

**Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
«Национальный исследовательский университет “Высшая школа
экономики”»**

**Сборник работ Первой студенческой
конференции факультета компьютерных наук
НИУ ВШЭ**

2016

Москва

Организационный комитет

Игнатов Дмитрий Игоревич – доцент, департамент анализа данных и искусственного интеллекта

Макаров Илья Андреевич – заместитель руководителя, департамент анализа данных и искусственного интеллекта

Черняк Екатерина Леонидовна – старший преподаватель, Департамент анализа данных и искусственного интеллекта

Долгалева Ирина Игоревна – стажер Тьюторского центра ФКН

Программный комитет

Миркин Борис Григорьевич – председатель ПК, Ординарный профессор НИУ ВШЭ

Дворянский Леонид Владимирович – научный сотрудник, Научно-учебная лаборатория процессно-ориентированных информационных систем (ПОИС)

Друца Алексей – Yandex, Исследования

Ильвовский Дмитрий Алексеевич – старший преподаватель, Департамент анализа данных и искусственного интеллекта

Каленкова Анна Алексеевна – старший научный сотрудник, Научно-учебная лаборатория процессно-ориентированных информационных систем (ПОИС)

Максименкова Ольга Вениаминовна – старший преподаватель, Департамент программной инженерии

Мицюк Алексей Александрович – младший научный сотрудник, Научно-учебная лаборатория процессно-ориентированных информационных систем (ПОИС)

Незнанов Алексей Андреевич – доцент, Департамент анализа данных и искусственного интеллекта

Объедков Сергей Александрович – заместитель декана по научной работе и международному сотрудничеству, Факультет компьютерных наук

Панов Александр Игоревич – научный сотрудник, Научно-учебная лаборатория процессно-ориентированных информационных систем (ПОИС)

Пузыревский Иван Витальевич – Департамент больших данных и информационного поиска, Базовая кафедра Яндекс

Шень Александр Ханиевич – кандидат физико-математических наук, старший научный сотрудник ИППИ РАН, научный сотрудник CNRS

Яворский Ростислав Эдуардович – доцент, Департамент анализа данных и искусственного интеллекта

Яковлев Константин Сергеевич – старший преподаватель, Базовая кафедра "Математические методы системного анализа" Института системного анализа РАН

Родригес Залепинос Рамон Антонио – доцент, Департамент программной инженерии

СОДЕРЖАНИЕ

1. Кудинов А.Н. «Сочетание локальных и глобальных методов при моделировании внимания»	1
2. Тюрюмина Э.Я. «Объединенная математическая модель роста первичной опухоли и вторичных метастазов рака молочной железы»	2
3. Егорова Е.Е. «Композиционные коды для канала ограниченного множественного доступа»	6
4. Сахарова Е.А. «Изучение возможности калибровки сосудистого дерева под конкретного пациента»	10
5. Моисеев Г. «Classification of e-commerce websites by product categories».	15
6. Ермакова В. «Верификация вложенных сетей Петри с помощью метода развёртки»	18
7. Кушнир И.В. «Визуализация областей интегрирования в различных системах координат»	22
8. Цымбалов Е. «Компактные разностные аппроксимации для задачи Штурма-Лиувилля»	26
9. Ким П.В. «Метод конвертации сервисных автоматов в модели на языке BPEL»	35
10. Филин Д.А. «Генерация предикторов для аттракторной модели внимания»	42
11. Шемендюк А.А. «Граничные условия, имитирующие задачу Коши для разностных уравнений в частных производных»	47
12. Шугуров И.С. «Симуляция бизнес-процессов и генерация искусственных логов событий»	50
13. Токмаков М.А. «Разработка игрового искусственного интеллекта с имитацией реакции человека на появление противника в трехмерном лабиринте на Unreal Engine 4»	55
14. Давыдова К.В. «Метод автоматического построения UML-диаграмм последовательности с задаваемым уровнем детализации на основе журналов событий»	58
15. Родин И. «Визуализация динамических графов референций»	64
16. Токмакова Л.Р. «Визуальное распознавание противника в трехмерном лабиринте на Unreal Engine 4»	71
17. Бадретдинов Т.Р. «Преобразование казуальных сетей в бесконфликтные сети потоков работ»	74
18. Кошелева Е., Петров М., Бенькович Н., Баранецкая Д., Быков Н. «Оценка объема рынка DIY в России»	78

Кудинов Антон Николаевич, 2 курс, Прикладная математика и информатика

Сочетание локальных и глобальных методов при моделировании внимания.

