Упражнения из учебника

Кокорин Илья, М3439

25 сентября 2019 г.

1 Задача 1.8

1.1 Условия эквивалетности декодирования по МП и по Минимуму расстояние Хемминга на нестёртых позициях

Пусть x - отправленное сообщение по двоичному каналу со стираниями, y - полученное.

Пусть $|x| = |y| = n$, при этом k позиций были стёрты, а на m позициях случились ошибки. Тогда расстояние Хеминга на нестёртых позициях между x и y (будем обозначать его $d_{ne}(x, y)$) равно m.

Если вероятность стирания равна ϵ, а вероятность замены равна p_0, тогда вероятность такого события равна $p(y|x) = p_0^m \cdot \epsilon^k \cdot (1 - p_0 - \epsilon)^{n-k-m}$, так как вероятность сохранения символа равна $1 - p_0 - \epsilon$, а сохранились ровно $n - k - m$. (напоминим, что рассматривается стационарный канал без памяти).

Декодированием по МАВ называется выбор такого кодового слова c_m, что $p(y|c_m) \to max$, а декодированием по минимуму расстояния Хеминга на нестёртых позициях - выбор такого кодового слова c_m, что $d_{ne}(y, c_m) \to min$. Значит, для эквивалетности этих методов при уменьшении расстояния Хеминга на нестёртых позициях (то есть $d_{ne}(x, y) = m$), должно уменьшаться $p(y|x)$

$p(y|x) = p_0^m \cdot \epsilon^k \cdot \frac{(1-p_0-\epsilon)^n}{(1-p_0-\epsilon)^{k+m}} = \frac{p_0^m}{(1-p_0-\epsilon)^m} \cdot \epsilon^k \cdot \frac{(1-p_0-\epsilon)^n}{(1-p_0-\epsilon)^k} = (\frac{p_0}{1-p_0-\epsilon})^m \cdot \epsilon^k \cdot (1 - p_0 - \epsilon)^{n-k}$

Заметим, что $\epsilon^k \cdot (1 - p_0 - \epsilon)^{n-k}$ является константой (так как мы зафиксировали все вероятности и количество стёртых позиций), поэтому для выполнения условия ($p(y|x)$ уменьшается с ростом m), нам необходимо выполнение $\lim\limits_{m\to\infty}(\frac{p_0}{1-p_0-\epsilon})^m = 0$, что эквивалетно $0 < \frac{p_0}{1-p_0-\epsilon} < 1$

Ограничение $\frac{p_0}{1-p_0-\epsilon} > 0$ тривиально выполняется, так как $p_0 > 0$ (то есть существует ненулевая вероятность ошибки) и $1 - p_0 - \epsilon > 0$ (то есть существует ненулевая вероятность корретной доставки). Очевидно, что это правда для любых каналов.

Рассмотрим второе ограничение: $\frac{p_0}{1-p_0-\epsilon} < 1 \Rightarrow p_0 < 1 - p_0 - \epsilon \Rightarrow p_0 < \frac{1-\epsilon}{2}$

1.2 Сколько стираний исправляет код с минимальным расстоянием d

Рассмотрим ситуацию, когда произошло 0 ошибок и $d - 1$ стираний. Все кодовые слова различаются как минимум в d битах. Пусть x - отправленное слово, y - полученное, в котором произошло $d - 1$ стираний. Заметим, что на остальных $n - d + 1$ позициях стоят те же символы, что и в x. Значит, как минимум в одном из нестёртых символов y отличается от $c_m \neq x$, значит, только x может быть отправленным кодовым словом, и эти стирания можно исправить. То есть такой код исправляет $d - 1$ стираний.

Рассмотрим другую ситуацию, при которой было отправлено кодовое слово x, и существует кодовое слово z, которое отличается от x в d позициях. Пусть при посылке по каналу кодового слова x было принято кодовое слово y, которое содержит d стёртых позиций, причём ровно тех, в которых x отличается от z. Тогда декодер не сможет решить, какое слово (x или z) было отправлено, и исправить стирания не получится. То есть такой код не исправляет d стираний.

Тогда максимально код может исправить $d - 1$ стираний.

