БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра РЭС

Реферат на тему:

**«Энтропия сложных сообщений, избыточность источника. Цель сжатия данных и типы систем сжатия»**

МИНСК, 2009

**Энтропия сложных сообщений, избыточность источника**

Рассмотренные выше характеристики источника - количество информации и энтропия - относились к одному источнику, вырабатывающему поток независимых или простых сообщений***,*** *или* *к источнику без памяти.*

Однако в реальных условиях независимость элементарных сообщений, вырабатываемых источником, - явление довольно редкое. Чаще бывает как раз обратное - сильная детерминированная или статистическая связь между элементами сообщения одного или нескольких источников.

Например, при передаче текста вероятности появления отдельных букв зависят от того, какие буквы им предшествовали. Для русского текста, например, если передана буква "*П*", вероятность того, что следующей будет "*А*", гораздо выше, чем "*Н*", после буквы "*Ъ*" никогда не встречается "*H*" и т.д. Подобная же картина наблюдается при передаче изображений - соседние элементы изображения имеют обычно почти одинаковые яркость и цвет.

При передаче и хранении данных часто также имеют дело с несколькими источниками, формирующими статистически связанные друг с другом сообщения. Сообщения, вырабатываемые такими источниками, называются *сложными сообщениями*, а сами источники - *источниками с памятью*.

Очевидно, что при определении энтропии и количества информации в сообщениях, элементы которых статистически связаны, нельзя ограничиваться только безусловными вероятностями - необходимо обязательно учитывать также условные вероятности появления отдельных сообщений.

Определим энтропию сложного сообщения, вырабатываемого двумя зависимыми источниками (подобным же образом определяется энтропия сложного сообщения, вырабатываемого одним источником с памятью).

Пусть сообщения первого источника принимают значения ***x1, x2, x3,....xk*** с вероятностями, соответственно, ***P(x1 ), P(x2 ),..... P(xk ),*** сообщения второго - ***y1, y2,.....ym*** с вероятностями ***P(y1 ), P(y2 ),..... P(ym ).***

Совместную энтропию двух источников ***X*** и ***Y*** можно определить следующим образом:

, (1)

где ***P(xi,yj )*** - вероятность совместного появления сообщений ***xi*** и ***yj .*** Поскольку совместная вероятность ***P(xi,yj )*** по формуле Байеса определяется как

, (2)

то выражение для совместной энтропии можно записать в следующем виде:

 (3)

Так как передаче сообщения ***xi*** обязательно соответствует передача одного из сообщений (любого) из ансамбля ***Y*** , то

 (4)

и совместная энтропия ***H(X,Y)*** определится как

, (5)

где ***H ( Y/xi )*** - так называемая частная условная энтропия, отражающая энтропию сообщения ***Y*** при условии, что имело место сообщение ***xi***. Второе слагаемое в последнем выражении представляет собой усреднение ***H ( Y/xi )*** по всем сообщениям ***xi*** и называется *средней условной энтропией источника Y при условии* *передачи сообщения* ***X***. И окончательно:

***H (X,Y ) = H (X) + H (Y/X) .*** (6)

Таким образом, *совместная энтропия двух сообщений равна сумме безусловной энтропии одного из них и условной энтропии второго*.

Можно отметить следующие основные свойства энтропии сложных сообщений:

1. При *статистически независимых* сообщениях ***X*** и ***Y*** совместная энтропия равна *сумме энтропий* каждого из источников:

***H (X,Y) = H (X) + H (Y) ,*** (7)

так как ***H (Y/X) = H (Y).***

2. При *полной статистической зависимости* сообщений ***X*** и ***Y*** совместная энтропия равна *безусловной энтропии одного* из сообщений. Второе сообщение при этом информации не добавляет. Действительно, при полной статистической зависимости сообщений условные вероятности ***P(yj/xi)*** и ***P(xi/y j)*** равны или нулю, или 1, тогда

***P(xi /yj )\*log P(xi /yj ) = P(yj /xi )\*log P(yj /xi ) = 0*** (8)

и, следовательно, ***H (X,Y) = H (X) = H (Y).***

3. Условная энтропия изменяется в пределах

***0 < H (Y /X ) < H (Y).*** (9)

4. Для совместной энтропии двух источников всегда справедливо соотношение

***H (X,Y ) ≤ H (X) + H (Y),*** (0)

при этом условие равенства выполняется только для независимых источников сообщений.