1 Формулировка проблемы.

В различных задачах компьютерного зрения зачастую возникает потребность в распознавании объектов, которые находятся на изображении. Однако, чтобы распознать, их нужно сначала найти. То есть, требуется отличать области, в которых присутствуют объекты, от остального изображения.

Ниже будет описан принцип работы каждого алгоритма при анализе по одному параметру (например, цвету). На практике, нужно применять его по отдельности к каждому параметру, а потом накладывать полученные результаты друг на друга.

Будут использоваться следующие понятия:

- Исходное изображение имеет размер $m \times n$ пикселей
- Карта параметра - двумерный массив $m \times n$, в i,j -ом элементе которого хранится значение параметра (M), например, интенсивность цвета (красного, синего или зеленого), яркость, насыщенность и пр.
- Карта активности внимания - двумерный массив $m \times n$, в i,j -ом элементе которого хранится число, пропорциональное тому, насколько "интересна" окрестность i,j -ого пикселя (A)

2 Алгоритм GBVS.

2.1 Построение цепи Маркова.

После получения на вход карты параметров алгоритм строит полный ориентированный граф без петель так, что вес ребра $(i,j) \rightarrow (p,q)$ равен $d((i,j)|(p,q)) * F(i-p, j-q)$, где d - логарифмическое расстояние между значениями параметра в (i,j) и (p,q) , т.е.

$$d((i,j)|(p,q)) = \left| \log\left(\frac{M(i,j)}{M(p,q)}\right) \right|,$$

$$F(a,b) = \exp\left(-\frac{a^2+b^2}{2\sigma^2}\right),$$

где σ - это некоторая константа.

Объединенная математическая модель роста первичной опухоли и вторичных метастазов рака молочной железы

Тюрюмина Элла¹

Прикладная математика и информатика, БПМИ133
eyatyuryumina_1@edu.hse.ru,

Ключевые слова: рак молочной железы, первичная опухоль, вторичные отдаленные метастазы, выживаемость, математическая модель, экспоненциальная модель роста

1 Введение

Рак молочной железы (РМЖ) на сегодняшний день является одним из самых распространенных онкологических заболеваний в мире. На РМЖ приходится 26% всех онкологических заболеваний у женщин [5]. Математическое моделирование играет очень важную роль в прогнозировании развития опухолевого процесса РМЖ. Построены различные классические математические модели, как для роста первичной опухоли, так и для роста вторичных метастазов [4]. Экспоненциальную модель роста считают классической математической моделью роста первичной опухоли РМЖ [3,4]. В 2015 году была предложена «Объединенная математическая модель роста РМЖ» (Объединенная модель) [6]. Важным свойством является то, что модель описывает рост первичной опухоли и вторичных отдаленных метастазов РМЖ, как единое целое: 1) «скрытый» период роста первичной опухоли; 2) «видимый» период роста первичной опухоли; 3) «скрытый» период роста вторичных отдаленных метастазов; 4) «видимый» период роста вторичных отдаленных метастазов. Как следствие, модель описывает известные клинические данные, обладает способностью определять различные периоды роста первичной опухоли и вторичных метастазов и предсказывать возможный прогноз прогрессирования РМЖ [6].

2 Цель

Адаптировать формулы и рассчитать все периоды роста первичной опухоли и вторичных отдаленных метастазов РМЖ по результатам, опубликованным в открытой печати [1,2].

3 Материалы и метод

Для математических расчетов использовалась Объединенная модель [6]. Для расчетов периода удвоения объема метастатической опухоли РМЖ бы-

ли адаптированы формулы из Объединенной модели:

$$PT_{\log(V)} + Nonvis_{log} + Vis_{log} = 60, \quad (1)$$

где $PT_{\log(V)}$ – число удвоений для первичной опухоли;

$Nonvis_{log}$ – число удвоений для периода «скрытого» роста вторичных отдаленных метастазов;

