УДК 519.6

Сравнение эффективности алгоритмов сверточных кодов

Е.А. Степанов, А.Н. Гамова
Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского

Среди задач поиска хороших корректирующих кодов на первое место выдвинулась задача разработки эффективных алгоритмов декодирования разных кодов. Ее решение гарантирует как степень эффективности использования каналов связи (КПД), так и уменьшение затрат на создание кодеров и декодеров, связанных с условиями выбора заданной достоверности передачи информации. По теореме Шеннона существуют такие коды, для которых при R/C вероятность ошибки будет стремиться к нулю с ростом длины кода. На практике большинство методов декодирования существенно менее эффективны, поэтому важно иметь удобные критерии эффективности применения кодирования.

В работе проведено тестирование кодов, являющихся представителями направлений, появлявшихся на пути развития теории кодирования [1,2,3]. Тесты проводились на предмет их применимости в разных условиях работы. Например, при вариации такого важного параметра, как кодовая скорость R, выбирается подходящая полоса частот 1/R, влияющая на выбор сложности аппаратуры декодирования. Отношение E_b/N_0 (битовой энергетики сигнала к спектральной плотности мощности шума в канале связи) позволяет учитывать отклонение системы от теоретического ограничения $R < C$. Как видно из графиков, с ростом кодовой скорости R граница для E_b/N_0 перемешается вверх, что является ограничением в выборе уровня избыточности кодирования [4]. В тестировании участвуют коды Витерби, порогового декодирования и многопорогового декодирования.

Для тестирования алгоритмов декодирования использовалась программа, написанная на языке C# [4]. Результаты тестирования приведены в таблицах 1 – 10.

В рамках работы были проведены серии тестов на проверку различных характеристик алгоритмов декодирования. На первом этапе мы проверили скорости работы алгоритмов при разных начальных параметрах. На вход декодерам подавались последовательности бит разной длины: 10,100,1000,10000 и т.д. Время работы алгоритмов указаны в миллисекундах. В скобках указаны степени порождающих полиномов.

Таблица 1 – Результаты работы программы с параметрами R=12 и порождающими полиномами (1,7)

	Скорость кодирования (мс)	АВ	ПД	МПД (4 итерации)
10	9	3	3	3
100	9	4	3	3
1000	14	14	6	6
10000	38	234	88	90
25000	59	701	305	310
50000	110	3088	1208	1228
100000	180	38160	6230	6283
250000	380	272866	43521	43223

Таблица 2 – Результаты работы программы с параметрами R=12 и порождающими полиномами (1,561)

	Скорость кодирования(мс)	АВ	ПД	МПД(4 итерации)
10	6	9	1	1
100	7	53	2	2
1000	11	650	4	5
10000	26	6540	77	81
25000	46	16846	315	327
50000	90	41897	958	991
100000	139	-	4463	4499

Знак «-» в последней строчке у алгоритма Витерби указывает на то, что программа перестала работать из-за слишком большого объема данных.

Таблица 3 – Результаты работы программы с параметрами R=13 и порождающими полиномами (1,15,17)

	Скорость кодирования (мс)	АВ	ПД	МПД(4 итерации)
10	7	2	2	2
100	11	3	2	2
1000	16	18	5	6
10000	33	331	81	84
25000	70	875	276	282
50000	109	3992	991	1029
100000	180	28800	4342	4528
250000	347	202524	30371	30784

Таблица 4 – Результаты работы программы с параметрами R=13 и порождающими полиномами (1, 12767,16461)

	Скорость кодирования (мс)	АВ	ПД	МПД(4 итерации)
10	7	250	2	2
100	8	1234	2	2
1000	14	10182	7	8
10000	40	-1	93	99

Как видно из приведенных выше результатов, скорость работы ПД и МПД остается линейной от длины кода, в отличие от алгоритма Витерби.

