from tens to hundreds) and M (tens). If not possible to solve it deterministically, propose a scheme which solves the problem as good as possible and implement it.

3 Example

An N-cube in $\mathbb{R}^3$ has vertices along (x,y,z), therefore $N = 3$. For two vertices ($M = 2$) selected as $\mathbf{g}_1 = [0,0,0]$ and $\mathbf{g}_2 = [1,1,1]$, the maximal distance to all vertices is equal to one. For a cube in $\mathbb{R}^4$ (tesseract) with $N = 4$ two pivots can be chosen to get maximal distance equal to two, see Figure 1.

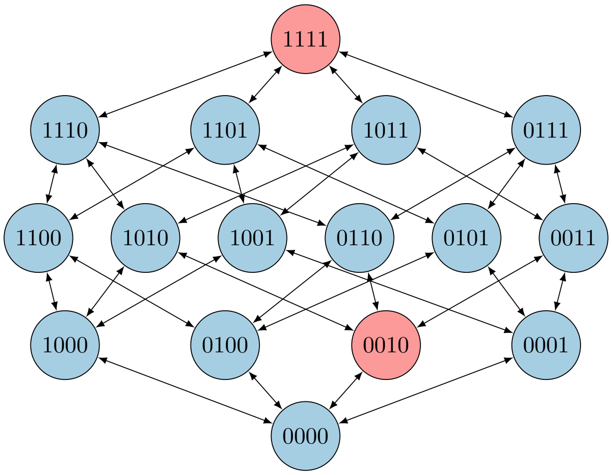

Figure 1: Representation of 2^N, $N = 4$ bit words. Two pivots, $\boldsymbol{g}_1 = 1111$ and $\boldsymbol{g}_2 = 0010$, are highlighted by red color. Hamming distance (1) from these to pivots to every other node is minimized to two.

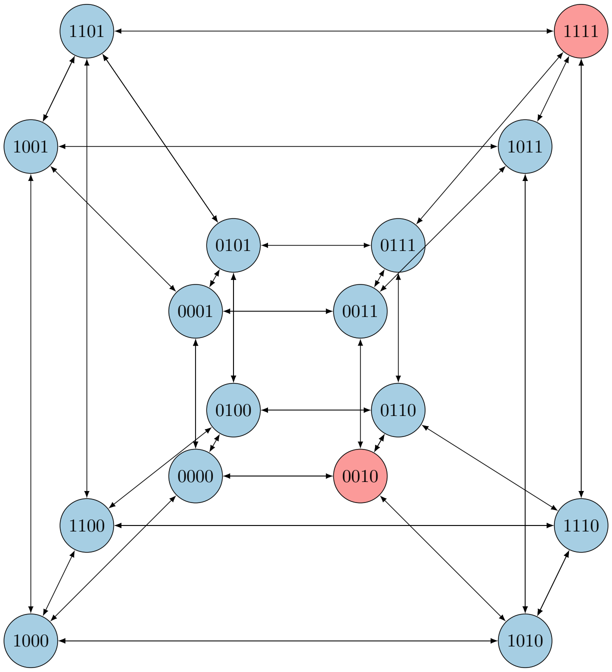

Figure 2: Projection of Fig. 1 in 4D cube.

Listing 1: `nCubePivots.m`

```
clear;
clc;
close all;

N = 4; % number of dimensions
M = 2; % number of pivots

D = dec2bin(0:(2^4-1))-'0'; % all bit words
P = nchoosek(1:2^N, M); % all permutations of pivots

% allocation
nP   = size(P, 1);
Dmax = nan(nP, 1);

% for each pivots combination compute distance between pivots and other
% nodes
for p = 1:nP
    Dmin = nan(2^N, M);
    % calculate the distance from each pivot
    for m = 1:M
        Dmin(:, m) = sum(xor(D, D(P(p,m),:)), 2);
    end
    % minimal distance to each node (no matter from which pivot)
    minDist = min(Dmin, [], 2);
    % maximal distance for this pivot combination
    Dmax(p) = max(minDist);
end

fprintf(1, 'All nodes are reachable in %1.0f moves\n', min(Dmax));

% Plot distance to the furthermost node for a given set of pivots
subplot(1,2,1);
stem(Dmax);
subplot(1,2,2);
hist(Dmax,1:N);
```

4 Criteria

4.1 General

- This project can be selected by unlimited number of students. However, no collaboration between students is expected.

- Project should be submitted including short documentation describing how the algorithm works.

- Like for regular projects, short presentation (couple of minutes) is expected.

- To be awarded with credits, it is enough to code a function dealing with the task.

- To participate in the competition (and have a chance to get some awards), all technical criterias must be fulfilled. In that case, the computational time required to get the correct solution will be measured on reference PC (Win10 + up-to-date edition on MATLAB). Fist three student with the fastest codes will be awarded. List of awards is attached at the end of the document.

- It is possible to always withdraw from the competition and select of of regular projects. This decision should be discussed with lecturers and their approval is required.

4.2 Technical

- The main function must have following header:
  `[pivots, maxD] = pivotDistribution(N, M);`

- where:
  - `N`: Length of bit words, `double, [1 x 1]`.
  - `M`: Number of pivots,`double, [1 x 1]`.
  - `pivots`: bit words of pivots, `double, [M x N]`
  - `maxD`: distance of the furthermost node for a givet set of pivots, `double, [1 x 1]`

- A brute-force method is shown in Listing 1.

- All information can be found at web page
  `http://cw.fel.cvut.cz/wiki/courses/b0b17mtb/projects/soutez`.

- The GUI is not required for this project.

- No toolboxes or external codes and libraries (dll, mex) are allowed.

5 List of Awards

Will be announced during the semester.