Vis_{log} – число удвоений для периода «видимого» роста вторичных отдаленных метастазов;

$$60 - PT_{\log(V)} = Nonvis_{log} + Vis_{log} = Survival, \quad (2)$$

$$TVDT_{non} = TVDT_{vis} = \frac{Nonvis_{days} + Vis_{days}}{Nonvis_{log} + Vis_{log}}, \quad (3)$$

где $TVDT_{non}$ – период удвоения объема опухоли (в днях) для периода «скрытого» роста вторичных отдаленных метастазов;

$TVDT_{vis}$ – период удвоения объема опухоли (в днях) для периода «видимого» роста вторичных отдаленных метастазов;

$Nonvis_{days}$ – период «скрытого» роста вторичных отдаленных метастазов в днях;

Vis_{days} – период «видимого» роста вторичных отдаленных метастазов в днях.

На основании адаптированных формул были рассчитаны: период удвоения объема метастатической опухоли, число удвоений «скрытого» роста, число удвоений «видимого» роста метастатической опухоли РМЖ в зависимости от размеров первичной опухоли. Средний диаметр первичной опухоли на момент операции для каждой стадии (pT1, pT2, pT3 и pT4) был взят из статьи [2]. Период «скрытого» роста (time from diagnosis to mts) и период «видимого» роста (time from mts to death) метастатической опухоли РМЖ в зависимости от размеров первичной опухоли для каждой стадии (pT1, pT2, pT3 и pT4) были получены из таблиц [1].

4 Результаты

Период удвоения объема метастатической опухоли РМЖ в зависимости от размеров первичной опухоли (критерий T) варьирует от 62,9 до 68,2 дней для pT1, pT2 и pT3. Период удвоения объема метастатической опухоли РМЖ в зависимости от размеров первичной опухоли (критерий T) варьирует от 45,7 до 48,1 дней для pT4, что свидетельствует об агрессивности роста опухоли РМЖ у больных с pT4. Число удвоений для периода «скрытого» роста вторичных отдаленных метастазов зависит от размеров первичной опухоли (критерий T) для pT1, pT2 и pT3. Число удвоений для периода «видимого» роста вторичных отдаленных метастазов не зависит от размеров первичной опухоли (критерий T) для pT1, pT2 и pT3. Число удвоений для метастатической опухоли на момент операции зависит от размеров первичной опухоли (критерий T). Полученные данные позволяют

определить примерный период появления первой метастатической клетки в зависимости от размеров первичной опухоли (критерий T) для pT1, pT2 и pT3. Полученные данные позволяют определить примерный период «проявления» первой метастатической опухоли в зависимости от размеров первичной опухоли (критерий T) для pT1, pT2 и pT3. Размер метастатической опухоли на момент диагностики составил 8-9 мм в зависимости от стадии pT1, pT2 и pT3.

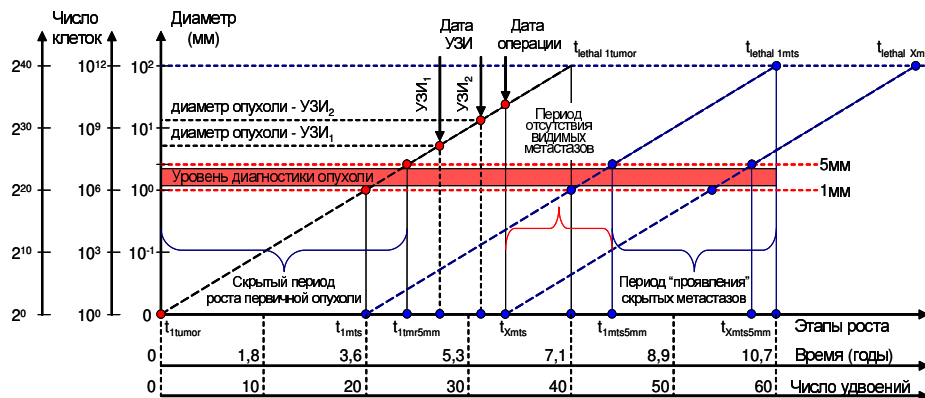


Рис. 1. Объединенная математическая модель роста первичной опухоли и вторичных отдаленных метастазов РМЖ [6]. По оси «Y» – диаметр опухоли (мм) в сопоставлении с количеством опухолевых клеток; по оси «X» – этапы и время роста (годы) в сопоставлении с числом удвоения объема опухоли. УЗИ1 – дата и размеры первого ультразвукового исследования первичной опухоли; УЗИ2 – дата и размеры второго ультразвукового исследования первичной опухоли. Период удвоения объема опухоли = 65 дней.

«Полная естественная история» роста первичной опухоли и вторичных отдаленных метастазов РМЖ от момента появления первой стволовой опухолевой клетки до даты смерти варьирует от 10,3 до 11,2 лет для pT1, pT2 и pT3 (рис. 1). «Полная естественная история» роста первичной опухоли и вторичных отдаленных метастазов РМЖ от момента появления первой стволовой опухолевой клетки до даты смерти варьирует от 7,5 до 7,9 лет для pT4, что свидетельствует об агрессивности роста опухоли РМЖ у больных с pT4.

5 Заключение

Продолжительность жизни больных РМЖ (выживаемость) включает период «скрытого» роста вторичных отдаленных метастазов (time from diagnosis to mts) и период «видимого» роста вторичных отдаленных метастазов (time

from mts to death) [6]. Применяя Объединенную модель, впервые была определена продолжительность выживаемости для pT1, pT2 и pT3 и она варьирует от 3,8 до 5,3 лет. Продолжительность выживаемости для pT4 составила от 2,9 до 3,0 года. Применяя Объединенную модель, впервые была установлена продолжительность периода «скрытого» роста вторичных отдаленных метастазов для pT1, pT2 и pT3 и она варьирует от 1,9 до 3,2 года. Продолжительность периода «скрытого» роста вторичных отдаленных метастазов для pT4 составила от 1,1 до 1,2 года. Впервые выявлено, что период «видимого» роста вторичных отдаленных метастазов РМЖ варьирует от 1,9 до 2,1 года для pT1, pT2 и pT3. Данные периода «видимого» роста вторичных отдаленных метастазов РМЖ для pT4 практически не отличаются от данных для pT1, pT2 и pT3 и составили 1,8-1,9 года.

Список литературы

1. *Engel, J., Eckel, R., Kerr, J., et al.* The process of metastatisation for breast cancer // *Eur J Cancer*. 2003; Vol. 39(12). P. 1794-1806.
2. *Engel, J., Emeny, R., Holzel, D.* Positive lymph nodes do not metastasize // *Cancer Metastasis Rev.* 2012; 31(1-2). P. 235-246.
3. *Rodriguez-Brenes, I., Komarova, N., Wodarz, D.* Tumor growth dynamics: insights into evolutionary processes // *Trends Ecol Evol.* 2013; Vol. 28(10). P. 597-604.
4. *Schwartz, M.* A biomathematical approach to clinical tumour growth // *Cancer*. 1961; Vol. 14. P. 1272-1294.
5. *Shah, R., Rosso, K., Nathanson, S.* Pathogenesis, prevention, diagnosis and treatment of breast cancer // *World J Clin Oncol.* 2014; Vol. 5(3). P. 283-298.
6. *Тюрюмина, Э. Я., Незнанов, А. А.* Объединенная математическая модель роста рака молочной железы / В сб. «39-я междисциплинарная школа-конференция «Информационные технологии и системы», Труды 39-ой междисциплинарной школы-конференции «Информационные технологии и системы». С. 38-51. — Сочи, Издательство Springer, 2015.

Композиционные коды для канала ограниченного множественного доступа

Егорова Е.Е.¹

Образовательная программа Науки о данных, специализация Теория моделирования сложных систем, группа м14 НоД ТМСС
eeegorova@edu.hse.ru

Ключевые слова: теория кодирования, аддитивный канал множественного доступа, сигнатурные коды

1 Введение

Известная задача о поиске фальшивой монеты [3] описывается следующим образом. Дано множество из M монет, s или меньше из которых фальшивые, т.е. отличаются по весу от настоящих монет. Задача заключается в нахождении минимального числа взвешиваний, необходимых для определения всего множества фальшивых монет. Одной из известных интерпретаций данной задачи является модель суммирующего канала ограниченного множественного доступа [1], [2]. Данная модель состоит в следующем: имеется M пользователей, каждому из пользователей соответствует двоичный вектор длины n , множество всех таких векторов будем называть кодом. В каждый отведенный для передачи момент времени не более s из M пользователей активны, т.е. принимают участие в передаче информации. Каждый активный пользователь передает по каналу свой двоичный вектор. На выходе канала по полученному суммарному вектору (сумма берется как обычная сумма векторов над действительными числами) декодер должен однозначно определить множество всех пользователей, участвующих в передаче. Коды, позволяющие это сделать, называются s -сигнатурными. Проводя аналогию с задачей о фальшивых монетах, нетрудно заметить, что число пользователей соответствует числу монет, а длина двоичных векторов соответствует числу взвешиваний. Задача заключается в следующем: при данном количестве пользователей M минимизировать длину векторов n .

