Численное моделирование общей задачи распределения количества bitmap-индексов

**Человек:** Предметом исследования является математическая модель в виде системы рекуррентных интегральных соотношений, описывающая распределение количества единичных интервалов, в которых произошло по крайней мере одно событие случайного потока с произвольной функцией распределения. Рассмотрены многочисленные аспекты численной реализации этой системы, такие как выбор метода обращения преобразования Лапласа, численное интегрирование вблизи точек разрыва, обеспечение устойчивости вычислений, контроль достоверности результатов, учет особенностей машинной арифметики действительных чисел. Особое внимание уделяется связи результатов вычислений с семантикой прикладной задачи, решение которой они представляют. Методологией исследования являются теория вероятностей (виды и свойства распределений), методы вычислительной математики (численное интегрирование, интерполяция, обращение преобразования Лапласа), программная реализация математической модели и выполнение вычислительных экспериментов. Основными выводами проведенного исследования являются валидность и счетная реализуемость построенной автором математической модели, обоснование получения численного решения для произвольного распределения потока случайных событий. Новизна исследования заключается в полученном численном решении задачи распределения количества bitmap-индексов для распределения Вейбулла, гамма-распределения, логнормального распределения и других, представлении и анализе различных зависимостей, таких как плотность распределения количества индексов и среднее число индексов для заданной длины интервала.

**Key words:** точка разрыва, квадратурные формулы, численное интегрирование, численные методы, преобразование Лапласа, плотность распределения вероятностей, устойчивость вычислений, биномиальное распределение, метод Симпсона, несобственный интеграл

=================================

**FastText\_KMeans\_Clean:** С общих позиций разработки программного обеспечения было бы довольно привлекательно написать универсальную программу, реализующую (1)-(5), в которой переменной частью были бы только реализации функции распределения F ( x ) и плотности распределения f ( x ) исходного потока случайных событий, отладить эту программу на эталонной задаче экспоненциального распределения, где точное решение заранее известно [11] и верить в то, что для любого распределения она даст нам такое же приемлемое по точности решение задачи. for(int i=0;i<l;i++) y=y/10;. Для многих распределений функция F(x) содержит экспоненту в отрицательной степени от величины аргумента, соответственно с ростом n F ( n - x ) быстро становится настолько малым, что вычисления теряют точность. Рис.5. Гамма-распределение, r=0.5. Логнормальное распределение .

**Key words part:** 0.43333333333333335

=================================

**FastText\_KMeans\_Raw/:** if (fabs(x-y)<=eps) ….; /\*x и y "равны"\*/. Для многих распределений функция F(x) содержит экспоненту в отрицательной степени от величины аргумента, соответственно с ростом n F ( n - x ) быстро становится настолько малым, что вычисления теряют точность. Вычисления велись для n=1.50. Точность выполнения (8) для всех расчетов – расхождение с единицей, не превышающее 0.01. Но так или иначе здесь вряд ли требуется высокая точность, т.к. исходное распределение случайного потока событий может удовлетворять гипотезе о соответствии некоторому известному теоретическому распределению только с изрядной долей условности. Рис.5. Гамма-распределение, r=0.5. Логнормальное распределение . Распределение Рэлея .

**Key words part:** 0.4

=================================

**FastText\_PageRank\_Clean/:** 1.1. Контроль достоверности. y=(float)(floor(y+0.5));. int combinations(int a, int b) {. if (b==0) return(1);. if (b==1) return(a);. return(round(res));. Общая проблема понятна. 2. Анализ результатов.

**Key words part:** 0.3

=================================

**FastText\_PageRank\_Raw/:** for(int i=0;i<l;i++) y=y\*10;. y=(float)(floor(y+0.5));. for(int i=0;i<l;i++) y=y/10;. int combinations(int a, int b) {. if (b==0) return(1);. if (b==1) return(a);. for(i=0;i<a-big;i++) res=res\*(big+i+1)/(i+1);. return(round(res));.

**Key words part:** 0.3

=================================

**Mixed\_ML\_TR/:** С общих позиций разработки программного обеспечения было бы довольно привлекательно написать универсальную программу, реализующую (1)-(5), в которой переменной частью были бы только реализации функции распределения F ( x ) и плотности распределения f ( x ) исходного потока случайных событий, отладить эту программу на эталонной задаче экспоненциального распределения, где точное решение заранее известно [11] и верить в то, что для любого распределения она даст нам такое же приемлемое по точности решение задачи. В первом случае для некоторого значения n производится расчет всех функций Hk ( n , x ) для k =1,… n , затем это же делается для значения n +1 и так далее до N . Для этого распределения . Можно написать код, вычисляющий эту функцию (а она тоже является интегралом), а можно также в [4, с. 1073] найти подходящую для нашего случая ( формулу. применив которую, получим окончательно. - для i=0: -0.167, 1.5, -4.333, 4.0. - для i=1: 4.0, -32.0, 76.0, -48.0. - для i=2: -13.5, 94.5, -189.0, 108.0. - для i=3: 10.667, -64.0, 117.333, -64.0. Заметим, что задача выбора наилучшего численного метода обращения преобразования Лапласа для уравнения (2) до конца не изучена и может составить отдельную тему серьезного математического исследования. Результаты экспериментов показали, что распределение количества интервалов, в которых произошли события, хоть и не является в общем случае биномиальным, достаточно хорошо приближается таковым при произвольном распределении входного случайного потока, а среднее количество таких интервалов можно вычислять как линейную функцию от n .