Вторым этапом проверим эффективность алгоритмов на случайных ошибках. Сначала посмотрим, количество ошибок, которые могут исправить алгоритмы в сравнении с минимальным кодовым ограничением. Скорость кода R=13, с порождающими полиномами (1,10023,200205). Данный код может исправить любые 4 ошибки на 48 бит. Исходная последовательность длиной в 50 бит. В таблице 5 указаны количество неисправленных ошибок после процедуры декодирования.

Таблица 5 – Эффективность алгоритмов на случайных ошибках

	Время работы	1	2	3	4	5	6	7	8	9	10	11	12	13
Витерби	22379	0	0	0	0	0	0	0	0	0	0	0	0	0
ПД	4	0	0	0	0	0	0	0	4	5	5	4	7	7
МПД	4	0	0	0	0	0	0	0	4	5	5	4	7	7

Как видно из таблицы, алгоритмы исправляют большее число ошибок, чем заявлено минимальным кодовым расстоянием.

Далее протестируем алгоритмы декодирования после передачи информации по каналу с шумами. В качестве шума будет выступать вероятность искажения бита. В заголовках столбцов указаны вероятности искажения бита и в скобках – количество ошибок. В самой таблице – количество неисправленных ошибок после декодирвания. В таблицах 6-8 приведены результаты декодирования информационной последовательности из 100, 1000, 10000 бит соответственно.

Таблица 6 – R=12, порождающие полиномы (1,123) и последовательность 100 бит

	Время работы	0,01 (1)	0,03 (9)	0,04 (13)	0,05 (19)	0,08 (21)	0,1 (28)	0,2 (43)	0,3 (63)	0,4 (83)	0,5 (113)
Витерби	17	0	0	0	1	0	7	20	46	45	54
ПД	3	0	0	4	3	10	10	23	35	35	59
МПД	3	0	0	4	5	10	10	17-22	38	42	57

Таблица 7 – R =12, порождающие полиномы (1,123) и последовательность 1000 бит

	Время работы	0,01 (19)	0,03 (75)	0,04 (84)	0,05 (98)	0,08 (155)	0,1 (186)	0,2 (384)	0,3 (645)	0,4 (827)	0,5 (1019)
Витерби	153	0	1	6	17	19	41	218	378	441	512
ПД	7	0	1	8	5	45	61	209	383	429	516
МПД	7	0	1	7	5	34	64	180-204	336-358	412-425	500-510

Таблица 8 – R =12, порождающие полиномы (1,123) и последовательность 10000 бит

	Время	0,01 (186)	0,03 (616)	0,04 (819)	0,05 (1015)	0,08 (1611)	0,1 (2029)	0,2 (4012)	0,3 (6065)	0,4 (8040)
Витерби	2371	0	5	39	81	243	588	2425	3546	4335
ПД	95	0	48	96	153	419	818	2343	3557	4272
МПД	98	0	38	91	136	386	712	1988-2260	3065-3406	4007-4216

В таблицах МПД при большом уровне шума записаны 2 значения через дефис. Это связано с тем, что при большом уровне шума в канале связи лучше увеличивать значения порога, чтобы кодер меньше вносил собственных ошибок. Как видно из результатов, при небольшом уровне шума (<0.1) алгоритм Витерби справляется с декодированием эффективнее чем пороговый и многопороговый декодеры. Однако при увеличении шума в канале алгоритмы работают примерно с одинаковой эффективностью, при этом ПД и МПД на порядок быстрее, что является их основным преимуществом.

И наконец посмотрим, как пороговый и многопороговый декодеры работают с «пачками ошибок».

Пусть дан код длиной в 1000 бит скоростью R=12 с порождающими полиномами (0),(0,22,41,57,72,93,99,139,147,173,217,220,234,273,283,285,296,303,328,387,388,392,416,425). В таблице приведено количество неисправленных ошибок. Будем заменять избыточные биты блоков разной длины нулями. Результаты представлены в таблице 10. Как видноиз результатов, данные алгоритмы прекрасно себя показывают при работе с «пачками ошибок».

