ЛЕКЦИИ ШКОЛЫ
АНАЛИЗА ДАННЫХ ЯНДЕКСА

Н.К. Верещагин, Е.В. Щепин

Информация, кодирование и предсказание

Яндекс

ИЗДАНИЕ ОСУЩЕСТВЛЯЕТСЯ
ПРИ ПОДДЕРЖКЕ КОМПАНИИ «ЯНДЕКС»

Н. К. Верещагин, Е. В. Щепин

Информация, кодирование и предсказание

Москва
ФМОП
МЦНМО
2012

УДК 519.72
ББК 32.81
В31

Издание осуществлено при поддержке Фонда Математического образования и Просвещения

Верещагин Н. К., Щепин Е. В.

В31 Информация, кодирование и предсказание. — М.: ФМОП, МЦНМО, 2012. — 236 с.

ISBN 978-5-904696-05-4 (ФМОП)
ISBN 978-5-94057-920-5 (МЦНМО)

Предлагаемая книга — это одновременно учебник и оригинальная монография по теории информации. Две независимые друг от друга части, составляющие книгу, написаны авторами на основе собственных лекций, читающихся в Школе анализа данных Яндекса.

Автор первой части, Е. В. Щепин, рассматривает понятия теории информации как базу для решения задач машинного обучения, и прежде всего — задач построения классификатора по эмпирическим данным. Специальное внимание автор уделяет изучению случаев многомерных ограниченных данных, когда прямые методы оценки функций распределения вероятностей неприменимы. Обсуждение этих вопросов редко встречается в работах по теории информации. В предлагаемой книге изложение доведено до описания практических методов.

Во второй части, написанной Н. К. Верещагиным, исследуются задачи о поиске на базе понятия информации по Хартли. В этой части описаны различные применения теории колмогоровской сложности (сложности описаний), даны основы логики знаний и теории коммуникационной сложности. К теоретическому материалу прилагается множество задач для самостоятельного решения.

В обеих частях отводится много места основам классической теории информации Шеннона и её применению к кодированию информации. В первой части это изложение ведется с позиций конструирования алгоритмов решения проблем, во второй части большое внимание уделено концептуальным аспектам классической теории Шеннона.

Книга завершается дополнением, взятым из выдающейся книги М. М. Бонгарда «Проблема узнавания» (1967), где с позиций теории информации изучается вопрос об оценке степени истинности описания. Эта важная тема, непосредственно примыкающая к рассматриваемым в книге проблемам, служит подтверждением перспективности теории информации для развития новых методов анализа данных.

ББК 32.81

978-5-904696-05-4 (ФМОП)
978-5-94057-920-5 (МЦНМО)

Оглавление

Часть вторая

Предисловие к книге Евгения Щепина и Николая Верещагина «Информация, кодирование и предсказание»

Интерес к применению методов теории информации к проблемам распознавания образов, и, более общо, к машинному обучению и анализу данных, возник очень давно. Норберт Винер в книге «Кибернетика» в 1947г. отмечал важность применения теоретико-информационного подхода для построения распознающей машины. Через 20 лет в 1967 Михаил Бонгард в книге «Проблема узнавания» впервые дал конкретный пример такого применения. Он показал, что с помощью такого подхода можно естественным образом количественно анализировать качество работы обученного распознающего автомата. Двумя годами позже, в 1969 году, примерно эту же проблему количественного анализа качества узнавания и догадки рассмотрел Сатоши Ватанабе в книге «Узнавать и догадываться: количественное изучение вывода и информация». В последние годы сильно возрос интерес к применению теории информации для исследования задач машинного обучения и анализа данных, особенно в связи с необходимостью решения этих задач для поиска среди очень больших массивов текстовой информации. Эта область приложений, в свою очередь, породила исследования, расширяющие базовые представления и предлагающие новые методы во внутренней структуре теории информации. Теория информации, можно сказать, переживает сейчас новый период развития. Хорошей иллюстрацией этого является предлагаемая Вам книга Евгения Щепина и Николая Верещагина. Задуманная как учебное пособие по теории информации, она параллельно с изложением фундаментальных моделей и методов традици-

онной теории информации ясно обозначает современные тенденции в ее развитии. Несмотря на небольшой объем, книга содержит модели и алгоритмы, определяющие новые направления развития теории информации, не представленные в современных публикациях по теории информации. Читатель, работающий в области анализа данных и желающий использовать в своих исследованиях теорию информации, найдет в этой книге много направляющих идей.

Книга состоит из двух частей. Первая написана Евгением Щепиным, вторая — Николаем Верещагиным. Для полноты изложения современного теоретико-информационного подхода к проблемам анализа данных, в конце книги в качестве дополнения мы приводим текст раздела из упомянутой выше монографии М. Бонгарда, одного из пионеров и классиков теории обучения машин.

Обе части книги написаны независимо и их можно читать независимо. Они обьединены тем, что обе нацелены на самые новейшие приложения теории информации, как сугубо инженерные, так и теоретические (ряд интереснейших задач, как отмечалось, в книге описан впервые). Изложение в обеих частях почти не требует вузовских знаний, однако требует определенного навыка точного мышления. Однако и при отсутствии такого навыка изучение материала книги доступно благодаря его структурной организации, позволяющей «не часто возвращаться назад» при чтении. Авторы обеспечили это подбором очень интересных вопросов-примеров, и в особенности тем, что многие из этих примеров даны в форме простых исследовательских задач, решение которых не только способствует лучшему усвоению материала, но и позволяет приобрести навыки придумывания и решения новых задач.

В первой части большое внимание уделено задачам, непосредственно примыкающим к проблемам машинного обучения, особенно связанным с построением классификатора на основе примеров. В частности, разбираются трудные задачи оценки функции плотности вероятностей в пространствах очень большой размерности по малым выборкам. Я назвал бы эти исследования Е. Щепина основами нового теоретико-информационного напрвления в построении классификаторов. Уже предложенные процедуры представляют большую практическую ценность и являются источником, вдохновляющим на новые глубокие теоретические разработки, особенно в деле поиска регуляризаторов для оценки редко наблюдаемых событий.

Одна из особенностей второй части заключается в том, что в ней

в рамках общей схемы теории информации рассматриваются задачи оценки колмогоровской сложности (в современной зарубежной литературе эти задачи часто объединяют термином «алгоритмическая теория информации»). В традиционных учебниках по классической теории информации, и в особенности в учебниках, ориентированных на инженеров, эта проблематика не рассматривается, несмотря на то, что сами задачи активно исследуются (по ним не только имеется большой поток научных статей, но и публикуются монографии). Поэтому этот относительно маленький раздел второй части представляется важным, особенно для студентов и специалистов, которые впервые изучают теорию информацию и методы ее применения. Николай Верещагин построил изложение этого раздела таким образом, что он может служить базой и для чтения других быстро развивающихся направлений в анализе данных, например, для теории машинного обучения, для которой критическими являются (1) изучение существующих оценок гарантий качества моделей, построенных на анализе примеров, и (2) разработка новых оценок. Исследования по решению проблем машинного обучения на основе моделей колмогоровской сложности — активная область, в развитии которой Н. Верещагин непосредственно участвует. Поэтому ему удалось, несмотря на краткость, настолько выразительно описать это перспективное направление, что впервые изучающий книгу читатель имеет возможность включиться в эти исследования.

В обеих частях, как уже говорилось, подробно разбираются и классические задачи теории информации. Особенно подробно разбираются задачи представления и кодирования данных для их сжатого хранения и передачи, а также задачи возможности и оценки предсказания в различных условиях.

В описании классических задач теории информации в текстах первой и второй части имеются дублирования, но они полезны, так как по-разному построены и дополняют друг друга. Особенно хорошо этот разный подход к изложению проявляется в подборе задач-упражнений. Если Е. Щепин старается показать, как сложные процедуры можно заменить эффективными аппроксимациями и на каких идеях можно строить эти хорошие аппроксимации, то Н. Верещагин ищет наиболее понятное, простое и короткое описание точных процедур.

Самое большое "дублирование"имеет место в описании базовых понятий — энтропии и информации и их свойств. В обеих частях читатель найдет ясное и полное их обсуждение. И тем не менее, и даже в

этих описаниях читатель найдет полезные дополнения. В частности, интересным представляется, что Н. Верещагин наиболее детально описывает алгебраическую природу необходимых конструкций, а Е. Щепин подчеркивает их вероятностную природу. Н. Верещагин уделяет большое внимание асимптотическим свойствам методов, а Е. Щепин — свойствам, которые проявляются на конечных конструкциях. В итоге читатель получает обьемную картину всего здания теории.

Книга Е. Щепина и Н. Верещагина — не просто техническое пособие к лекциям. Получилась новая интересная монография по основам и приложениям теории информации, которая будет полезна и студентам, и ученым, работающим в различных областях, использующих модели теории информации.

И. Б. Мучник
август 2011 г.

Часть первая

Введение

Курс нацелен на изучение способов применения теории информации в практических задачах распознавания образов. Хотя все требуемое от теории информации по существу является вариацией одной и той же формулы энтропии, многие из тех, кто изучал теорию информации, часто бывают поставлены в тупик простейшими вопросами, возникающими при ее применении. Например, большинство выпускников механико-математического факультета МГУ не понимают, как посчитать количество информации, которую сумма цифр трехзначного числа несет об их произведении.

Интуитивное понятие информации, возникающее из обычного языка, хорошо согласуется с определением количества информации, с которым оперирует наука. Развитию интуитивного понятия способствует решение задач и разбор доказательств основных теорем.

Доказательства теорем в курсе не отличаются строгостью, ведь все эти теоремы и так хорошо известны, их цель — дать читателю лучше прочувствовать формулировки и показать в работе основные приемы.

Данный курс лекций может быть использован как для первичного ознакомления с теорией информации, так и для более глубокого ее изучения. В последних двух лекциях излагаются новые экспериментальные подходы к применению теории информации в распознавании образов, возникшие у автора при работе в группе разработчиков Яндекса.

Глава 1. Информационная емкость символа

Информация может быть заключена в словах, числах, цветах, звуках и многом другом, но в конечном счете всякую информацию можно выразить словами любого человеческого языка. Всякий язык имеет свой *алфавит* — множество символов (букв, цифр, знаков препинания и т. д.). Последовательности символов представляют *слова* языка. Вопрос, который мы решим сегодня — сколько информации может нести одно слово. Единицей измерения информации служит *бит*: один ответ на вопрос типа да-нет. Количество битов информации, которое содержит данное слово, означает, на сколько вопросов типа да-нет может отвечать это слово. Например, информационная емкость символа трехбуквенного алфавита (то есть однобуквенного слова в языке с алфавитом из трех символов), как мы увидим в дальнейшем, приблизительно равна 1.5849625 битам.

1.1 Двоичные коды

Алфавит языка компьютера состоит из нуля и единицы, а словами являются последовательности нулей и единиц, называемые *двоичными словами*. Число элементов двоичного слова называется его *длиной*. Память компьютера состоит из ячеек, называемых *битами*, в каждом из которых записан либо ноль, либо единица. Биты объединяются в восьмерки, называемые *байтами*. Таким образом, каждый байт памяти компьютера хранит двоичное слово длины 8. Как следует из нижеприведенной леммы, всего имеется $2^8 = 256$ различных *байтовых слов*.

Лемма 1.1. *Общее количество двоичных слов длины n равно 2^n.*

Доказательство. Доказательство проводится индукцией по длине последовательности. Для $n = 1$ утверждение верно. Пусть уже доказано,

что число двоичных слов длины n равно 2^n. Все двоичные слова длины $n+1$ делятся на два типа: начинающиеся с нуля и начинающиеся с единицы. Число слов каждого типа совпадает с числом слов длины n, то есть в силу предположения индукции равно 2^n. Тогда число последовательностей длины $n+1$ равно $2^n + 2^n = 2^{n+1}$. □

Количества байтовых слов хватает, чтобы представить все буквы русского и английского алфавитов, цифры, знаки препинания и все символы, изображенные на клавиатуре компьютера. Например, стандартным двоичным кодом русской буквы «а» является «10100000», код цифры «0» — «00110000», пробела — «00100000».

Таким образом, любое предложение русского языка может быть представлено с помощью двоичных кодов таким образом, что займет ровно столько байт, сколько символов (включая пробелы и знаки препинания) оно содержало.

1.2 Оптимальное кодирование

Одна из важных практических задач — как поместить в компьютер как можно больше информации.

Например, мы хотим внести в компьютер данные переписи населения России таким образом, чтобы они занимали как можно меньше памяти. Рассмотрим для начала только данные о поле граждан. Для кодирования пола человека достаточно одного бита — например, можно обозначать женский пол нулем, а мужской единицей. Получается очень компактный способ кодирования, более того, как мы сейчас докажем, лучшего способа кодирования пола не существует.

Двоичным кодированием множества M называется отображение c, ставящее в соответствие каждому элементу множества $x \in M$ двоичное слово $c(x)$ — его код. Кодирование называется *инъективным*, если коды различных элементов множества M различны.

Лемма 1.2. *Множество M допускает двоичное инъективное кодирование с длиной кода, равной n, в том и только том случае, когда число элементов множества M не превосходит 2^n.*

Доказательство. Так как число двоичных слов длины n равно 2^n по лемме 1.1, то теорема сводится к следующему очевидному утверждению: одно множество можно инъективно отобразить в другое в том и

только том случае, когда число элементов первого множества не превышает числа элементов второго множества. □

Доказанная лемма позволяет давать эффективные оценки минимально необходимого объема памяти компьютера для запоминания различного рода анкетных данных. Например, для запоминания информации о поле всех граждан России (население России будем считать равным 140 миллионам человек), мы должны зарезервировать не менее 140 миллионов бит памяти компьютера. Действительно, пусть d — какой-то способ кодирования полов. Предположим, что все записи упорядочены по алфавиту. Обозначим через c кодирование, ставящее 0 в соответствие женскому полу и 1 — мужскому. Результат переписи, кодированной по принципу «женский – 0, мужской – 1», представляет собой двоичную последовательность длины $140 \cdot 10^6$. Заранее мы не можем исключать никакого варианта переписи, поэтому все возможные результаты переписи должны инъективно кодироваться кодом d. Поэтому dc^{-1} должно инъективно кодировать множество всех двоичных слов длины $140 \cdot 10^6$. Так как число таких слов $2^{140000000}$ и по доказанной выше лемме их нельзя инъективно закодировать словами длины меньшей, чем $140 \cdot 10^6$, то, следовательно, нашу информацию нельзя разместить в памяти, заняв менее 140 миллионов битов.

1.3 Объединение в блоки

А теперь рассмотрим задачу кодирования результатов голосования. Имеется три варианта голосования: «за», «против» и «воздержался». Каждый результат можно было бы кодировать с помощью двухбитовых слов — «11», «00», «01». При этом одна комбинация, «10», осталась неиспользованной. При таком кодировании n результатов голосования займут объем $2n$ бит памяти. Но этот результат можно улучшить и поместить все результаты, заняв лишь $5n/3$ бит памяти, то есть потратив в среднем на один вариант голосования $5/3$ бита. Чтобы это сделать, мы будем объединять результаты голосования в тройки. Тогда число вариантов голосования для тройки бюллетеней равняется $3 \cdot 3 \cdot 3 = 27 < 2^5$ и может быть кодировано пятибитовыми словами. Так как число троек равно $n/3$, а каждая тройка занимает 5 бит памяти, то мы получаем обещанную оценку $5n/3$. Но и этот результат можно улучшить, если объединять в блоки по пять бюллетеней сразу. В этом

случае число возможных исходов голосования для блока составляет $3^5 = 243 < 256 = 2^8$ и потому результат по пятерке бюллетеней записывается в один байт. Таким образом, в среднем на бюллетень придется $8/5$ бита памяти. С другой стороны, так как $3^2 > 2^3$, результаты голосования по n бюллетеням имеют более чем $2^{3n/2}$ вариантов и поэтому не могут занимать меньше $3n/2$ бит памяти, то есть не существует способа кодирования, при котором на один бюллетень отводится менее 1.5 бит.

Рассмотрим теперь k-блочное разбиение бюллетеней. Пусть m — такое натуральное число, что $2^m < 3^k < 2^{m+1}$. Тогда, с одной стороны, мы можем кодировать результаты голосования, потратив в среднем на бюллетень не более $\frac{m+1}{k}$ бит, с другой стороны — не можем кодировать, потратив менее $\frac{m}{k}$ бит. k-блочное кодирование с ростом k дает сколь угодно близкие к оптимальному (1.5 бита на бюллетень) результаты.

1.4 Бит и трит

Одной из первых вычислительных машин в СССР была основанная на троичной системе счисления «Сетунь». Представим себе винчестер троичной машины, у которого элементарная ячейка памяти имеет не два, как у современных компьютеров, а три состояния — записанная в нем информация кодируется символами алфавита из трех символов (можно взять, например, 0, 1, 2), а информационнная емкость измеряется в *тритах*, выражающих информационную емкость символа троичного алфавита. Каково же соотношение бита и трита? Предположим, у нас есть «бинарный» винчестер и мы хотим переписать информацию с него на винчестер троичной машины. Какую емкость в тритах он должен иметь? Не заботясь о практической реализуемости алгоритма, мы можем решить вопрос следующим образом: содержимое двоичного винчестера, представляющее собой последовательность нулей и единиц, можно трактовать как натуральное число в двоичной системе счисления. Точно так же содержимое винчестера троичной машины представляется последовательностью нулей, единиц и двоек, которую можно трактовать как натуральное число, записанное в троичной системе счисления. Перекодирование двоичной последовательности в троичную можно реализовать как переход от одной системе счисления к другой — запись числа, представляющего собой содержимое двоичного винчестера, в троичной системе счисления и будет

являться перекодированием информации для троичной машины. Аналогично, содержимое винчестера троичной машины можно трактовать как натуральное число в троичной системе счисления, перевод которого в двоичную систему представляет собой обратное, декодирующее, отображение. Итак, пусть емкость двоичного винчестера равна n бит, а емкость троичного m трит. Тогда максимальное число, представленное содержимым двоичного винчестера, равно $2^n - 1$, а содержимым троичного винчестера — $3^m - 1$. Если $2^n < 3^m$, то любое содержимое двоичного винчестера перекодируется в троичное число, которое может поместиться на троичном винчестере. Поэтому в данном случае мы приходим к заключению, что n бит должно быть меньше, чем m трит. Если же $3^m < 2^n$, то содержимое троичного винчестера перекодируется в число, помещающееся на двоичном винчестере. Если n и m таковы, что выполнено неравенство $2^n < 3^m < 2^{n+1}$, то m трит заключено между n и $n+1$ битом. Логарифмирование последнего неравенства по основанию 2 дает $n < m \log_2 3 < n + 1$, откуда мы видим, что разность между тритом и $\log_2 3$ не превосходит $\frac{1}{m}$. Ввиду произвольности m отсюда следует, что трит следует считать равным $\log_2 3$.

1.5 Формула Хартли

Будем говорить, что *информативность* символа k-элементного алфавита M меньше чем a в том и только том случае, если при некотором n множество всех слов длины n из алфавита M может быть кодировано не более чем na-битовыми словами. Будем называть информативностью символа наибольшее число I, для которого информативность символа не меньше чем I.

Теорема 1.1 (Хартли)**.** *Информативность символа k-элементного алфавита равна* $\log_2 k$

Доказательство. Пусть M — k-элементный алфавит. Обозначим через M^n множество всех слов длины n в алфавите M. Мощность этого множества равна k^n. Пусть s таково, что

$$2^s < k^n \leqslant 2^{s+1} \tag{1.1}$$

Тогда, с одной стороны, можно кодировать M^n с помощью $s + 1$-битовых последовательностей, поэтому информативность элемента ал-

фавита M не превышает $\frac{s+1}{n}$, а с другой стороны, невозможно кодировать слова длины n с помощью s-битовых слов, откуда следует, что информативность символа алфавита M не меньше $\frac{s}{n}$. Логарифмируя неравенство (1.1) по основанию 2, получаем неравенство $s < n \log_2 k < s + 1$, откуда следует, что разность между информативностью символа и $\log_2 k$ не превосходит $\frac{1}{n}$ для любого n. Отсюда и следует требуемое равенство. □

1.6 Задача сжатия файла

Одной из необходимых программ на современном компьютере является программа сжатия-разжатия файлов — архиватор. Всякий архиватор выполняет две операции: во-первых, он пакует файлы, преобразуя их в файлы меньшего размера, чтобы сохранить на диске больше свободного пространства, а во-вторых, он делает обратное преобразование — распаковывает упакованные им файлы, восстанавливая их в точности такими, какими они были до архивации. Таким образом, всякий архиватор осуществляет взаимно-однозначное отображение множества всех файлов на его подмножество, состоящее из упакованных файлов.

На чем же основана возможность сжатия файлов? Всякий файл можно рассматривать как двоичное слово. Текстовые файлы более естественно рассматривать как байтовое или двухбайтовое (юникод) слово. В любом случае файл можно представлять себе как слово в некотором алфавите.

Первая возможность для сжатия файла заключается в оптимизации двоичного кодирования его алфавита. Например, если файл представляет собой последовательность десятичных цифр, закодированных стандартным образом, при этом каждой цифре соответствует байт, то мы можем на основании проведенных выше рассуждений закодировать его так, чтобы на символ в среднем приходилось примерно $\log_2 10 \approx \frac{10}{3}$ бита, то есть сжать его почти в два с половиной раза.

1.7 Равночастотное кодирование

Рассмотрим некоторое слово (файл) m-символьного алфавита. Пусть числа $n_1, \ldots, n_m$ выражают количество встретившихся в нем символов алфавита: n_i — количество вхождений в слово i-го символа алфавита.

Николай Константинович Верещагин
Евгений Витальевич Щепин

ИНФОРМАЦИЯ, КОДИРОВАНИЕ И ПРЕДСКАЗАНИЕ

Подписано в печать 6.02.2012. Формат $60\times90\,{}^{1}\!/_{16}$.
Бумага офсетная №1. Печать офсетная. Печ. л. 15. Тираж 1500 экз.

Издательство Московского центра
непрерывного математического образования.
119002, Москва, Большой Власьевский пер., д. 11. Тел. (499) 241-74-83.

Отпечатано с готовых диапозитивов в ООО «Принт Сервис Групп».
Москва, ул. Борисовская, д. 14.