Algebra liniowa w analizie danych. Projekt 2. Kody liniowe.

Kacper Rodziewicz, Gaspar Sekula

10.05.2023

Zadanko 1.

Dowód. Wykażemy, że odległość Hamminga jest metryką.
Rozważmy funkcję $D : V \times V \to \mathbb{R}$ będącą odległością Hamminga, gdzie $V \subseteq \mathbb{K}^n$, $\mathbb{K}^n$ jest przestrzenią liniową nad ciałem $\mathbb{K}$ i V jest niepustym zbiorem.
To, że pierwsze dwa warunki są spełnione, jest oczywiste. Jednak gwoli profesjonalizmu, wyjaśnimy tę oczywistość. Weźmy dowolne $u, v, w \in V$.

(1) Równość $D(u, v) = 0$ zachodzi wtedy i tylko wtedy, gdy wszystkie współrzędne wektorów u i v są równe, co oznacza, iż $u = v$.

(2) Symetria wynika bezpośrednio z równoważności: u różni się od v o k współrzędnych wtedy i tylko wtedy, gdy v różni się od u o k współrzędnych.

(3) Wykażemy, że $D(u, w) \leq D(u, v) + D(v, w)$. Niech $i \in [n]$. Możliwe są dwa przypadki:

- (🐦) $u_i \neq w_i$ Możliwe są trzy przypadki:
  - (⚀) $u_i \neq v_i \land v_i \neq w_i$
  - (⚁) $u_i = v_i \land v_i \neq w_i$
  - (⚂) $u_i \neq v_i \land v_i = w_i$

  Przypadek $u_i = v_i \land v_i = w_i$ nie zachodzi, bowiem wówczas $u_i = w_i$.
- (🦴) $u_i = w_i$ Możliwe są dwa przypadki:
  - (⚀) $u_i = v_i \land v_i = w_i$
  - (⚁) $u_i \neq v_i \land v_i \neq w_i$

  Przypadki $u_i = v_i \land v_i \neq w_i$ oraz $u_i \neq v_i \land v_i = w_i$ nie zachodzą, gdyż $u_i = w_i$.

Zauważmy, że licząc odległość Hamminga, iterujemy po współrzędnych od 1 do n. Gdy wartość $D(u, w)$ rośnie o 1, to wartość $D(u, v) + D(v, w)$ rośnie o 1 lub 2, podobnie gdy wartość $D(u, w)$ nie zmienia się, to wartość $D(u, v) + D(v, w)$ rośnie o 0 lub 2. Z tych obserwacji wynika, że $D(u, w) \leq D(u, v) + D(v, w)$.

Toteż z (1), (2) i (3) odległość Hamminga jest metryką.

□

Zadanko 2.

Dowód. Niech $\mathcal{B} = (e_1, e_2, ..., e_k)$ będzie bazą $\mathcal{C}$, $\mathbb{G}$ będzie macierzą generującą (n,k)-kodu liniowego $\mathcal{C}$ nad $\mathbb{K}$, powstałą z bazy $\mathcal{B}$ oraz $v = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{pmatrix}$, $a_i \in \mathbb{K}$, $i \in [k]$. Należy wykazać, że $w = (v^T \cdot \mathbb{G})^T \in \mathcal{C}$.

Mamy więc

$$w = (v^T \cdot \mathbb{G})^T = \mathbb{G}^T \cdot v = (e_1|e_2|...|e_k) \cdot v = a_1 \cdot e_1 + a_2 \cdot e_2 + ... + a_k \cdot e_k,$$

gdzie $a_1 \cdot e_1 + a_2 \cdot e_2 + ... + a_k \cdot e_k$ jest kombinacją liniową wektorów z bazy $\mathcal{B}$, stąd $w \in \mathcal{C}$. $\square$

Zadanko 3.