Таблица 9 – Работа алгоритмов с «пачками ошибок», состоящие из инвертирующих бит

	10 бит	15 бит	20 бит	25 бит	30 бит	35 бит	40 бит	45 бит
МПД	0	0	0	0	0	0	0	22
ПД	0	0	0	0	0	25	22	31

Таблица 10 – Работа алгоритмов с «пачками ошибок», состоящих из нулей

	80 бит	90 бит	100 бит	110 бит
МПД	0	22	22	23
ПД	0	32	38	39

В качестве вывода можно выделить следующее:

1. скорость работы ПД и МПД на порядок выше, чем у АВ;
2. алгоритм Витерби лучше использовать при небольшой вероятности искажения бита (<= 0.1);
3. многопороговый декодер лучше использовать при большой вероятности искажения бита (>0.1);
4. 4. для МПД и ПД при маленьком шуме (<= 0.1) в канале нужно использовать порог (T = d/2), а при увеличении шума увеличивать и порог, чтобы алгоритм вносил меньше собственных ошибок в декодирование;
5. МПД очень хорошо исправляет пачки ошибок, ПД тоже исправляет их, но с меньшей надежностью, АВ исправляет только случайные ошибки;
6. все данные алгоритмы, в общем случае, исправляют ошибок больше, чем это предполагает минимальное расстояние.

Библиографический список

1. Морелос–Сарагоса, Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применения / пер. с англ. В. Б. Афанасьева –М. Техносфера. 2005. – 320 с.
2. Витерби А. Границы ошибок для сверточных кодов и асимптотически оптимальный алгоритм декодирования / Некоторые вопросы теории кодирования. М.: Мир, 1970. С.142–165.
3. Золотарев В.В. Многопороговые декодеры и оптимизационная теория декодирования / В. В. Золотарев, Ю. Б Зубарев, Г. В. Овечкин; под ред. акад. РАН В.К. Левина. М.: Горячая линия – телеком, 2012. 239 с.

4. Степанов Е.А., Гамова А.Н. Алгоритм декодирования по максимуму апостериорной вероятности. // Ломоносовские чтения на Алтае: фундаментальные проблемы науки и образования: избранные труды международной конференции, Барнаул, 11–14 ноября 2017 г. – Барнаул: Изд-во Алтайского университета, 2017. С. 818-828.

УДК 51-74, 004.6

Наука о данных в контексте статистики

Т.О. Сундукова, Г.В. Ваныкина
Тульский государственный педагогический университет им. Л.Н. Толстого, г. Тула

Наука о данных как научная дисциплина находится под влиянием информатики, математики, исследования операций и статистики, а также прикладных наук. В 1996 году термин Data Science впервые был включен в название статистической конференции (Международная федерация классификационных обществ (IFCS) «Data Science, классификация и смежные методы») [1]. Несмотря на то, что этот термин был основан статистиками в общедоступном имидже Data Science, важность компьютерных наук и бизнес-приложений часто гораздо больше подчеркивается, особенно в эпоху больших данных. В 1970-х годах идеи J.W. Tukey [2] изменили точку зрения статистики с чисто математической установки, например, статистического тестирования, на вывод гипотез из данных (исследовательская установка), то есть первично понять данные, прежде чем выдвигать гипотезы. Другим этимологическим корнем Data Science является «Обнаружение знаний в базах данных» (Knowledge Discovery in Databases – KDD) [3] с его подтемой Data Mining. KDD уже объединяет множество различных подходов к обнаружению знаний, включая индуктивное обучение, (байесовскую) статистику, оптимизацию запросов, экспертные системы, теорию информации и нечеткие множества. Таким образом, KDD является большим строительным блоком для стимулирования взаимодействия между различными областями для общей цели выявления знаний в данных.

В настоящее время эти идеи объединены в понятии Data Science, что приводит к различным определениям. Одно из наиболее полных определений науки о данных было недавно дано L. Cao в виде формулы [4, с. 28]: