Московский государственный университет им. М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Задание № 2.
Низкоплотностные коды

Автор: Арбузова Дарья
Группа: 417

Москва, 2014

Содержание

1 Цели задания

1. Реализовать алгоритм декодирования низкоплотностного кода на основе loopy BP;
2. Провести эксперименты с различными расписаниями пересчёта сообщений и коэффициентами дэмпфирования;
3. Реализовать алгоритм оценки вероятности битовой и блоковой ошибки кода с помощью метода статистических испытаний;
4. Провести эксперименты по оцениванию битовой и блоковой ошибки низкоплотностного кода для различных значений параметров;
5. Провести эксперименты по сравнению низкоплотностного кода с кодами БЧХ.

2 Теория помехоустойчивого кодирования

Рассмотрим решение задачи безошибочной передачи потока битовой информации по каналу с шумом с помощью кодов, исправляющих ошибки.

Пусть передаваемое сообщение разделяется на блоки длины k, и каждый из них кодируется независимо. Для исправления возможных помех при передаче информации необходимо добавить избыточность: пусть каждый блок кодируется словом длины $n > k$. Таким образом, каждому возможному блоку длины k сопоставляется одно из 2^k кодовых слов длины n. Множество этих слов называется $(n, k)-$блоковым кодом, а $r = \frac{k}{n}$ — скоростью кода.

Стадии жизни сообщения:

$$u \in \{0,1\}^k \xrightarrow[\text{вание}]{\text{кодиро-}} v \in \{0,1\}^n \xrightarrow[\text{помехами}]{\text{канал с}} w \in \{0,1\}^n \xrightarrow[\text{ошибок}]{\text{восстановление}} \hat{v} \in \{0,1\}^n \xrightarrow[\text{рование}]{\text{декоди-}} \hat{u} \in \{0,1\}^k$$

Рассмотрим следующую модель канала: пусть ошибка в каждом бите совершается с вероятностью $q \in \left(0; \frac{1}{2}\right)$. Пропускной способностью канала называется максимальная скорость, с которой может быть осуществлена надёжная передачи информации. В рассматриваемом случае определяется величиной $c = 1 + q\log_2 q + (1-q)\log_2(1-q)$.

Зададим линейный $(n, k)-$блоковый код его проверочной матрицей $H \in \{0,1\}^{(n-k)\times n}$: $Hv = 0 \,(\mathrm{mod}\ 2) \iff v$ — кодовое слово. По ней можно найти порождающую матрицу кода для кодирования блоков $G \in \{0,1\}^{n\times k}$: $Gu = v$. Подробное описание алгоритма можно найти в [1].

Особенностью низкоплотностных кодов является сильная разреженность матрицы H.

Пусть получено сообщение $w \in \{0,1\}^n$, и требуется восстановить вектор ошибок $e \in \{0,1\}^n : w = v + e$. Назовём синдромом w вектор $s \in \{0,1\}^{n-k} : s = Hw = H(v+e) = = Hv + He = He$.

Основная задача декодирования состоит в решении уравнения $s = He$, и делать это можно многими разными способами. В данном задании был исследован метод, использующий аппарат графических моделей.

При использовании побитовой функции потерь $\lambda(e, \tilde{e}) = \sum\limits_{i=1}^{n}[e_i \neq \tilde{e}_i]$ оптимальная процедура декодирования связана с максимизацией маргиналов отдельных переменных: $\hat{e}_i = \arg\max\limits_{e_i} p(e_i|s)$. Для поиска маргинальных распределений $p(e_i|s)$ применяется алгоритм sum-product loopy belief propagation на фактор-графе.

В результате работы алгоритма возможны 3 ситуации:

1. Найден вектор ошибок e, удовлетворяющий решаемому уравнению;

2. Произошла стабилизация оценок на маргинальные распределения;
3. Достигнуто максимальное число итераций.

Отметим, что ни один из этих вариантов не гарантирует правильного (или неправильного) результата декодирования сообщения.

При передаче сообщений в алгоритме возможны две схемы: параллельное расписание (сначала все вершины посылают сообщения во все факторы, а затем все факторы — во все вершины) и последовательное (на каждой итерации алгоритма сообщения обновляются в случайном порядке).