**Key words part:** 0.7

=================================

**MultiLingual\_KMeans/:** В первом случае для некоторого значения n производится расчет всех функций Hk ( n , x ) для k =1,… n , затем это же делается для значения n +1 и так далее до N . Для этого распределения . Можно написать код, вычисляющий эту функцию (а она тоже является интегралом), а можно также в [4, с. 1073] найти подходящую для нашего случая ( формулу. применив которую, получим окончательно. - для i=0: -0.167, 1.5, -4.333, 4.0. - для i=1: 4.0, -32.0, 76.0, -48.0. - для i=2: -13.5, 94.5, -189.0, 108.0. - для i=3: 10.667, -64.0, 117.333, -64.0. Заметим, что задача выбора наилучшего численного метода обращения преобразования Лапласа для уравнения (2) до конца не изучена и может составить отдельную тему серьезного математического исследования.

**Key words part:** 0.6333333333333333

=================================

**Multilingual\_PageRank/:** Эта подзадача реализации (1)-(5) является наиболее трудной. y=(float)(floor(y+0.5));. 1.7. F(x) не является элементарной функцией. Данное распределение является heavy tailed. рафики изображены на рис.7, . Распределение Накагами . Распределение Накагами. Распределение является heavy-tailed.

**Key words part:** 0.36666666666666653

=================================

**RuBERT\_KMeans\_Without\_ST/:** Достаточно оригинальный подход, основанный на элементарном свойстве преобразования Лапласа и построении его конечно-разностной аппроксимации, предложен в [3], но и он, увы, на большинстве примеров страдает тем же недостатком – численной неустойчивостью, о чем упоминает и сам автор этой работы. for(int i=0;i<l;i++) y=y/10;. Конкретно, при расчетах он брался равным 0.0001, а искомый интеграл считался как сумма двух численных интегралов: первый в пределах от 0.0001 до 0.01 с шагом 0.0001, второй – от 0.01 до заданного верхнего предела с шагом 0.01. Вообще, если функция плотности распределения в нуле обращается в бесконечность, следует тщательно контролировать, чтобы при численной реализации (1)-(5) нигде не потребовалось ее вычисление в нуле. Вычисления велись для n=1.50. Точность выполнения (8) для всех расчетов – расхождение с единицей, не превышающее 0.01. Но так или иначе здесь вряд ли требуется высокая точность, т.к. исходное распределение случайного потока событий может удовлетворять гипотезе о соответствии некоторому известному теоретическому распределению только с изрядной долей условности.

**Key words part:** 0.6

=================================

**RuBERT\_KMeans\_With\_ST/:** Это различие становится особенно очевидным, если взять, например, обычное распределение и распределение с "тяжелым хвостом" (heavy-tailed) [12, 16]. Причина этого, как правило, такова: счетная формула метода предусматривает суммирование знакопеременного ряда с быстро возрастающим по модулю n -ым членом. for(int i=0;i<l;i++) y=y\*10;. Вычисления велись для n=1.50. Точность выполнения (8) для всех расчетов – расхождение с единицей, не превышающее 0.01. Но так или иначе здесь вряд ли требуется высокая точность, т.к. исходное распределение случайного потока событий может удовлетворять гипотезе о соответствии некоторому известному теоретическому распределению только с изрядной долей условности. Рис.11. Экспоненциальное и распределения Вейбулла.

**Key words part:** 0.5

=================================

**RUBERT\_page\_rank\_Without\_ST/:** Общая проблема понятна. 2. Анализ результатов. Графики (рис.6) соответствуют параметрам µ=-0.5, σ=1. рафики изображены на рис.7, . Распределение Накагами.

**Key words part:** 0.36666666666666653

=================================

**RUBERT\_page\_rank\_With\_ST/:** for(int i=0;i<l;i++) y=y\*10;. for(int i=0;i<l;i++) y=y/10;. if (b==1) return(a);. Распределение Вейбулла . Если же картина меняется.

**Key words part:** 0.36666666666666653

=================================

**RUSBERT\_KMeans\_Without\_ST/:** Эта система имеет следующий вид:. При неэлементарной F ( x ) 1- F ( n - x ) выражается каким-то образом через некоторую специальную функцию, и нужно найти подходящее приближение для этой специальной функции, подстановка которого и позволит произвести необходимые сокращения. Численное решение с большой точностью (максимальное отклонение не превышает 0.001), но, как уже было сказано, это совершенно не означает, что такая точность будет иметь место для других распределений. Рис.5. Гамма-распределение, r=0.5. Логнормальное распределение .

**Key words part:** 0.43333333333333335

=================================

**RUSBERT\_KMeans\_With\_ST/:** Например, мы знаем, что для эталонной задачи при фиксированном n распределение Hk ( n ) является биномиальным, следовательно, если мы получаем результаты, похожие на биномиальное распределение, это в какой-то степени говорит в пользу правильности расчетов. Вот код, позволяющий вычислить количество сочетаний, если оно само по себе не является сверхбольшим числом. Экспоненциальное распределение (рис.1). Численное решение с большой точностью (максимальное отклонение не превышает 0.001), но, как уже было сказано, это совершенно не означает, что такая точность будет иметь место для других распределений.

**Key words part:** 0.4666666666666667

=================================

**RUSBERT\_page\_rank\_Without\_ST/:** Эта система имеет следующий вид:. 1. Численная реализация системы (1)-(5). 1.2. Внешний и внутренний цикл. if (b==0) return(1);. Данное распределение является heavy tailed.

**Key words part:** 0.43333333333333335

=================================

**RUSBERT\_page\_rank\_With\_ST/:** 1.2. Внешний и внутренний цикл. Данное распределение является heavy tailed. Распределение Накагами . Распределение Накагами. Распределение является heavy-tailed.