1.3 Сколько ошибок исправляет код с минимальным расстоянием d при наличии s стираний

Для начала заметим, что если $s \geq d$, то код не сможет исправить ни одной ошибки (так как все позиции, в котороых два кодовые слова различаются, могут быть стёрты, и тогда будет не ясно, как восстанавливать ошибки).

Рассмотрим ситуацию, при которой $s < d$. Минимальное расстояние, если считать его по нестёртым позициям, будет равно $d - s$. Теперь, учитывая тот факт, что код с минимальным расстоянием t исправляет $\lfloor \frac{t-1}{2} \rfloor$ ошибок, а минимальное расстояние по нестёртым позициям $d - s$, то такой код сможет исправлять до $\lfloor \frac{d-s-1}{2} \rfloor$ ошибок.

2 Задача 2.1

2.1 k = 1

При $k = 1$ существует $2^1 = 2$ кодовых слова, одно из которых всегда нулевое (так как нулевой вектор всегда принадлежит линейному подпространству).

Единственный оставшийся ненулевой вектор должен иметь максимальный вес, и, очевидно, этот вектор $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$. Тогда

$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

2.2 k = 2

При $k = 2$ существует $2^2 = 4$ кодовых слов, из которых 3 ненулевые. 2 из них будут строками матрицы G, и последний оставшийся - их линейной комбинацией с коэффициентами $\begin{pmatrix} 1 & 1 \end{pmatrix}$

Покажем существование кода с $d = 4$

$G = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$

Заметим, что все три ненулевых вектора имеют вес 4.

Докажем, что большее минимальное расстояние получить невозможно. Для того, чтобы d стало равным пяти, необходимо, чтобы все ненулевые вектора имели вес равный пяти. При этом если мы попробуем получить два шестиэлементных вектора, в каждом из которых содержится по пять единиц, то их линейная комбинация будет содержать не более 2 позиций, в которых единицы не перекрываются, а следовательно, иметь вес не более 2.

3 k = 3

$k = 3, n = 6 \Rightarrow r = 3$

Рассмотрим проверочные матрицы H.

Покажем существование кода с $d = 3$

$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$

Заметим, что все столбцы проверочной матрицы попарно различны (что в F_2 эквивалетно линейной независимости двух векторов), но при этом $\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$

То есть 1, 2 и 3 столбцы линейно зависимы. Тогда $d = 3$.

Докажем, что нельзя получить 4.

Для получения $d = 4$ необходимо, чтобы любые 3 столбца матрицы H были линейно независимы.

Для этого нам, как минимум, нужно, чтобы все вектора были попарно неравны. Всего двоичных векторов длины 3 имеется $2^3 = 8$, из них нужно выбрать 6 различных так, чтобы любые 3 из этих 6 были линейно независимы. Напишем программу, показывающую невозможность такого выбора. (Программа приложена к письму)

4 k = 4

$k = 4, n = 6 \Rightarrow r = 2$

Для начала, продемонстрируем возможность получения кода с $d = 2$.

$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$

Очевидно, для этого кода $d = 2$.

Докажем, что мы не можем получить минимального расстояния больше двух.

Для того, чтобы минимальное расстояние кода было больше двух, все столбцы матрицы H должны быть попарно линейно независимы (а значит, попарно различны). Существует $2^2 = 4$ битовых вектор-столбцов размерности 2, из которых нужно выбрать 6 попарно различных. Очевидно, этого сделать нельзя.

5 k = 5

$k = 5, n = 6 \Rightarrow r = 1$

Для начала, продемонстрируем возможность получения кода с $d = 2$.

$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

Очевидно, для этого кода $d = 2$.

Минимальное расстояние не может быть больше двух согласно границе Синглтона: теорема о границе Синглтона заявляет, что $d \leq n - k + 1 = 6 - 5 + 1 = 2$

6 k = 6

Минимальное расстояние не может быть больше одного согласно границе Синглтона: теорема о границе Синглтона заявляет, что $d \leq n - k + 1 = 6 - 6 + 1 = 1$

Выберем тривиальный код для демонстрации примера кода с минимальным расстоянием 1.

$G = I_6$

Очевидно, что в этом коде минимальный ненулевой вектор имеет вес 1 (этот вектор $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$)