Ещё одной модификацией алгоритма является демпфирование, когда сообщение обновляется выпуклой комбинацией старого сообщения и пересчитанного нового: $\mu^{t+1} = \lambda\mu_{new} + (1-\lambda)\mu^t, \lambda \in (0;1]$, иными словами, происходит экспоненциальное сглаживание значений.

В экспериментах ниже будет рассмотрено общее поведение представленного алгоритма декодирования, а также влияние различных параметров на качество его работы.

3 Эксперименты

3.1 Различные расписания и коэффициенты демпфирования

Исследуем поведение алгоритма в зависимости от расписания и коэффициента демпфирования.

Зафиксируем параметры $n = 50, k = 10, q = 0.1$ и оценим время работы алгоритма декодирования сообщений. Результаты представлены в таблице 1:

λ	Параллельное расписание	Последовательное расписание
$\frac{1}{8}$	0.1034	0.7213
$\frac{1}{4}$	0.0514	0.7252
$\frac{3}{8}$	0.0325	0.5082
$\frac{1}{2}$	0.0151	0.3162
$\frac{5}{8}$	0.0121	0.2048
$\frac{3}{4}$	0.0082	0.1389
$\frac{7}{8}$	0.0049	0.0845
1	0.0035	0.0582

Таблица 1: Среднее время работы алгоритма декодирования в секундах

Видно, что с ростом λ уменьшается время работы алгоритма, а параллельное расписание работает быстрее последовательного, в частности потому, что допускает лучшую векторизацию.

Исследуем долю стабилизировавшихся beliefs (оценок маргинальных распределений) от номера итерации. Результаты приведены на рисунках 1 - 2:

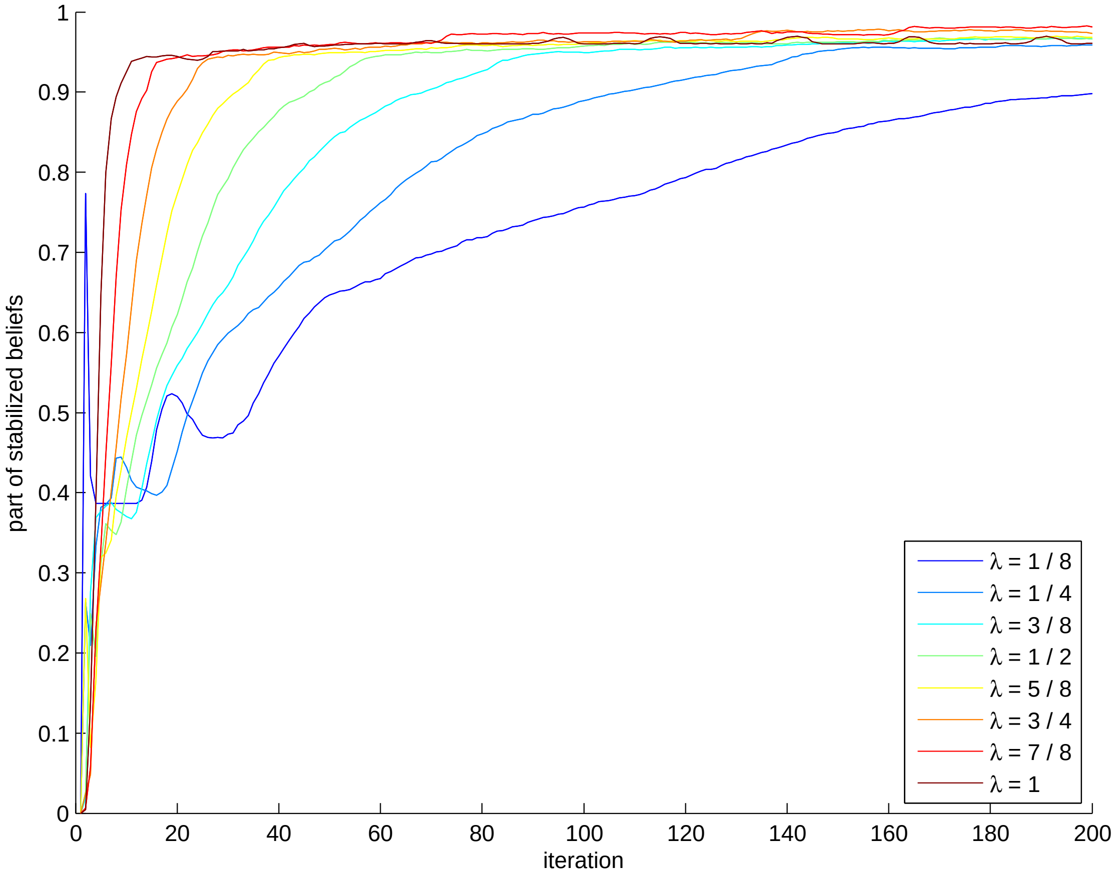


Рис. 1: Параллельное расписание

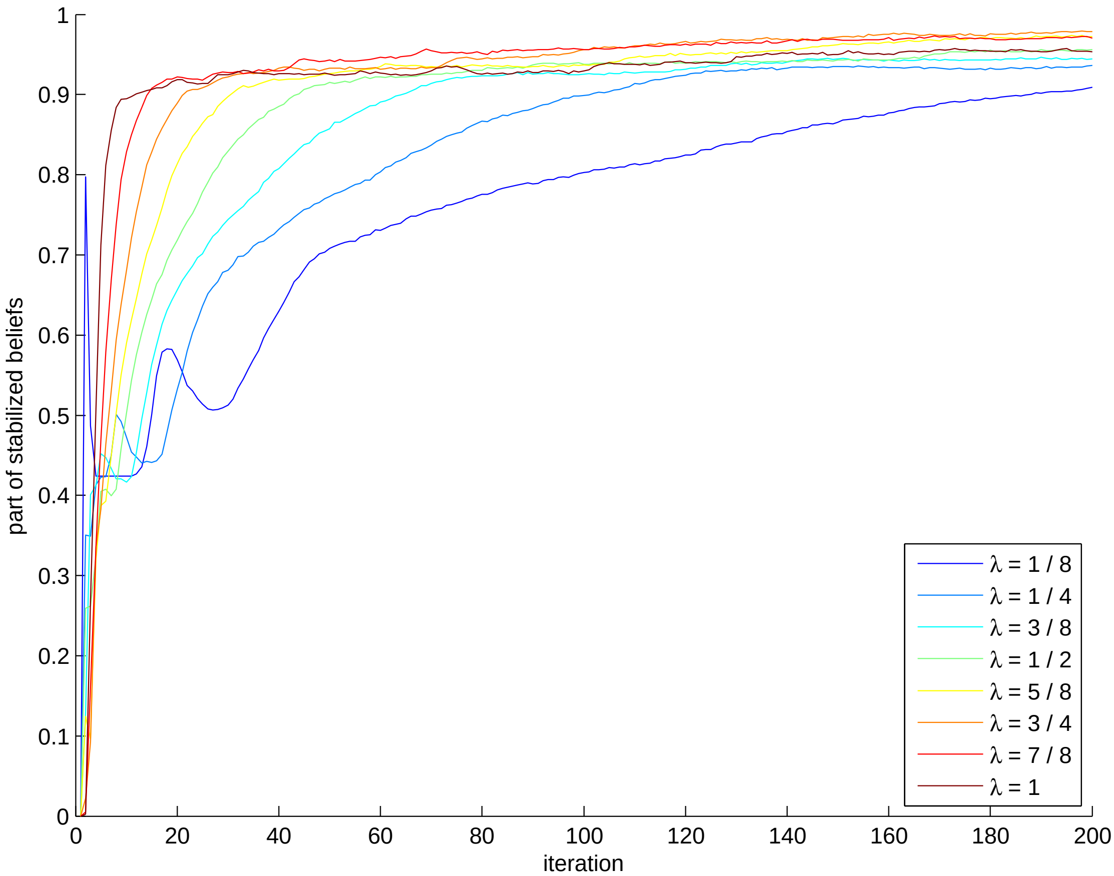


Рис. 2: Последовательное расписание

Видно, что при параллельном расписании алгоритм сходится быстрее. Также можно отметить, что выигрывают большие значения λ, однако $\lambda = 1$ — не лучшее из них, то есть учёт информации о предыдущих итерациях имеет смысл.