**Key words part:** 0.36666666666666653

=================================

**Simple\_PageRank/:** В первой из них рассмотрены различные аспекты численной реализации (1)-(5), во второй – приведены примеры решений, полученных с помощью этой системы для различных видов распределений, оценена их достоверность и рассмотрены характерные особенности. С общих позиций разработки программного обеспечения было бы довольно привлекательно написать универсальную программу, реализующую (1)-(5), в которой переменной частью были бы только реализации функции распределения F ( x ) и плотности распределения f ( x ) исходного потока случайных событий, отладить эту программу на эталонной задаче экспоненциального распределения, где точное решение заранее известно [11] и верить в то, что для любого распределения она даст нам такое же приемлемое по точности решение задачи. В связи с этим при реализации (2) и (3) нет необходимости для каждой f(x) выполнять отдельную аналитическую работу по вычислению прямого преобразования Лапласа, а делать это всегда одним способом – "в лоб", по определению, путем численного вычисления несобственного интеграла для различных значений s. Чтобы обрисовать следующую проблему, возникающую при реализации численного интегрирования применительно к задаче (1)-(5), рассмотрим такой весьма простой определенный интеграл как . Конкретно, при расчетах он брался равным 0.0001, а искомый интеграл считался как сумма двух численных интегралов: первый в пределах от 0.0001 до 0.01 с шагом 0.0001, второй – от 0.01 до заданного верхнего предела с шагом 0.01. Вообще, если функция плотности распределения в нуле обращается в бесконечность, следует тщательно контролировать, чтобы при численной реализации (1)-(5) нигде не потребовалось ее вычисление в нуле. Итак, показана практическая применимость и вычислительная реализуемость системы уравнений (1)-(5) – как на эталонном примере, так и на большом количестве произвольных распределений с различными свойствами, в том числе и для heavy-tailed распределений, а также предложена методика оценки достоверности результатов.

**Key words part:** 0.6666666666666666

=================================

**TextRank/:** В работе [11] была выведена система интегральных рекуррентных соотношений, дающая теоретическое решение задачи о распределении количества единичных интервалов, в которые попадает хотя бы одно событие из некоторого случайного потока, заданного произвольной функцией распределения. Прежде всего, совершенно очевидно, что ответ на этот вопрос лежит в области численных методов, поскольку уже преобразование Лапласа, являющееся лишь частью этих уравнений, для подавляющего большинства известных распределений (пожалуй, только кроме экспоненциального и равномерного) не является элементарной функцией. С общих позиций разработки программного обеспечения было бы довольно привлекательно написать универсальную программу, реализующую (1)-(5), в которой переменной частью были бы только реализации функции распределения F ( x ) и плотности распределения f ( x ) исходного потока случайных событий, отладить эту программу на эталонной задаче экспоненциального распределения, где точное решение заранее известно [11] и верить в то, что для любого распределения она даст нам такое же приемлемое по точности решение задачи. В связи с этим при реализации (2) и (3) нет необходимости для каждой f(x) выполнять отдельную аналитическую работу по вычислению прямого преобразования Лапласа, а делать это всегда одним способом – "в лоб", по определению, путем численного вычисления несобственного интеграла для различных значений s. Конкретно, при расчетах он брался равным 0.0001, а искомый интеграл считался как сумма двух численных интегралов: первый в пределах от 0.0001 до 0.01 с шагом 0.0001, второй – от 0.01 до заданного верхнего предела с шагом 0.01. Вообще, если функция плотности распределения в нуле обращается в бесконечность, следует тщательно контролировать, чтобы при численной реализации (1)-(5) нигде не потребовалось ее вычисление в нуле. Результаты экспериментов показали, что распределение количества интервалов, в которых произошли события, хоть и не является в общем случае биномиальным, достаточно хорошо приближается таковым при произвольном распределении входного случайного потока, а среднее количество таких интервалов можно вычислять как линейную функцию от n .

**Key words part:** 0.7333333333333333

=================================

**TF-IDF\_KMeans/:** – функция распределения случайного потока событий;. В первой из них рассмотрены различные аспекты численной реализации (1)-(5), во второй – приведены примеры решений, полученных с помощью этой системы для различных видов распределений, оценена их достоверность и рассмотрены характерные особенности. В первом случае для некоторого значения n производится расчет всех функций Hk ( n , x ) для k =1,… n , затем это же делается для значения n +1 и так далее до N . Рис.3. Распределение Вейбулла, r=0.5. Гамма-распределение . Получена формула, по которой можно оценить значение углового коэффициента этой функции.

**Key words part:** 0.4666666666666667

=================================

**Текст:** В работе [11] была выведена система интегральных рекуррентных соотношений, дающая теоретическое решение задачи о распределении количества единичных интервалов, в которые попадает хотя бы одно событие из некоторого случайного потока, заданного произвольной функцией распределения. Прикладное значение данной задачи [10] заключается в том, что она описывает количество bitmap-индексов, попадающих в результаты запроса за некоторый интервал времени, где случайный поток событий интерпретируется как поток занесения новых записей в базу данных. Эта система имеет следующий вид:. . (1). . . (2). . . (3). . (4). . (5). . Напомним смысл обозначений:. - плотность распределения для момента наступления последнего события случайного потока на интервале от 0 до n , при условии, что события происходили ровно на k единичных подинтервалах;. – функция распределения случайного потока событий;. – функция плотности распределения случайного потока событий;. – математическое ожидание интервала между событиями потока (первый момент);. - преобразование Лапласа для функции .. Если функция известна, то можно вычислить суммарную вероятность наличия ровно k индексов на интервале длины n. . (6). . и среднее количество индексов на интервале длины n. . (7). Последняя характеристика уже имеет инженерное значение для оценки производительности поисковых запросов.. Какова же практическая ценность (1)-(5), можно ли и как пользоваться ими для проведения расчетов? Прежде всего, совершенно очевидно, что ответ на этот вопрос лежит в области численных методов, поскольку уже преобразование Лапласа, являющееся лишь частью этих уравнений, для подавляющего большинства известных распределений (пожалуй, только кроме экспоненциального и равномерного) не является элементарной функцией. Это обстоятельство предопределяет логическое разбиение работы на две части. В первой из них рассмотрены различные аспекты численной реализации (1)-(5), во второй – приведены примеры решений, полученных с помощью этой системы для различных видов распределений, оценена их достоверность и рассмотрены характерные особенности.. . 1. Численная реализация системы (1)-(5). Хорошо известно, что от математически безупречной аналитической формулы или уравнения до доведения их до верифицируемых чисел зачастую пролегает дистанция немалого размера. И выполнение этой работы требует от математика-вычислителя знаний и усилий никак не меньших, а иногда и больших, чем вывод этих формул и уравнений от математика-теоретика. Более того, если теоретик опирается для получения результата на теоретические познания и строгую формальную логику, для вычислителя этого иногда оказывается недостаточно. Здесь нередки опора на опыт, интуицию и принятие решения в условиях частичной неопределенности, что делает сложную расчетную задачу, так же как и теоретическую, задачей творческой. Бывают нередки различные эвристические приемы, которые активно работающие профессионалы откровенно называют «вычислительные хитрости» [6]. Поскольку теоретическим основам и искусству применения численных методов научил меня в свое время совсем недавно ушедший из жизни мой научный руководитель и выдающийся знаток этой области математики профессор Л.П.Фельдман [13, 14], пользуюсь здесь возможностью почтить его память.. Содержимое первой части настоящей работы во многом предопределяется удивительно точными словами из [1], которые присутствуют и в более ранних изданиях этого классического труда: «Не следует думать, что совершенное знание математики, численных методов и навыки работы с ЭВМ позволяют сразу решить любую прикладную математическую задачу. Во многих случаях требуется «доводка» методов, приспособление их к решению конкретных задач. При этом типична обстановка, когда используются методы, применение которых теоретически не обосновано… при выборе метода решения задачи и анализе результатов приходится полагаться на опыт предшествующего решения задач, на интуицию и сравнение с экспериментом и при этом приходится отвечать за достоверность результата. Поэтому для успеха в работе необходимы развитое неформальное мышление, умение рассуждать по аналогии, дающие основания ручаться за достоверность результата там, где с позиций логики и математики, вообще говоря, ручаться нельзя. … При численном решении конкретных трудных задач, математик действует как естествоиспытатель, полагаясь во многом лишь на опыт и «правдоподобные» рассуждения. Крайне желательно, чтобы такая эмпирическая работа подкреплялась теоретическими разработками методов, аккуратной проверкой качества методов на контрольных задачах с известным решением или частным сравнением с экспериментом ». Эти слова, как мы далее убедимся, полностью применимы к нашей задаче. Перейдем к рассмотрению ее конкретики.. . 1.1. Контроль достоверности. С общих позиций разработки программного обеспечения было бы довольно привлекательно написать универсальную программу, реализующую (1)-(5), в которой переменной частью были бы только реализации функции распределения F ( x ) и плотности распределения f ( x ) исходного потока случайных событий, отладить эту программу на эталонной задаче экспоненциального распределения, где точное решение заранее известно [11] и верить в то, что для любого распределения она даст нам такое же приемлемое по точности решение задачи. Однако, численный метод – не сортировка массива, и реальность развеивает такие надежды – результаты счета очень сильно зависят от особенностей функции f ( x ) . Программа, дающая практически идеальное решение для функции из эталонной задачи, терпит крах или выдает заведомо абсурдные результаты уже на первых шагах вычислений для другой функции. Это различие становится особенно очевидным, если взять, например, обычное распределение и распределение с «тяжелым хвостом» (heavy-tailed) [12, 16]. Как же оценивать достоверность вычислений для произвольных f ( x ), когда о точном решении ничего не известно? В первую очередь, на помощь приходят различные качественные оценки, которые позволяют простыми рассуждениями и грубой прикидкой либо отвергнуть очевидно неверные результаты, либо признать, что бросающихся в глаза противоречий нет. Например, мы знаем, что для эталонной задачи при фиксированном n распределение Hk ( n ) является биномиальным, следовательно, если мы получаем результаты, похожие на биномиальное распределение, это в какой-то степени говорит в пользу правильности расчетов. В целом, очень важно также, чтобы за числами, полученными по формулам численных методов, математик не терял их смысла и постоянно соотносил их с семантикой решаемой прикладной задачи, задавая себе вопрос: с точки зрения здравого смысла такое возможно? Но достоверность расчетов все же как-то надо суметь оценить не только качественно, но и количественно, иначе цена им будет невысока. По счастью, для нашей задачи такая естественная оценка существует, т.