Dowód. Wykażemy, że dla dowolnego (n,k)-kodu liniowego $\mathcal{C}$ nad skończonym ciałem $\mathbb{K}$ i jego macierzy generującej $\mathbb{G}$ powstałej z bazy kodu $\mathcal{B}$ algorytm `MinimizeHammingDistance` użyty do dekodowania słowa kodowego $w \in \mathcal{C}$ zwróci taki wektor $v \in \mathbb{K}^k$, który w wyniku zakodowania go z użyciem macierzy $\mathbb{G}$ da wektor w.
Ustalmy $w \in \mathcal{C}$. Zauważmy, że $|C| < |\mathbb{K}^k| = |\mathbb{K}|^k \in \mathbb{N}$, ponieważ $|\mathbb{K}|$ jest skończona. Znalezienie m odbywa się w skończonej liczbie kroków, bowiem wystarczy przeiterować po każdym $v \in \mathcal{C}$ i liczyć $d(v,w)$. Szukane $m = 0$, gdyż $w \in \mathcal{C}$ i znajdziemy $v \in \mathcal{C}$, takie że $d(v,w) = 0$ (innymi słowy, w zbiorze $\mathcal{C}$ znajdziemy $v = w$).
Drugi etap polega na wzięciu ze zbioru L wektora w (zbiór L jest jednoelementowy, bowiem $d(v,w) = 0$ wtedy i tylko wtedy, gdy $v = w$).
Pozostało zapisać wektor w w bazie $\mathcal{B}$, co, jak w poprzednich etapach, odbywa się w skończonej liczbie kroków.

Niech $\mathcal{B} = (e_1, e_2, ..., e_k)$ będzie bazą $\mathcal{C}$ (e_i zapisujemy pionowo), $v = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{pmatrix}$ będzie zapisem współrzędnych wektora w w bazie $\mathcal{B}$, $w = a_1 \cdot e_1 + a_2 \cdot e_2 + ... + a_k \cdot e_k$ oraz niech $\mathbb{G}$ będzie macierzą generującą kodu $\mathcal{C}$, z definicji macierzy generującej kodu liniowego $\mathbb{G} = \begin{pmatrix} e_1^T \\ e_2^T \\ \vdots \\ e_k^T \end{pmatrix}$.
Wówczas $(v^T \cdot \mathbb{G})^T = \mathbb{G}^T \cdot v = a_1 \cdot e_1 + a_2 \cdot e_2 + ... + a_k \cdot e_k = w$.

□

Zadanko 4.

Dowód. Ustalmy $u, v, x \in V$. Wykażemy, że $d(u, v) = d(u + x, v + x)$. Niech $d(u, v) = k$, gdzie $k \in [n] \cup \{0\}$, czyli wektory u i v różnią sie na k indeksach. Dodawszy do u i v wektor x, wektory $u + x$ i $v + x$ wciąż różnią się na k indeksach. Gwoli wyjaśnienia, dzieje się tak dlatego, że jeśli $u_i \neq v_i$ dla $i \in [n]$, to $u_i + x_i \neq v_i + x_i$; paralelnie jeśli $u_i = v_i$ dla $i \in [n]$, to $u_i + x_i = v_i + x_i$.

□

Napisane przez nas programy, na podstawie których rozwiązaliśmy zadania 5-8, dostępne są w repozytorium.

Zadanko 5.

Niech $u := (1,2,0,1)^T$, $v := (0,0,0,1)^T$. Łatwo zauważyć, że odległość Hamminga między wektorami u i v wynosi 2, sprawdzamy to w *Mathematice* i rzeczywiście jest to 2. Niech $w_1 := (1,2,1,2,0)^T$, $w_2 := (1,1,1,1,1)^T$, $w_3 := (0,0,2,1,1)^T$, $w_4 := (2,2,2,1,0)^T$. Korzystamy z dobrze znanego programu *Mathematica* i otrzymujemy:
$d(w_1, w_2) = 3$,
$d(w_1, w_3) = 5$,
$d(w_1, w_4) = 3$,
$d(w_2, w_3) = 3$,
$d(w_2, w_4) = 4$,
$d(w_3, w_4) = 3$,
wobec czego najbliżej w sensie Hamminga są w_1 i w_2, w_1 i w_4, w_2 i w_3 oraz w_3 i w_4.

Zadanko 6.

Wygenerujemy wszystkie słowa kodowe dla (5, 3)-kodu liniowego $\mathcal{C}$ nad ciałem $\mathbb{Z}_7$, gdzie bazą kodu liniowego $\mathcal{C}$ jest

$$\mathcal{B} = \left(\begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 4 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 5 \\ 6 \end{pmatrix} \right).$$

Naszym celem jest znalezienie wszystkich wektorów $v \in \mathbb{Z}_7^5$, które są kombinacjami liniowymi wektorów z bazy $\mathcal{B}$. Napisawszy prosty program w *Pythonie*, otrzymujemy wszystkie słowa kodowe (jest ich 343):

$(0,0,0,0,0)^T$, $(0,0,1,5,6)^T$, $(0,0,2,3,5)^T$, $(0,0,3,1,4)^T$, $(0,0,4,6,3)^T$, $(0,0,5,4,2)^T$, $(0,0,6,2,1)^T$, $(0,1,0,1,0)^T$,
$(0,1,1,6,6)^T$, $(0,1,2,4,5)^T$, $(0,1,3,2,4)^T$, $(0,1,4,0,3)^T$, $(0,1,5,5,2)^T$, $(0,1,6,3,1)^T$, $(0,2,0,2,0)^T$, $(0,2,1,0,6)^T$,
$(0,2,2,5,5)^T$, $(0,2,3,3,4)^T$, $(0,2,4,1,3)^T$, $(0,2,5,6,2)^T$, $(0,2,6,4,1)^T$, $(0,3,0,3,0)^T$, $(0,3,1,1,6)^T$, $(0,3,2,6,5)^T$,
$(0,3,3,4,4)^T$, $(0,3,4,2,3)^T$, $(0,3,5,0,2)^T$, $(0,3,6,5,1)^T$, $(0,4,0,4,0)^T$, $(0,4,1,2,6)^T$, $(0,4,2,0,5)^T$, $(0,4,3,5,4)^T$,
$(0,4,4,3,3)^T$, $(0,4,5,1,2)^T$, $(0,4,6,6,1)^T$, $(0,5,0,5,0)^T$, $(0,5,1,3,6)^T$, $(0,5,2,1,5)^T$, $(0,5,3,6,4)^T$, $(0,5,4,4,3)^T$,
$(0,5,5,2,2)^T$, $(0,5,6,0,1)^T$, $(0,6,0,6,0)^T$, $(0,6,1,4,6)^T$, $(0,6,2,2,5)^T$, $(0,6,3,0,4)^T$, $(0,6,4,5,3)^T$, $(0,6,5,3,2)^T$,
$(0,6,6,1,1)^T$, $(1,0,0,2,4)^T$, $(1,0,1,0,3)^T$, $(1,0,2,5,2)^T$, $(1,0,3,3,1)^T$, $(1,0,4,1,0)^T$, $(1,0,5,6,6)^T$, $(1,0,6,4,5)^T$,
$(1,1,0,3,4)^T$, $(1,1,1,1,3)^T$, $(1,1,2,6,2)^T$, $(1,1,3,4,1)^T$, $(1,1,4,2,0)^T$, $(1,1,5,0,6)^T$, $(1,1,6,5,5)^T$, $(1,2,0,4,4)^T$,
$(1,2,1,2,3)^T$, $(1,2,2,0,2)^T$, $(1,2,3,5,1)^T$, $(1,2,4,3,0)^T$, $(1,2,5,1,6)^T$, $(1,2,6,6,5)^T$, $(1,3,0,5,4)^T$, $(1,3,1,3,3)^T$,
$(1,3,2,1,2)^T$, $(1,3,3,6,1)^T$, $(1,3,4,4,0)^T$, $(1,3,5,2,6)^T$, $(1,3,6,0,5)^T$, $(1,4,0,6,4)^T$, $(1,4,1,4,3)^T$, $(1,4,2,2,2)^T$,
$(1,4,3,0,1)^T$, $(1,4,4,5,0)^T$, $(1,4,5,3,6)^T$, $(1,4,6,1,5)^T$, $(1,5,0,0,4)^T$, $(1,5,1,5,3)^T$, $(1,5,2,3,2)^T$, $(1,5,3,1,1)^T$,