Заметим, что графики немонотонны, то есть делать вывод о «стабилизации» belief'а на основании сравнения последних двух значений на самом деле нельзя, оно может соответствовать локальному минимуму.

3.2 Теорема Шеннона

Теорема Шеннона

$\forall r < c$ существует код, такой что вероятность ошибки декодирования стремится к нулю $p_{err} \to 0$, когда длина блока стремится к бесконечности $n \to \infty$.

Проверим работу этой теоремы и исследуем зависимость характеристик кода (вероятности битовой и блоковой ошибки и расходимости алгоритма) от различных параметров. Вероятности будут приближены своей частотной оценкой в ходе математических испытаний методом Монте-Карло.

1. Зависимость от скорости r.

   Теорема Шеннона определяет пропускную способность канала как максимально допустимую скорость кода, при которой возможно осуществление надёжной коммуникации.

   Зафиксируем параметры $n = 200, q = 0.1, \lambda = \frac{7}{8}$ и исследуем эффективность кода (см. рис. 3):

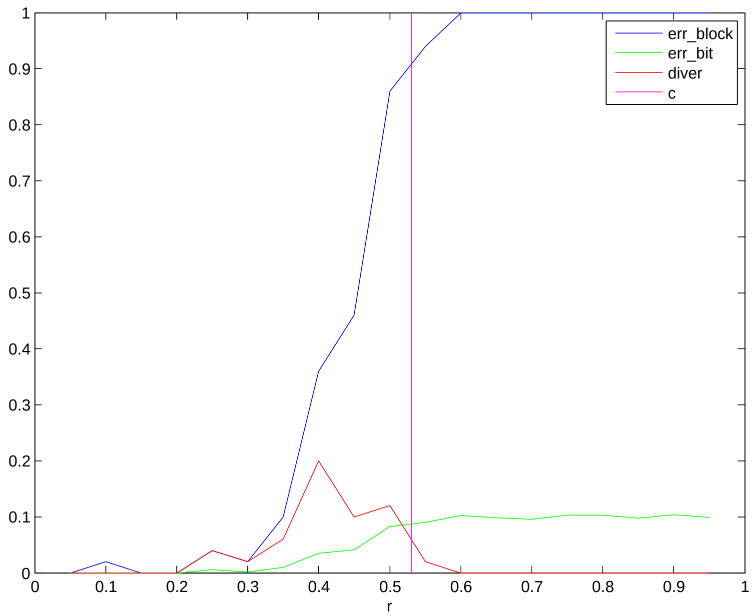
   

   Рис. 3: Характеристики кода в зависимости от r

   Действительно, после превышения пропускной способности канала блоковая ошибка становится равной 1.

2. Зависимость от длины кодового слова n.

   Теорема Шеннона предполагает, что качество кода растёт при увеличении длины кодового слова n.

   Зафиксируем параметры $r = 0.3, q = 0.1, \lambda = \frac{7}{8}$ и исследуем эффективность кода (см. рис. 4):

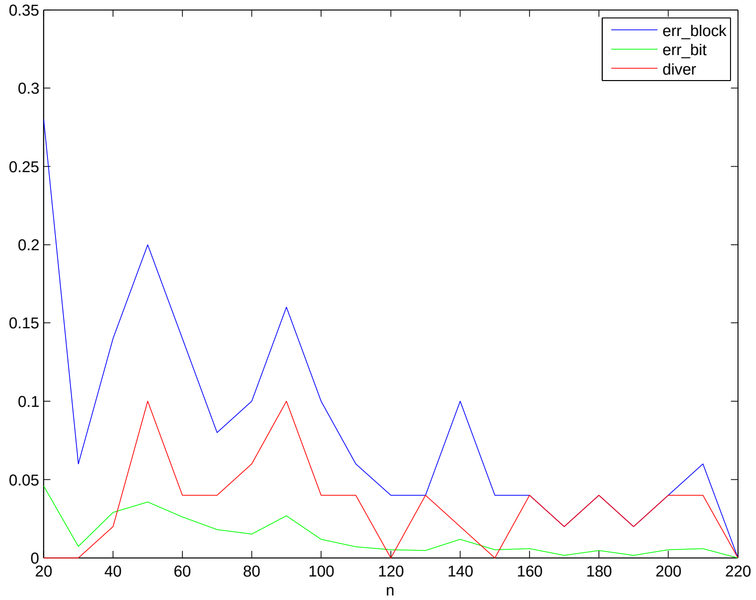


Рис. 4: Характеристики кода в зависимости от n

Действительно, в среднем с увеличением n качество кода растёт.

3. Зависимость от среднего количества единиц в столбце проверочной матрицы j.

   Одно из следствий теоремы Шеннона утверждает, что хорошими кодами являются коды со случайной проверочной матрицей H, и, в частности, качество кода должно расти с увеличением среднего количества единиц в столбце этой матрицы.

   Проверим это утверждение; зафиксируем параметры $n = 500, k = 60, q = 0.1, \lambda = \frac{7}{8}$ и исследуем эффективность кода (см. рис. 5):

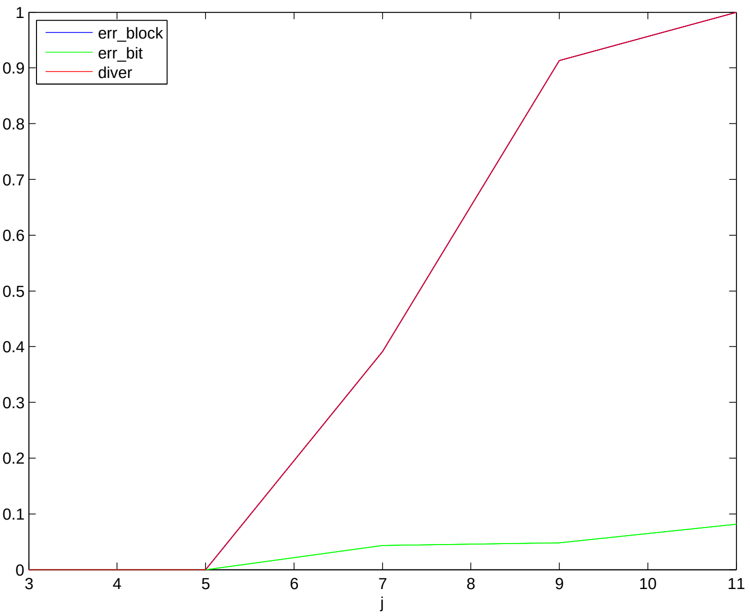


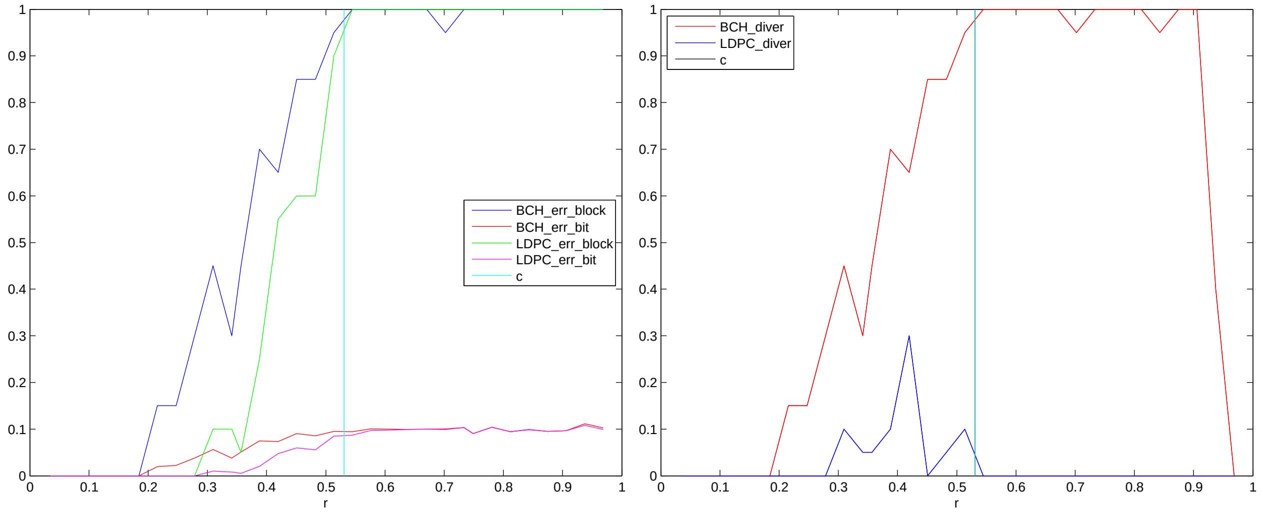
Рис. 5: Характеристики кода в зависимости от j
(err_block совпадает с diver)

Однако получен обратный эффект, и с увеличением j качество кода быстро падает. Возможно, это связано с тем, что в фактор-графе появляются циклы длины большей трёх, и это усложняет работу алгоритма.