к., исходя именно из смысла задачи, должно выполняться равенство. . (8). Именно по величине отклонения суммы из левой части (8) от единицы можно судить о достоверности и точности решения задачи, а по изменению этого отклонения с ростом n – об устойчивости вычислений. Если точность выполнения (8) не устраивает вычислителя, он может поэкспериментировать с различными параметрами численной реализации (1)-(5) (о которых далее пойдет речь) и подобрать такие, которые дают максимальную точность. Следует учитывать и тот факт, что численное вычисление интеграла от таблично построенной функции Hk ( n , x ) тоже вносит погрешность.. . 1.2. Внешний и внутренний цикл. Алгоритмически существуют два способа счета по формулам (1)-(5) – с внешним циклом по n и внутренним по k и наоборот. Остановимся на каждом из них подробнее. Предположим, что расчет по n ведется до некоторого максимального значения N . В первом случае для некоторого значения n производится расчет всех функций Hk ( n , x ) для k =1,… n , затем это же делается для значения n +1 и так далее до N . Во втором случае для некоторого значения k производится расчет всех функций Hk ( n , x ) для n = k ,… N , затем это же делается для значения k +1 и так далее до N . С точки зрения точности или общего объема вычислений ни один способ не является более предпочтительным по отношению к другому, но здесь вступают в силу соображения несколько иного характера:. контроль достоверности вычислений по формуле (8). Вычислитель хочет для каждого промежуточного значения n сразу же проверить выполнимость равенства (8) и в случае неудовлетворительного результата вовремя прервать заведомо бесперспективные вычисления. В первом случае для проверки (8) для некоторого n = c потребуется вычислить c ( c +1)/2 табличных функций, во втором случае Nc - c ( c +1)/2 табличных функций. Разница особенно велика для малых c , а ведь именно с них придется начинать;. параллелизм. Обратим внимание на то, что в (1)-(5) нет зависимости Hk ( n , x ) от. Hk -1 ( n , x ) . Это позволяет применить параллелизм вычислений по k на каждом шаге внешнего цикла по n и написать многопоточную программу с шириной параллелизма, согласующейся с количеством процессорных ядер конкретной компьютерной архитектуры. Зависимость же Hk ( n , x ) от Hk ( n -1, x ) существует, что делает распараллеливание для второго способа вычислений невозможным.. В силу указанных обстоятельств предпочтение было отдано первому способу.. . 1.3. Прямое преобразование Лапласа. Чтобы реализовать уравнения (2) и (3) необходимо вычислить сначала прямое, а затем обратное преобразование Лапласа. Располагая такими фундаментальными справочниками как [2] и [4], было бы заманчиво пойти по следующему пути: получить функцию прямого преобразования, затем записать выражение и в разделе обратных преобразований в [2] попытаться найти это выражение или похожее на него. Проблема, однако, в том, что практически для всех известных распределений, кроме экспоненциального, преобразование Лапласа от функции плотности распределения не является элементарной функцией. И если соответствующий интеграл и удается найти в справочнике, он представляет собой некоторое выражение, содержащее одну или несколько специальных функций, вычисляемых через «неберущийся» интеграл, который все равно придется брать численно. Проиллюстрируем это утверждение на примере распределения Парето.. Для этого распределения . Соответственно, преобразование Лапласа имеет вид . Сделав подстановку y = sx , сведем его к интегралу . В [4, с. 332] для этого интеграла находим формулу, применив которую получим. где Wa,b(x) – специальная функция, называемая функцией Уиттакера. Можно написать код, вычисляющий эту функцию (а она тоже является интегралом), а можно также в [4, с. 1073] найти подходящую для нашего случая ( формулу. . применив которую, получим окончательно. . К сожалению, мы сделали круг: полученный интеграл такой же, как и интеграл, входящий в определение преобразования Лапласа (один легко сводится к другому). И такая ситуация, как уже отмечено выше, типична.. В связи с этим при реализации (2) и (3) нет необходимости для каждой f(x) выполнять отдельную аналитическую работу по вычислению прямого преобразования Лапласа, а делать это всегда одним способом – «в лоб», по определению, путем численного вычисления несобственного интеграла для различных значений s. Реализация же численного интегрирования – это уже следующая задача, которая применительно к (1)-(5) будет рассмотрена нами в п.1.5.. . 1.4. Обратное преобразование Лапласа. Эта подзадача реализации (1)-(5) является наиболее трудной. Литература по методам численного обращения преобразования Лапласа имеется. Это в первую очередь хорошо известные труды [7] и [8], а в наши дни в этой области плодотворно работает российский математик профессор В.М.Рябов [9]. Трудность же заключается в том, что при всей математической обоснованности и безупречности описанных в этих трудах методов, применить их на практике очень и очень непросто ввиду их численной неустойчивости – результаты либо слишком далеки от точных значений на эталонных примерах, либо вообще быстро достигают астрономических значений. Причина этого, как правило, такова: счетная формула метода предусматривает суммирование знакопеременного ряда с быстро возрастающим по модулю n -ым членом. И хотя «в идеале» вычитание большого числа из большого должно согласно формуле дать «нормальное» число, этого не происходит – неизбежная потеря точности при вычислении n -го члена приводит к быстрому накоплению ошибки. Мы попадаем в своеобразные ножницы: если ограничиться совсем небольшим количеством членов ряда – вычисление обратного преобразования будет очень неточным, увеличение же этого количества приводит к уже упомянутой неустойчивости. В качестве примера приведем тригонометрический метод [8]. В нем оригинал вычисляется по формуле:. . где σ – некоторый параметр метода, зависящий от значения оригинала, в котором он вычисляется. Указанное явление мы как раз наблюдаем в формуле для коэффициентов cn.. Достаточно оригинальный подход, основанный на элементарном свойстве преобразования Лапласа и построении его конечно-разностной аппроксимации, предложен в [3], но и он, увы, на большинстве примеров страдает тем же недостатком – численной неустойчивостью, о чем упоминает и сам автор этой работы.. Для того, чтобы все-таки принять решение, следует учесть некоторые свойства изображений и оригиналов, с которыми нам нужно работать:. 1) если есть изображение плотности распределения, то его значение в нуле равно единице, а изображение терпит разрыв в нуле. Это не должно смущать, т.к., например, для изображения C/s существует вполне «нормальный» оригинал .. 2) исходя из (2) и (3), оригинал обращенного изображения потребуется вычислять только для аргументов между нулем и единицей, что избавляет от проблемы накопления погрешности, характерной для многих методов обращения при больших значениях аргумента.. В итоге в результате множества численных экспериментов наилучшую точность при данных условиях показал метод, основанный на вычислении интеграла Меллина с помощью квадратурной интерполяционной формулы с равноотстоящими узлами [8, гл. 4]. Формула имеет следующий вид:. . где есть коэффициенты разложения многочлена по степеням 1/p:. . Таблица значений этих коэффициентов в [8] отсутствует, но рекуррентные формулы для их расчета легко выводятся.. Пусть необходимо вычислить коэффициенты для многочлена степени R . Обозначим – коэффициент частичного произведения степени z при слагаемом степени j, 1≤z≤R, 0≤j≤z, . Тогда:. . В нашем случае при вычислении коэффициентов для многочлена. …, , а значения будут искомыми значениями .. Для проведения вычислений было выбрано R=3. Легко проверить, что коэффициенты получаются следующие (по возрастанию степеней):. - для i=0: -0.167, 1.5, -4.333, 4.0. - для i=1: 4.0, -32.0, 76.0, -48.0. - для i=2: -13.5, 94.5, -189.0, 108.0. - для i=3: 10.667, -64.0, 117.333, -64.0. Заметим, что задача выбора наилучшего численного метода обращения преобразования Лапласа для уравнения (2) до конца не изучена и может составить отдельную тему серьезного математического исследования. Дело в том, что во всех известных источниках, рассматривающих эту тему, примеры применения методов берутся для задач механики и электродинамики, т.е. для дифференциальных уравнений и уравнений с частными производными. Применению же к задачам теории вероятностей, когда искомый оригинал является плотностью распределения некоторой случайной величины, а изображение представляет собой суперпозицию изображений других распределений, внимание практически не уделяется.. . 1.5. Численное интегрирование. Для реализации формул (2), (3), (4) необходимо решить задачу численного вычисления интегралов – как с конечным (для (2) и (3), а также (8)), так и с бесконечным верхним пределом (прямое вычисление преобразования Лапласа и (4)). Правда, для (4) интеграл можно вычислить в пределах от 0 до n, а затем вычесть полученное значение из единицы. В качестве рабочего метода был использован метод Симпсона [1], для которого при выбранном шаге интегрирования h элементарная площадь вычисляется по формуле:. . (9). Функции табулировались с шагом 0.01, внутренний интеграл в (2) вычислялся с тем же шагом, а для внешнего интеграла в (2) шаг, соответственно, был равен 0.02. Несмотря на простоту и хорошую апробированность метода, даже здесь возникают важные вычислительные «нюансы», на которых сто ит остановиться.. Первый из них – это хорошо известные вычислителям особенности арифметики действительных чисел, которые имеют отличия в сравнении с арифметикой чисел целых. Так, действительные числа нельзя сравнивать на точное равенство, которое, скорее всего, не выполнится никогда. Такое сравнение следует заменять проверкой на непревышение модуля разности между ними некоей заданной точности вычислений. На языке C это будет выглядеть примерно так:. float eps=1E-6;. if (fabs(x-y)<=eps) …..; /\*x и y «равны»\*/. Но это еще не все. Предположим, что вычисляется интеграл с целыми пределами интегрирования a и b и шагом 0.01. Переменная интегрирования x инициализируется значением a , на каждом шаге цикла инкрементируется выражением x = x + h , а условием завершения цикла является некоторая близость значений x и b . Может возникнуть ситуация (она тем вероятнее, чем больше расстояние между a и b ), когда условие завершения цикла никогда не выполнится, x превысит b и пойдет дальше, а частичная интегральная сумма будет расти до бесконечности. Причина этого явления заключается в том, что присваивание x = x + h не выполняется точно, а с некоторой погрешностью. Если интервал интегрирования велик по отношению к выбранному шагу к концу интервала эта погрешность может накопиться настолько, что условие завершения цикла не выполнится, а если даже и выполнится, промежуточные «искаженные» значения x все равно повлияют на точность результата. Решением проблемы является гасить эту погрешность либо на каждом шаге, либо с какой-то определенной частотой путем округления значения x до количества знаков после запятой, которое имеется в шаге, т.е. при шаге 0.01 до двух. На языке C такой код может выглядеть следующим образом (y – округляемое число, l – количество знаков после запятой):. for(int i=0;i<l;i++) y=y\*10;. y=(float)(floor(y+0.5));. for(int i=0;i<l;i++) y=y/10;. return(y);. Чтобы обрисовать следующую проблему, возникающую при реализации численного интегрирования применительно к задаче (1)-(5), рассмотрим такой весьма простой определенный интеграл как . Его легко вычислить аналитически, значение равно 2. Однако, попытка посчитать его численно по формуле Симпсона приведет к падению кода уже на первом шаге вычислений – в точке 0 подынтегральная функция терпит разрыв и обращается в бесконечность. Для функций плотности распределения такая ситуация – не редкость, она имеет место, например, для распределения Вейбулла и гамма-распределения с параметром меньше единицы, а также для распределения Фишера с параметром меньше двух (все эти распределения далее в работе будут рассмотрены). Методы вычисления таких интегралов с особенностями рассмотрены в [1, гл. 3]. При реализации (1)-(5) в этом случае применяется сгущение сетки вблизи особенности подынтегральной функции, а нижний предел интегрирования заменяется каким-то достаточно близким к нулю числом. Конкретно, при расчетах он брался равным 0.0001, а искомый интеграл считался как сумма двух численных интегралов: первый в пределах от 0.0001 до 0.01 с шагом 0.0001, второй – от 0.01 до заданного верхнего предела с шагом 0.01. Вообще, если функция плотности распределения в нуле обращается в бесконечность, следует тщательно контролировать, чтобы при численной реализации (1)-(5) нигде не потребовалось ее вычисление в нуле.. . 1.6. Порядок вычислений. Обрисуем проблему опять-таки на элементарном примере. Пусть требуются вычислить количество сочетаний . Зная формулу, по которой вычисляется количество сочетаний , можно, не прибегая ко помощи компьютера, сразу же сказать, что результат равен 100. Вопрос: а как реализовать вычисление количества сочетаний программно? Реализация этой формулы «в лоб» путем вычисления трех факториалов не даст значение 100 для описанного примера, а приведет к быстрому падению кода из-за переполнения, т.к. 100! и 99! вычислить невозможно. Программист должен вникнуть в суть формулы и оптимизировать количество операций, проведя все возможные сокращения множителей. Вот код, позволяющий вычислить количество сочетаний, если оно само по себе не является сверхбольшим числом. Этот код не дает «убегать» в бесконечность промежуточным результатам вычислений и является предельно экономичным по количеству операций.. int combinations(int a, int b) {. float res;. int i,big;. if (b==0) return(1);. if (b==1) return(a);. if (b>a-b) big=b;. else big=a-b;. res=1.0;. for(i=0;i<a-big;i++) res=res\*(big+i+1)/(i+1);. return(round(res));. }. Общая проблема понятна. Отношение двух чисел согласно своей математической семантике является «нормальным» числом, но числитель и знаменатель или очень велики (что не позволяет их вычислить вообще), либо очень малы, близки к нулю (что приводит к потере значимости). Чтобы получить в этой ситуации правильный результат, следует провести аналитически все возможные сокращения, чтобы вычисление «плохих» сомножителей не попало в программный код. Нетрудно видеть, что именно такую ситуацию мы можем получить для выражения. . (10). в формуле (2). Для многих распределений функция F(x) содержит экспоненту в отрицательной степени от величины аргумента, соответственно с ростом n F ( n - x ) быстро становится настолько малым, что вычисления теряют точность. Между тем, по виду (10) можно предположить, что числитель и знаменатель достаточно близки по значению, и их отношение является вполне вычислимым. При программной реализации требуется просто не вычислять это отношение в «лоб», через функцию F ( x ), а предварительно выполнить преобразование, индивидуальное для каждого распределения. Примеры:. 1) экспоненциальное распределение, . Отношение (10) равно .. 2) распределение Вейбулла, . Отношение (10) равно экспоненте от значения . При не слишком больших k вычисление разности проблемы не представляет, если же k само по себе велико, то это уже совсем другая проблема.. Подобный прием можно применить и для отношения входящего в (3).. 1.7. F(x) не является элементарной функцией. При реализации приема, описанного в предыдущем разделе, могут возникнуть проблемы, когда функция распределения F ( x ) не является элементарной. Такая ситуация в теории вероятностей встречается нередко, например, для гамма-распределения или распределения Фишера. В этом случае сокращаемая составляющая все равно существует, только в аналитически неявном виде, и ее необходимо как-то выделить. Типовой план может быть таким. При неэлементарной F ( x ) 1- F ( n - x ) выражается каким-то образом через некоторую специальную функцию, и нужно найти подходящее приближение для этой специальной функции, подстановка которого и позволит произвести необходимые сокращения. Рассмотрим, как это может происходить на примере гамма-распределения. Это распределение является распределение Пирсона типа III [5], для него . Заменой t = y θ данное выражение приводится к виду. . (11). Таким образом, задача сводится к замене неполной верхней гамма-функции неким выражением, в котором явным образом присутствовала бы показательная составляющая. Сложность этой задачи существенным образом зависит от значения k. Если оно целое и невелико, (11) можно вычислить непосредственно. Так, в случае k=2 имеем для (10) выражение . В случае k=0.5 можно воспользоваться известной формулой [15]. , где erfc (x) – дополнительная функция ошибок, так же как и erf(x) входящая в стандартную математическую библиотеку языка С. Взяв в [15] приближенную формулу для erfc (x), для (10) можно в итоге получить , где в числителе и знаменателе стоят некоторые полиномы третьего порядка с табличными коэффициентами. В общем же случае, для произвольного k можно пойти различными путями. Например, в [4, c.955] приведена формула. . С ее помощью задача избавления от экспоненциальной составляющей решается, а входящий в нее несобственный интеграл можно заранее табулировать, чтобы не перевычислять его многократно при одном и том же x . Так или иначе в каждом конкретном случае то или иное вычислительное решение при условии наличия достаточного количества справочной математической информации всегда может быть найдено.. Безусловно, перечисленными проблемами всё обилие нюансов численной реализации столь комплексной системы как (1)-(5) не исчерпывается, но не будем перегружать работу мелкими техническими деталями, которые и так хорошо известны программистам-вычислителям.. 2. Анализ результатов. В этой части работы представлены некоторые результаты расчетов по формулам (6) и (7) для различных распределений вероятностей. Вычисления велись для n=1..50. Точность выполнения (8) для всех расчетов – расхождение с единицей, не превышающее 0.01. Но так или иначе здесь вряд ли требуется высокая точность, т.