Следовательно, *при наличии связи между элементарными сообщениями* *энтропия источника снижается*, причем в тем большей степени, чем сильнее связь между элементами сообщения.

Таким образом, можно сделать следующие выводы относительно степени информативности источников сообщений:

1. *Энтропия* источника и количество информации *тем больше, чем* *больше размер алфавита* источника.

2. Энтропия источника зависит от статистических свойств сообщений. *Энтропия максимальна, если сообщения источника равновероятны и статистически независимы.*

3. *Энтропия источника, вырабатывающего неравновероятные сообщения, всегда меньше максимально достижимой*.

4. *При наличии статистических связей между элементарными сообщениями (памяти источника) его энтропия уменьшается*.

В качестве примера рассмотрим источник с алфавитом, состоящим из букв русского языка ***а ,б, в,.....,ю, я***. Будем считать для простоты, что размер алфавита источника ***К = 25 = 32***.

Если бы все буквы русского алфавита имели одинаковую вероятность и были статистически независимы, то средняя энтропия, приходящаяся на один символ, составила бы

***H ( λ )max = log2 32*** = ***5 бит/букву.***

Если теперь учесть лишь различную вероятность букв в тексте (а нетрудно проверить, что так оно и есть), расчетная энтропия составит

***H ( λ )*** = ***4,39 бит/букву.***

С учетом корреляции (статистической связи) между двумя и тремя соседними буквами (после буквы “*П”* чаще встречается “*A*” и почти никогда – “*Ю*” и “*Ц*”) энтропия уменьшится, соответственно, до

***H ( λ )*** = ***3,52 бит/букву*** и ***H ( λ )*** = ***3,05 бит/букву***.

Наконец, если учесть корреляцию между восемью и более символами, энтропия уменьшится до

***H ( λ )*** = ***2,0 бит/букву***

и далее остается без изменений.

В связи с тем, что реальные источники с одним и тем же размером алфавита могут иметь совершенно различную энтропию (а это не только тексты, но и речь, музыка, изображения и т.д.), вводят такую характеристику источника, как *избыточность*

***ρи = 1 - H ( λ ) / H ( λ )max = 1 - H ( λ )/log K ,*** (11)

где ***H (λ )*** - энтропия реального источника, ***log K*** - максимально достижимая энтропия для источника с объемом алфавита в ***К*** символов.

Тогда, к примеру, избыточность литературного русского текста составит

***ρи*** = ***1 - ( 2 бита/букву )/( 5 бит/букву ) = 0,6*** .

Другими словами, при передаче текста по каналу связи каждые шесть букв из десяти передаваемых не несут никакой информации и могут безо всяких потерь просто не передаваться.

Такой же, если не более высокой ( ***ρи***= 0,9...0,95) избыточностью обладают и другие источники информации - речь, и особенно музыка, телевизионные изображения и т.д.

Возникает законный вопрос: нужно ли занимать носитель информации или канал связи передачей символов, практически не несущих информации, или же возможно такое преобразование исходного сообщения, при котором информация "втискивалась" бы в минимально необходимое для этого число символов?

Оказывается, не только можно, но и необходимо. Сегодня многие из существующих радиотехнических систем передачи информации и связи просто не смогли бы работать, если бы в них не производилось такого рода кодирование. Не было бы цифровой сотовой связи стандартов GSM и CDMA. Не работали бы системы цифрового спутникового телевидения, очень неэффективной была бы работа Internet, а уж о том, чтобы посмотреть видеофильм или послушать хорошую музыку с лазерного диска, не могло быть и речи. Все это обеспечивается эффективным или экономным кодированием информации в данных системах.

Изучению этого раздела современной радиотехники – основ теории и техники экономного, или безызбыточного, кодирования - и посвящена следующая часть нашего курса.