3.3 Сравнение низкоплотностных кодов с БЧХ кодами

Коды Боуза — Чоудхури — Хоквингема [??] также являются линейными (n, k)–блоковыми кодами, сравним их с LDPC-кодами.

1. Зависимость от r.
   Зафиксируем параметры $n = 255, q = 0.1, \lambda = \frac{7}{8}$, и будем перебирать k среди таких значений, что пара (n, k) корретно задаёт код БЧХ.



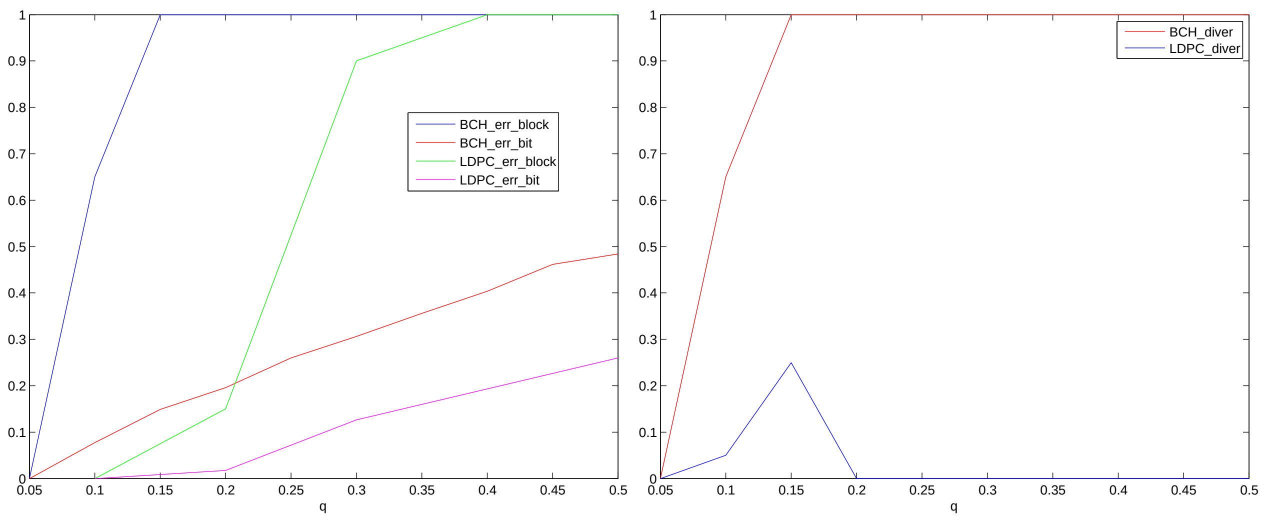
(a) Вероятности битовой и блоковой ошибок (b) Вероятности расходимости алгоритма

Рис. 6: Сравнение характеристик LDPC- и БЧХ-кодов в зависимости от r

   Ожидаемо, что при преодолении скоростью пропускной способности канала, вероятность блоковой ошибки становится равной 1.

   Видно, что LPDC-коды выигрывают у БЧХ по всем параметрам: вероятность ошибки и расходимости алгоритмов меньше. (В случае БЧХ под «расходимостью» понимаем отказ от декодирования.)

2. Зависимость от q.
   Зафиксируем параметры $n = 255, k = 87, \lambda = \frac{7}{8}$.



(a) Вероятности битовой и блоковой ошибок (b) Вероятности расходимости алгоритма

Рис. 7: Сравнение характеристик LDPC- и БЧХ-кодов в зависимости от q

Очевидно, что с ростом вероятности ошибки в канале ухудшается и качество декодирования сообщений. В рассматриваемом случае LDPC вновь превзошли БЧХ.

4 Использованная литература

[1]
[2]
[3]