$(1,5,4,6,0)^T$, $(1,5,5,4,6)^T$, $(1,5,6,2,5)^T$, $(1,6,0,1,4)^T$, $(1,6,1,6,3)^T$, $(1,6,2,4,2)^T$, $(1,6,3,2,1)^T$, $(1,6,4,0,0)^T$,
$(1,6,5,5,6)^T$, $(1,6,6,3,5)^T$, $(2,0,0,4,1)^T$, $(2,0,1,2,0)^T$, $(2,0,2,0,6)^T$, $(2,0,3,5,5)^T$, $(2,0,4,3,4)^T$, $(2,0,5,1,3)^T$,
$(2,0,6,6,2)^T$, $(2,1,0,5,1)^T$, $(2,1,1,3,0)^T$, $(2,1,2,1,6)^T$, $(2,1,3,6,5)^T$, $(2,1,4,4,4)^T$, $(2,1,5,2,3)^T$, $(2,1,6,0,2)^T$,
$(2,2,0,6,1)^T$, $(2,2,1,4,0)^T$, $(2,2,2,2,6)^T$, $(2,2,3,0,5)^T$, $(2,2,4,5,4)^T$, $(2,2,5,3,3)^T$, $(2,2,6,1,2)^T$, $(2,3,0,0,1)^T$,
$(2,3,1,5,0)^T$, $(2,3,2,3,6)^T$, $(2,3,3,1,5)^T$, $(2,3,4,6,4)^T$, $(2,3,5,4,3)^T$, $(2,3,6,2,2)^T$, $(2,4,0,1,1)^T$, $(2,4,1,6,0)^T$,
$(2,4,2,4,6)^T$, $(2,4,3,2,5)^T$, $(2,4,4,0,4)^T$, $(2,4,5,5,3)^T$, $(2,4,6,3,2)^T$, $(2,5,0,2,1)^T$, $(2,5,1,0,0)^T$, $(2,5,2,5,6)^T$,
$(2,5,3,3,5)^T$, $(2,5,4,1,4)^T$, $(2,5,5,6,3)^T$, $(2,5,6,4,2)^T$, $(2,6,0,3,1)^T$, $(2,6,1,1,0)^T$, $(2,6,2,6,6)^T$, $(2,6,3,4,5)^T$,
$(2,6,4,2,4)^T$, $(2,6,5,0,3)^T$, $(2,6,6,5,2)^T$, $(3,0,0,6,5)^T$, $(3,0,1,4,4)^T$, $(3,0,2,2,3)^T$, $(3,0,3,0,2)^T$, $(3,0,4,5,1)^T$,
$(3,0,5,3,0)^T$, $(3,0,6,1,6)^T$, $(3,1,0,0,5)^T$, $(3,1,1,5,4)^T$, $(3,1,2,3,3)^T$, $(3,1,3,1,2)^T$, $(3,1,4,6,1)^T$, $(3,1,5,4,0)^T$,
$(3,1,6,2,6)^T$, $(3,2,0,1,5)^T$, $(3,2,1,6,4)^T$, $(3,2,2,4,3)^T$, $(3,2,3,2,2)^T$, $(3,2,4,0,1)^T$, $(3,2,5,5,0)^T$, $(3,2,6,3,6)^T$,
$(3,3,0,2,5)^T$, $(3,3,1,0,4)^T$, $(3,3,2,5,3)^T$, $(3,3,3,3,2)^T$, $(3,3,4,1,1)^T$, $(3,3,5,6,0)^T$, $(3,3,6,4,6)^T$, $(3,4,0,3,5)^T$,
$(3,4,1,1,4)^T$, $(3,4,2,6,3)^T$, $(3,4,3,4,2)^T$, $(3,4,4,2,1)^T$, $(3,4,5,0,0)^T$, $(3,4,6,5,6)^T$, $(3,5,0,4,5)^T$, $(3,5,1,2,4)^T$,
$(3,5,2,0,3)^T$, $(3,5,3,5,2)^T$, $(3,5,4,3,1)^T$, $(3,5,5,1,0)^T$, $(3,5,6,6,6)^T$, $(3,6,0,5,5)^T$, $(3,6,1,3,4)^T$, $(3,6,2,1,3)^T$,
$(3,6,3,6,2)^T$, $(3,6,4,4,1)^T$, $(3,6,5,2,0)^T$, $(3,6,6,0,6)^T$, $(4,0,0,1,2)^T$, $(4,0,1,6,1)^T$, $(4,0,2,4,0)^T$, $(4,0,3,2,6)^T$,
$(4,0,4,0,5)^T$, $(4,0,5,5,4)^T$, $(4,0,6,3,3)^T$, $(4,1,0,2,2)^T$, $(4,1,1,0,1)^T$, $(4,1,2,5,0)^T$, $(4,1,3,3,6)^T$, $(4,1,4,1,5)^T$,
$(4,1,5,6,4)^T$, $(4,1,6,4,3)^T$, $(4,2,0,3,2)^T$, $(4,2,1,1,1)^T$, $(4,2,2,6,0)^T$, $(4,2,3,4,6)^T$, $(4,2,4,2,5)^T$, $(4,2,5,0,4)^T$,
$(4,2,6,5,3)^T$, $(4,3,0,4,2)^T$, $(4,3,1,2,1)^T$, $(4,3,2,0,0)^T$, $(4,3,3,5,6)^T$, $(4,3,4,3,5)^T$, $(4,3,5,1,4)^T$, $(4,3,6,6,3)^T$,
$(4,4,0,5,2)^T$, $(4,4,1,3,1)^T$, $(4,4,2,1,0)^T$, $(4,4,3,6,6)^T$, $(4,4,4,4,5)^T$, $(4,4,5,2,4)^T$, $(4,4,6,0,3)^T$, $(4,5,0,6,2)^T$,
$(4,5,1,4,1)^T$, $(4,5,2,2,0)^T$, $(4,5,3,0,6)^T$, $(4,5,4,5,5)^T$, $(4,5,5,3,4)^T$, $(4,5,6,1,3)^T$, $(4,6,0,0,2)^T$, $(4,6,1,5,1)^T$,
$(4,6,2,3,0)^T$, $(4,6,3,1,6)^T$, $(4,6,4,6,5)^T$, $(4,6,5,4,4)^T$, $(4,6,6,2,3)^T$, $(5,0,0,3,6)^T$, $(5,0,1,1,5)^T$, $(5,0,2,6,4)^T$,
$(5,0,3,4,3)^T$, $(5,0,4,2,2)^T$, $(5,0,5,0,1)^T$, $(5,0,6,5,0)^T$, $(5,1,0,4,6)^T$, $(5,1,1,2,5)^T$, $(5,1,2,0,4)^T$, $(5,1,3,5,3)^T$,
$(5,1,4,3,2)^T$, $(5,1,5,1,1)^T$, $(5,1,6,6,0)^T$, $(5,2,0,5,6)^T$, $(5,2,1,3,5)^T$, $(5,2,2,1,4)^T$, $(5,2,3,6,3)^T$, $(5,2,4,4,2)^T$,
$(5,2,5,2,1)^T$, $(5,2,6,0,0)^T$, $(5,3,0,6,6)^T$, $(5,3,1,4,5)^T$, $(5,3,2,2,4)^T$, $(5,3,3,0,3)^T$, $(5,3,4,5,2)^T$, $(5,3,5,3,1)^T$,
$(5,3,6,1,0)^T$, $(5,4,0,0,6)^T$, $(5,4,1,5,5)^T$, $(5,4,2,3,4)^T$, $(5,4,3,1,3)^T$, $(5,4,4,6,2)^T$, $(5,4,5,4,1)^T$, $(5,4,6,2,0)^T$,
$(5,5,0,1,6)^T$, $(5,5,1,6,5)^T$, $(5,5,2,4,4)^T$, $(5,5,3,2,3)^T$, $(5,5,4,0,2)^T$, $(5,5,5,5,1)^T$, $(5,5,6,3,0)^T$, $(5,6,0,2,6)^T$,
$(5,6,1,0,5)^T$, $(5,6,2,5,4)^T$, $(5,6,3,3,3)^T$, $(5,6,4,1,2)^T$, $(5,6,5,6,1)^T$, $(5,6,6,4,0)^T$, $(6,0,0,5,3)^T$, $(6,0,1,3,2)^T$,
$(6,0,2,1,1)^T$, $(6,0,3,6,0)^T$, $(6,0,4,4,6)^T$, $(6,0,5,2,5)^T$, $(6,0,6,0,4)^T$, $(6,1,0,6,3)^T$, $(6,1,1,4,2)^T$, $(6,1,2,2,1)^T$,
$(6,1,3,0,0)^T$, $(6,1,4,5,6)^T$, $(6,1,5,3,5)^T$, $(6,1,6,1,4)^T$, $(6,2,0,0,3)^T$, $(6,2,1,5,2)^T$, $(6,2,2,3,1)^T$, $(6,2,3,1,0)^T$,
$(6,2,4,6,6)^T$, $(6,2,5,4,5)^T$, $(6,2,6,2,4)^T$, $(6,3,0,1,3)^T$, $(6,3,1,6,2)^T$, $(6,3,2,4,1)^T$, $(6,3,3,2,0)^T$, $(6,3,4,0,6)^T$,