**Цель сжатия данных и типы систем сжатия**

Передача, хранение и обработка информации требуют достаточно больших затрат. И чем с большим количеством информации нам приходится иметь дело, тем дороже это стоит. К сожалению, большая часть данных, которые нужно передавать по каналам связи и сохранять, имеет не самое компактное представление. Скорее, эти данные хранятся в форме, обеспечивающей их наиболее простое использование, например: обычные книжные тексты, ASCII коды текстовых редакторов, двоичные коды данных ЭВМ, отдельные отсчеты сигналов в системах сбора данных и т.д. Однако такое наиболее простое в использовании представление данных требует вдвое - втрое, а иногда и в сотни раз больше места для их сохранения и полосу частот для их передачи, чем на самом деле нужно. Поэтому сжатие данных – это одно из наиболее актуальных направлений современной радиотехники.

*Таким образом, цель сжатия данных - обеспечить компактное представление данных, вырабатываемых источником, для их более экономного сохранения и передачи по каналам связи.*

Учитывая *чрезвычайную важность* процедуры экономного кодирования данных при их передаче, выделим ее из обобщенной схемы *РТС ПИ* и подробно рассмотрим в настоящем разделе нашего курса.

Ниже приведена условная структура *системы сжатия данных*:

Данные источника → Кодер → Сжатые данные → Декодер → Восстановленные данные

В этой схеме вырабатываемые источником данные определим как *данные источника*, а их компактное представление - как *сжатые данные*. Система сжатия данных состоит из *кодера* и *декодера источника.* Кодер преобразует данные источника в сжатые данные, а декодер предназначен для восстановления данных источника из сжатых данных*. Восстановленные* данные, вырабатываемые декодером, могут либо абсолютно точно совпадать с исходными *данными источника*, либо незначительно отличаться от них.

Существуют два типа систем сжатия данных:

* системы сжатия без потерь информации (неразрушающее сжатие);
* системы сжатия с потерями информации (разрушающее сжатие).

### **Сжатие без потерь информации**

*В системах сжатия без потерь декодер восстанавливает данные источника абсолютно точно, таким образом, структура системы сжатия выглядит следующим образом:*

Вектор данных **X** → Кодер → **B (X)** → Декодер → **X**

Вектор данных источника ***X***, подлежащих сжатию, представляет собой последовательность ***X***= (***x1, x2,… xn*** ) конечной длины. Отсчеты ***xi*** *-* составляющие вектора ***X -*** выбраны из конечного *алфавита данных* ***A*.** При этом размер вектора данных ***n***ограничен, но он может быть сколь угодно большим. Таким образом, источник на своем выходе формирует в качестве данных***X*** последовательность длиной ***n*** из алфавита ***A* .**

Выход кодера - сжатые данные, соответствующие входному вектору ***X,*** - представим в виде двоичной последовательности ***B(X) = ( b1,b2,…bk ),***размер которой ***k*** зависит от ***X***. Назовем ***B(X)*** кодовым словом, присвоенным вектору ***X***кодером (или кодовым словом, в которое вектор ***X*** преобразован кодером). Поскольку система сжатия - неразрушающая, одинаковым векторам ***Xl = Xm***должны соответствовать одинаковые кодовые слова ***B ( Xl ) = = B ( Xm ).***

При решении задачи сжатия естественным является вопрос, насколько эффективна та или иная система сжатия. Поскольку, как мы уже отмечали, в основном используется только двоичное кодирование, то такой мерой может служить *коэффициент сжатия* ***r****,* определяемый как отношение

**размер данных источника в битах *n log 2 (dim A)*** (12)

***r =* *= ,***

**размер сжатых данных в битах *k***

где ***dim A* -** размер алфавита данных ***A.***

Таким образом, коэффициент сжатия ***r*** ***= 2*** означает, что объем сжатых данных составляет половину от объема данных источника. Чем больше коэффициент сжатия ***r,*** тем лучше работает система сжатия данных.

Наряду с коэффициентом сжатия ***r*** эффективность системы сжатия может быть охарактеризована *скоростью сжатия* ***R,*** определяемой как отношение

***R = k/n* (**13)

и измеряемой в "количестве кодовых бит, приходящихся на отсчет данных источника". Система, имеющая *больший* коэффициент сжатия, обеспечивает *меньшую* скорость сжатия.

###

### **Сжатие с потерей информации**

*В системе сжатия с потерями (или с разрушением) кодирование производится таким образом, что декодер не в состоянии восстановить данные источника в первоначальном виде. Структурная схема системы сжатия с разрушением выглядит следующим образом:*

**X** → Квантователь → **Xq** → Неразрушающий кодер → **B (Xq)** → Декодер → **X\***

Как и в предыдущей схеме, ***X = ( x1, x2,… xn )*** - вектор данных, подлежащих сжатию. Восстановленный вектор обозначим как ***X\* = ( x1, x2,… xn ).*** Отметим наличие в этой схеме сжатия элемента, который отсутствовал при неразрушающем сжатии, - *квантователя.*

Квантователь применительно к вектору входных данных ***X*** формирует вектор ***Xq***, достаточно близкий к ***X*** в смысле среднеквадратического расстояния. Работа квантователя основана на понижении размера алфавита (простейший квантователь производит округление данных до ближайшего целого числа).

Далее кодер подвергает неразрушающему сжатию вектор квантованных данных ***Xq*** таким образом, что обеспечивается однозначное соответствие между ***Xq*** и ***B*(*Xq*)** (для ***Xlq = Xm q***выполняется условие***B (Xlq) = B (Xmq)).*** Однако система в целомостается разрушающей, поскольку двум различным векторам ***X*** может соответствовать один и тот же вектор ***X\**.**

Разрушающий кодер характеризуется двумя параметрами - скоростью сжатия ***R*** и величинойискажений ***D,*** определяемых как

##### R = k/n,

***D = (1/n) ∑(xi - xi)2*.**(14)

Параметр ***R*** характеризует скорость сжатия в битах на один отсчет источника, величина ***D*** является мерой среднеквадратического различия между ***X\**** и ***X*.**

Если имеются система разрушающего сжатия со скоростью и искажениями ***R1*** и ***D1*** соответственно и вторая система со скоростью ***R2*** и искажениями ***D2,*** то первая из них лучше, если ***R1*** ‹ ***R2*** и ***D1*** ‹ ***D2.*** Однако, к сожалению, невозможно построить систему разрушающего сжатия, обеспечивающую одновременно снижение скорости ***R*** и уменьшение искажений ***D,*** поскольку эти два параметра связаны обратной зависимостью. Поэтому целью оптимизации системы сжатия с потерями может быть либо минимизация скорости при заданной величине искажений, либо получение наименьших искажений при заданной скорости сжатия.

Выбор системы неразрушающего или разрушающего сжатия зависит от типа данных, подлежащих сжатию. При сжатии текстовых данных, компьютерных программ, документов, чертежей и т.п. совершенно очевидно, что нужно применять неразрушающие методы, поскольку необходимо абсолютно точное восстановление исходной информации после ее сжатия. При сжатии речи, музыкальных данных и изображений, наоборот, чаще используется разрушающее сжатие, поскольку при практически незаметных искажениях оно обеспечивает на порядок, а иногда и на два меньшую скорость ***R***. В общем случае разрушающее сжатие обеспечивает, как правило, существенно более высокие коэффициенты сжатия, нежели неразрушающее.

Ниже приведены ряд примеров, иллюстрирующих необходимость процедуры сжатия, простейшие методы экономного кодирования и эффективность сжатия данных.

Пример 1. Предположим, что источник генерирует цифровое изображение (кадр) размером 512\*512 элементов, содержащее 256 цветов. Каждый цвет представляет собой число из множества {0,1,2… 255}. Математически это изображение представляет собой матрицу 512х512, каждый элемент которой принадлежит множеству {0,1,2… 255}. (Элементы изображения называют пикселами).

В свою очередь, каждый пиксел из множества {0,1,2… 255} может быть представлен в двоичной форме с использованием 8 бит. Таким образом, размер данных источника в битах составит 8х512х512= 221, или 2,1 Мегабита.

На жесткий диск объемом в 1 Гигабайт поместится примерно 5000 кадров изображения, если они не подвергаются сжатию (видеоролик длительностью примерно в пять минут). Если же это изображение подвергнуть сжатию с коэффициентом r = 10, то на этом же диске мы сможем сохранить уже почти часовой видеофильм!

Предположим далее, что мы хотим передать исходное изображение по телефонной линии, пропускная способность которой составляет 14000 бит/с. На это придется затратить 21000000 бит/14000 бит/с, или примерно 3 минуты. При сжатии же данных с коэффициентом r = 40 на это уйдет всего 5 секунд !

Пример 2. В качестве данных источника, подлежащих сжатию, выберем фрагмент изображения размером 4х4 элемента и содержащее 4 цвета: R = ="красный", O = "оранжевый", Y = "синий", G = "зеленый":

|  |  |  |  |
| --- | --- | --- | --- |
| R | R | O | Y |
| R | O | O | Y |
| O | O | Y | G |
| Y | Y | Y | G |

Просканируем это изображение по строкам и каждому из цветов присвоим соответствующую интенсивность, например, R = 3, O = 2, Y = 1 и G = 0, в результате чего получим вектор данных X = (3,3,2,1,3,2,2,1,2,2,1,0,1,1,1,0).

Для сжатия данных возьмем кодер, использующий следующую таблицу перекодирования данных источника в кодовые слова (вопрос о выборе таблицы оставим на будущее):

|  |
| --- |
| *Кодер* |
| Отсчет | Кодовое слово |
| 3 | 001 |
| 2 | 01 |
| 1 | 1 |
| 0 | 000 |

Используя таблицу кодирования, заменим каждый элемент вектора ***X*** соответствующей кодовой последовательностью из таблицы (так называемое *кодирование без памяти).* Сжатые данные (кодовое слово ***B(X))*** будут выглядеть следующим образом:

***B(X)* =** ( 0,0,1,0,0,1,0,1,1,0,0,1,0,1,0,1,1,0,1,0,1,1,0,0,0,1,1,1,0,0,0).

Коэффициент сжатия при этом составит ***r*** = 32/31, или 1,03. Соответственно скорость сжатия ***R*** = 31/16 бит на отсчет.

Пример 3. Сравним два различных кодера, осуществляющих сжатие одного и того же вектора данных

***X = ABRACADABRA* .**

Первый кодер - кодер без памяти, аналогичный рассмотренному в предыдущем примере (каждый элемент вектора ***X*** кодируется независимо от значений других элементов - *кодер без памяти*). Таблица кодирования для него выглядит следующим образом:

|  |
| --- |
| ***Кодер 1*** |
| Символ | Кодовое слово |
| **A** | 0 |
| **B** | 10 |
| **R** | 110 |
| **C** | 1110 |
| **D** | 1111 |

Второй кодер при кодировании текущего символа учитывает значение предшествующего ему символа, таким образом, кодовое слово для текущего символа **A** будетразличным в сочетаниях **RA, DA** и **CA** ( иными словами, *код обладает памятью* в один символ источника):

|  |
| --- |
| ***Кодер 2*** |
| Символ, предыдущий символ | Кодовое слово |
| (A,-) | 1 |
| (B,A) | 0 |
| (C,A) | 10 |
| (D,A) | 11 |
| (A,R) | 1 |
| (R,B) | 1 |
| (A,C) | 1 |
| (A,B) | 1 |

Кодовые слова, соответствующие вектору данных **X = ABRACADABRA,** при кодировании с использованием этих двух таблиц будут иметь вид:

***B1(X)* = 01011001110011110101100,**

***B2(X)* = 10111011111011.**

Таким образом, скорость сжатия при использовании ***кодера 1*** (без памяти) составит 23/11 = 2,09 бита на символ данных, тогда как для ***кодера 2*** - 13/11 = =1,18 бита на символ. Использование второго кодера, следовательно, является более предпочтительным, хотя он и более сложен.

**ЛИТЕРАТУРА**

1. Лидовский В.И. Теория информации. - М., «Высшая школа», 2002г. – 120с.
2. Метрология и радиоизмерения в телекоммуникационных системах. Учебник для вузов. / В.И. Нефедов, В.И. Халкин, Е.В. Федоров и др. – М.: Высшая Школа, 2001 г. – 383с.
3. Цапенко М.П. Измерительные информационные системы. – М.: Энергоатом издат, 2005. - 440с.
4. Зюко А.Г. , Кловский Д.Д., Назаров М.В., Финк Л.М. Теория передачи сигналов. М: Радио и связь, 2001 г. –368 с.
5. Б. Скляр. Цифровая связь. Теоретические основы и практическое применение. Изд. 2-е, испр.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003 г. – 1104 с.