к. исходное распределение случайного потока событий может удовлетворять гипотезе о соответствии некоторому известному теоретическому распределению только с изрядной долей условности. Гораздо важнее получить представление о качественном поведении полученных зависимостей и возможности их приближения какой-нибудь простой функцией.. На рис. 1 – 10 нанесены зависимости от k для n =10 (тонкая линия), n =25 (линия средней толщины), n =50 (утолщенная линия). Для всех распределений, за исключением распределения Фишера, параметры были подобраны таким образом, чтобы математическое ожидание всюду было равно 1. Носителем всех распределений являются неотрицательные значения x.. Экспоненциальное распределение (рис.1).. Здесь зависимость имеет строго биномиальное распределение. Численное решение с большой точностью (максимальное отклонение не превышает 0.001), но, как уже было сказано, это совершенно не означает, что такая точность будет иметь место для других распределений.. . . Рис.1. Экспоненциальное распределение.. Распределение Вейбулла .. Свойства этого распределения существенным образом зависят от значения r. На рис.2 показана зависимость для r=2, λ=10/9. По сравнению с экспоненциальным (соответствующим r=1) максимумы возросли, подъемы и спуски стали несколько круче.. . Рис.2. Распределение Вейбулла, r=2.. Если же картина меняется. В этом случае распределение становится heavy-tailed, кроме того, в нуле оно обращается в бесконечность, и для него возникает ситуация, описанная в п.1.5. Графики изображены на рис. 3. r=0.5, λ=0.5.. Максимумы стали меньше, но самое главное – гораздо бо льшая суммарная вероятность стала приходиться на «хвосты», ибо уменьшилось количество аргументов, в которых значение функции равно нулю – кривые стали опускаться к нулю в обе стороны от своего максимума гораздо медленнее Иными словами, «хвосты» очевидным образом «утяжелились». И эта особенность выражена тем более ярче, чем меньше значение параметра r. Интегралы же, входящие в (1)-(5), сходятся в нуле , т.к. отрицательная степень, в которую возводится x, по модулю всегда меньше единицы при 0<r<1.. . . Рис.3. Распределение Вейбулла, r=0.5.. Гамма-распределение . . γ – нижняя неполная гамма-функция. Свойства этого распределения при и соотносятся между собой примерно так же, как и для распределения Вейбулла. График для r=2, θ=0.5 изображен на рис.4, для r=0.5, θ=2 – на рис. 5. Первый момент в обоих случаях равен единице.. . Рис.4. Гамма-распределение, r=2. . Рис.5. Гамма-распределение, r=0.5. Логнормальное распределение . ,. , erf ( x ) – функция ошибок [15]. Так как x здесь стои т под знаком логарифма, рост модуля показателя экспоненты не столь велик и, по крайней мере, при прием, описанный в п.1.6, применять не нужно. Данное распределение является heavy tailed. Графики (рис.6) соответствуют параметрам µ=-0.5, σ=1.. . Рис. 6. Логнормальное распределение. Распределение Рэлея .. рафики изображены на рис.7, .. . Рис. 7. Распределение Рэлея.. Распределение Накагами . ,. На рис.8 µ=2, ω≈1.131. Для (10) здесь так же можно избавиться от возведения показателя экспоненты в квадрат, как в случае распределения Рэлея.. . Рис. 8. Распределение Накагами.. Распределение Бёрра . , ,. B(x,y) – бета-функция. Распределение является heavy-tailed. При не слишком больших значениях c и r необходимости преобразовывать выражение (10) нет. Графики изображены на рис.9, c=2, r=1.5.. . Рис.9. Распределение Бёрра.. Распределение Фишера . , , . I – частичная бета-функция (верхний предел интегрирования равен не 1, а нижнему индексу при I). Поскольку первый момент для этого распределения всегда строго превышает единицу, на рис.10 изображены графики для . Соответственно, Преобразовывать (10) здесь не требуется. Входящий же в неполную бета-функцию определенный интеграл для данных параметров легко можно вычислить аналитически, что дает. , где. . Рис.10. Распределение Фишера.. На рис. 11 показана зависимость для распределений экспоненциального (тонкая линия), Вейбулла, r=0.5 (утолщенная линия), Вейбулла, r=2 (толстая линия); на рис. 12 – для распределений гамма, r=0.5 (тонкая линия), гамма, r=2 (утолщенная линия), логнормального (толстая линия); на рис.13 – для распределений Бёрра (тонкая линия), Накагами (утолщенная линия), Рэлей (толстая линия), Фишера (нижняя линия).. . Рис.11. Экспоненциальное и распределения Вейбулла.. . . Рис.12. Логнормальное и гамма-распределения. . Рис.13. Распределения Бёрра, Накагами, Рэлея, Фишера.. Из графиков видно, что зависимости для всех распределений являются линейными или, во всяком случае, при данных значениях параметров очень близки к ним. Угловой коэффициент для каждой из этих прямых можно оценить как. Это есть вероятность того, что остаточное время ожидания события не превысит 1.. . Заключение. Итак, показана практическая применимость и вычислительная реализуемость системы уравнений (1)-(5) – как на эталонном примере, так и на большом количестве произвольных распределений с различными свойствами, в том числе и для heavy-tailed распределений, а также предложена методика оценки достоверности результатов. Предложены решения проблем, возникающих во время расчетов для тех или иных исходных распределений. Результаты экспериментов показали, что распределение количества интервалов, в которых произошли события, хоть и не является в общем случае биномиальным, достаточно хорошо приближается таковым при произвольном распределении входного случайного потока, а среднее количество таких интервалов можно вычислять как линейную функцию от n . Получена формула, по которой можно оценить значение углового коэффициента этой функции. Такая оценка вполне достаточна для инженера, которому необходимо оценить количество индексов в результатах запроса к базе данных и спрогнозировать производительность обработки запросов.. В то же время ввиду значительной трудоемкости вычислений и ограниченности вычислительных ресурсов, находящихся в распоряжении автора, для каждого из рассмотренных распределений не были построены зависимости в широком диапазоне значений их параметров – для каждого из распределений расчеты проводились только для одного-двух значений. Вполне вероятна и такая ситуация, что при достаточно больших или, наоборот, достаточно малых значениях параметров и, соответственно, математического ожидания, свойство биномиальности итоговых распределений будет выражено всё менее явно.. .