$(6,3,5,5,5)^T$, $(6,3,6,3,4)^T$, $(6,4,0,2,3)^T$, $(6,4,1,0,2)^T$, $(6,4,2,5,1)^T$, $(6,4,3,3,0)^T$, $(6,4,4,1,6)^T$, $(6,4,5,6,5)^T$, $(6,4,6,4,4)^T$, $(6,5,0,3,3)^T$, $(6,5,1,1,2)^T$, $(6,5,2,6,1)^T$, $(6,5,3,4,0)^T$, $(6,5,4,2,6)^T$, $(6,5,5,0,5)^T$, $(6,5,6,5,4)^T$, $(6,6,0,4,3)^T$, $(6,6,1,2,2)^T$, $(6,6,2,0,1)^T$, $(6,6,3,5,0)^T$, $(6,6,4,3,6)^T$, $(6,6,5,1,5)^T$, $(6,6,6,6,4)^T$.

Zadanko 7.

Nietrudno zauważyć, że macierz generująca $\mathbb{G} = \begin{pmatrix} 1 & 0 & 0 & 2 & 4 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 5 & 6 \end{pmatrix}$. Niech $v = (2, 1, 3, 6, 0)^T$, oczywiście $v \in \mathbb{Z}_7^5$. Zaimplementowaliśmy algorytm `MinimizeHammingDistance` w *Pythonie*, wykorzystując bibliotekę `scipy.spatial.distance` i otrzymaliśmy, że najbliższy v w sensie Hamminga jest wektor $w = (2, 1, 3, 6, 5)^T$ oraz, że wektor r współrzędnych wektora w w bazie $\mathcal{B}$ to $r = (2, 1, 3)$. Sprawdzamy "manualnie" zgodność z prawdą i otrzymujemy pozytywny wynik ☺.

Zadanko 8.

W celu symulacji zaimplementowaliśmy odpowiedni kod w *Pythonie*, obliczenia wykonujemy dla stałego ziarna generatora liczb losowych $random.seed = 2137$.

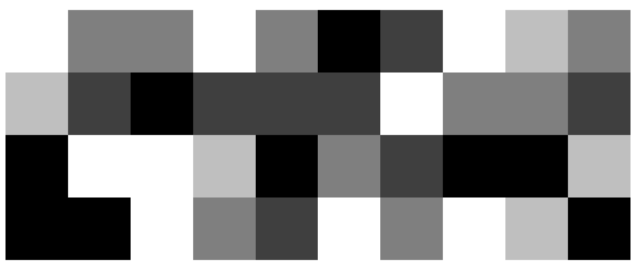
(a) Losowo wygenerowana macierz o 10 kolumnach i 4 wierszach, o elementach z ciała $\mathbb{Z}_5$ to

$$A = \begin{pmatrix} 4 & 2 & 2 & 4 & 2 & 0 & 1 & 4 & 3 & 2 \\ 3 & 1 & 0 & 1 & 1 & 1 & 4 & 2 & 2 & 1 \\ 0 & 4 & 4 & 3 & 0 & 2 & 1 & 0 & 0 & 3 \\ 0 & 0 & 4 & 2 & 1 & 4 & 2 & 4 & 3 & 0 \end{pmatrix}.$$

(b) Normujemy macierz A do przedziału $[0, 1]$, dzieląc każdy jej element przez 4. Otrzymujemy:

$$X = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & 0 & \frac{1}{4} & 1 & \frac{3}{4} & \frac{1}{2} \\ \frac{3}{4} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} \\ 0 & 1 & 1 & \frac{3}{4} & 0 & \frac{1}{2} & \frac{1}{4} & 0 & 0 & \frac{3}{4} \\ 0 & 0 & 1 & \frac{1}{2} & \frac{1}{4} & 1 & \frac{1}{2} & 1 & \frac{3}{4} & 0 \end{pmatrix}.$$

W celu stworzenia obrazu macierzy X, korzytsamy z funkcji `Image` w Mathematice, otrzymujemy:



Rysunek 1: Obraz macierzy X.

(c) *Dowód.* Wykażemy, że istnieje (11,4)-kod liniowy $\mathcal{C}$ nad ciałem $\mathbb{Z}_6$,

taki że $\mathbb{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 4 & 4 & 2 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 3 & 0 & 2 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 4 & 3 & 0 \end{pmatrix}$ jest macierzą generującą kodu $\mathcal{C}$.

Niech v_i będzie i-tym wektorem macierzy $\mathbb{G}$, $i \in [4]$ oraz $\mathcal{C} = \mathcal{L}(v_1, v_2, v_3, v_4)$ (nad ciałem $\mathbb{Z}_5$). Zbiór (v_1, v_2, v_3, v_4) jest liniowo niezależny, więc jest bazą $\mathcal{C}$. Ponadto $\mathcal{C} < \mathbb{Z}_5^{11}$, gdzie $\mathbb{Z}_5^{11}$ jest przestrzenią liniową nad ciałem $\mathbb{Z}_5$. Ciało $\mathbb{Z}_5$ ma skończoną liczność. Wobec tego $\mathcal{C}$ jest (11,4)-kodem liniowym nad ciałem $\mathbb{Z}_5$, $\mathcal{B} = (v_1, v_2, v_3, v_4)$ - bazą kodu $\mathcal{C}$, toteż $\mathbb{G}$ macierzą generującą kodu $\mathcal{C}$.

□

(d) Mamy daną macierz generującą $\mathbb{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 4 & 4 & 2 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 3 & 0 & 2 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 4 & 3 & 0 \end{pmatrix}.$

Niech $v :=$ pierwsza kolumna macierzy A, czyli $v = (4, 3, 0, 0)^T$ oraz $w :=$ kodowanie wektora v przy użyciu macierzy generującej $\mathbb{G}$, $w = (v^T \cdot \mathbb{G})^T$. Oczywiście, każdy wektor musi mieć elementy z ciała Z_5, toteż działania wykonujemy modulo 5. Otrzymujemy $w = (4, 3, 0, 0, 0, 0, 1, 4, 1, 2, 4)^T$.

(e) Kodujemy każdą kolumnę v_i ($i \in [10]$) macierzy A. Zapisujemy zakodowane wektory w macierzy A', wówczas $A' = (w_1, w_2, ..., w_{10})$, gdzie $w_i = (v_i^T \cdot \mathbb{G})^T$, $i \in [10]$. Oczywiście, każdy wektor musi mieć elementy z ciała Z_5, toteż działania wykonujemy modulo 5. Mamy:

$$A' = \begin{pmatrix} 4 & 2 & 2 & 4 & 2 & 0 & 1 & 4 & 3 & 2 \\ 3 & 1 & 0 & 1 & 1 & 1 & 4 & 2 & 2 & 1 \\ 0 & 4 & 4 & 3 & 0 & 2 & 1 & 0 & 0 & 3 \\ 0 & 0 & 4 & 2 & 1 & 4 & 2 & 4 & 3 & 0 \\ 0 & 0 & 4 & 2 & 1 & 4 & 2 & 4 & 3 & 0 \\ 0 & 4 & 1 & 0 & 1 & 2 & 3 & 2 & 3 & 2 \\ 1 & 3 & 3 & 1 & 3 & 0 & 4 & 1 & 2 & 3 \\ 4 & 0 & 3 & 3 & 1 & 4 & 1 & 2 & 0 & 4 \\ 1 & 1 & 0 & 3 & 1 & 0 & 2 & 0 & 1 & 0 \\ 2 & 2 & 3 & 4 & 1 & 0 & 2 & 3 & 4 & 1 \\ 4 & 1 & 1 & 2 & 2 & 2 & 2 & 4 & 3 & 0 \end{pmatrix}.$$

Dla każdego zakodowanego wektora (tj. kolumny macierzy) zasymulujemy wysłanie go do pewnego użytkownika poprzez kanał, który dla przesyłanego wektora v dla każdej pozycji dodaje modulo 5 losową liczbę ze zbioru $\{0, 3\}$. Mamy więc $M :=$ "przesłana" macierz A':

$$M = \begin{pmatrix} 4 & 2 & 2 & 4 & 2 & 0 & 1 & 2 & 3 & 2 \\ 3 & 1 & 3 & 1 & 1 & 1 & 4 & 2 & 2 & 1 \\ 0 & 4 & 2 & 3 & 0 & 2 & 1 & 0 & 0 & 3 \\ 0 & 0 & 4 & 2 & 4 & 4 & 2 & 4 & 3 & 0 \\ 0 & 0 & 4 & 2 & 1 & 4 & 2 & 4 & 3 & 0 \\ 3 & 4 & 1 & 0 & 1 & 2 & 1 & 2 & 3 & 2 \\ 1 & 3 & 3 & 1 & 3 & 0 & 4 & 1 & 2 & 3 \\ 4 & 0 & 3 & 3 & 1 & 4 & 1 & 2 & 0 & 4 \\ 1 & 1 & 0 & 3 & 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 2 & 3 & 4 & 4 & 0 & 2 & 3 & 4 & 1 \\ 4 & 1 & 1 & 2 & 0 & 2 & 2 & 4 & 3 & 0 \end{pmatrix}.$$

(f, g) Dla każdego zakodowanego wektora (każdej kolumny macierzy B) odkodowujemy przy użyciu algorytmu `MinimizeHammingDistance`. Wyniki zapisujemy w macierzy $\mathcal{M}$ tak, że i-ta kolumna odpowiada i-temu wektorowi współrzędnych zakodowanego i przesłanego wektora r współczynników wektora w w bazie $\mathcal{B}$ (oznaczenia jak w algorytmie `MinimizeHammingDistance`),
gdzie $\mathcal{B} = ((1, 0, 0, 0, 0, 4, 4, 2, 0, 1, 1)^T, (0, 1, 0, 0, 0, 3, 0, 2, 2, 1, 0)^T, (0, 0, 1, 0, 0, 2, 0, 1, 1, 1, 1)^T,$
$(0, 0, 0, 1, 1, 0, 0, 0, 4, 3, 0)^T)$. Mamy:

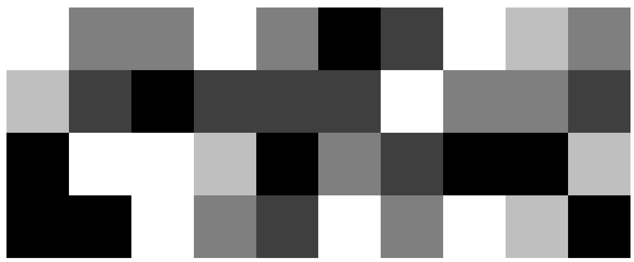
$$\mathcal{M} = \begin{pmatrix} 4 & 2 & 2 & 4 & 2 & 0 & 1 & 4 & 3 & 2 \\ 3 & 1 & 0 & 1 & 1 & 1 & 4 & 2 & 2 & 1 \\ 0 & 4 & 4 & 3 & 0 & 2 & 1 & 0 & 0 & 3 \\ 0 & 0 & 4 & 2 & 1 & 4 & 2 & 4 & 3 & 0 \end{pmatrix}.$$

(h) Zauważamy, że **wszystkie** kolumny zostały dobrze odkodowane ☺.

(i) Normujemy macierz $\mathcal{M}$ do przedziału $[0,1]$, dzieląc każdy jej element przez 4. Otrzymujemy:

$$Y = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & 0 & \frac{1}{4} & 1 & \frac{3}{4} & \frac{1}{2} \\ \frac{3}{4} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} \\ 0 & 1 & 1 & \frac{3}{4} & 0 & \frac{1}{2} & \frac{1}{4} & 0 & 0 & \frac{3}{4} \\ 0 & 0 & 1 & \frac{1}{2} & \frac{1}{4} & 1 & \frac{1}{2} & 1 & \frac{3}{4} & 0 \end{pmatrix}.$$

Tworzymy obraz macierzy Y, korzytsamy z funkcji `Image` w Mathematice, otrzymujemy:



Rysunek 2: Obraz macierzy Y.