В данной работе в терминах модели ограниченного множественного доступа формулируется обобщение данной задачи, а также вводится понятие s -композиционного кода. Основным результатом данной работы является вычисление верхней и нижней границ скорости такого кода.

2 Постановка задачи

Рассмотрим канал с s входами и одним выходом. На каждый вход канала поступает слово длины n над алфавитом $Q = \{0, 1, \dots, q-1\}$ мощности q . Если на входы канала поступили q -ичные слова $\{x_i = (x_{i1}, \dots, x_{in}) | i = 1, \dots, s\}$,

то на выходе канала появляется слово-композиция длины n

$$(\text{comp}(x_{11}, \dots, x_{s1}), \dots, \text{comp}(x_{1n}, \dots, x_{sn})),$$

в котором j -ая координата $j = 1, \dots, n$ равна композиции j -х координат слов $x_i, i = 1, \dots, s$. Напомним, что композицией q -ичного слова $(a_1, \dots, a_s)$ называется набор неотрицательных целых чисел $\{w_0, \dots, w_{q-1}\}$, где w_k — число вхождений символа k в слово $(a_1, \dots, a_s)$. Ясно, что $w_0 + w_1 + \dots + w_{q-1} = s$ и число различных композиций при заданных q и s равно $\binom{s+q-1}{q-1}$. Если всего имеется M пользователей канала, но при этом передачу по каналу могут одновременно передавать не более, чем s из них (что соответствует модели ограниченного доступа), то разумно потребовать, чтобы по выходу канала можно было правильно восстановить всех пользователей, участвующих в передаче. Впервые такой канал был представлен в работе [6] под названием B -channel, однако рассматривалось другое использование такого канала.

2.1 s -композиционный код

Рассмотрим множество $C = \{c_1, \dots, c_M\}$ векторов длины n над конечным алфавитом $Q = \{0, 1, \dots, q-1\}$, причем $|C| = M$ и $|Q| = q$. Далее мы не будем различать такое множество и матрицу, столбцами которой являются элементы C . Пусть $U \subseteq C$ — некоторая совокупность векторов из C , причем $1 \leq |U| \leq M$. Рассмотрим функцию S (далее будем называть её композицией), которая ставит в соответствие подмножеству $U \subseteq C$ матрицу W размера $n \times q$, в которой элемент w_{ij} равняется количеству элементов $(j-1) \in Q$ встретившихся в i -ой координате векторов из множества U .

Определение 1. Пусть s — натуральное число, причем $s \geq 2$ и $s \ll M$. Будем говорить, что множество C является s -композиционным кодом, если композиции любых s или меньше различных векторов (кодových слов) из C различны, т.е. $S(U) \neq S(V)$ для любых $U, V \subseteq C$ таких, что $U \neq V$ и $|U|, |V| \leq s$.

Примечание 1. При $q = 2$, т.е. в случае двоичного алфавита, s -композиционный код является s -сигнатурным.

Основная наша задача заключается в оценке скорости оптимального s -композиционного кода C , т.е. величины $R(C) = \frac{\log_q M}{n}$ при $n \rightarrow \infty$.

3 Результаты

В данной работе доказывается, что $R(C)$ лежит в следующих границах:

$$\frac{q-1}{4} \frac{\log s}{s} \lesssim R \lesssim \frac{q-1}{2} \frac{\log s}{s}. \quad (1)$$

Для получения нижней границы используется метод случайного кодирования. Основная идея данного метода заключается в вычислении вероятности того, что случайная матрица, т.е. матрица элементы которой выбираются равновероятно из алфавита $Q = \{0, 1, \dots, q - 1\}$, не будет являться s -композиционным кодом мощности M . Доказывается, что при такой скорости эта вероятность строго меньше единицы. Из этого следует, что вероятность, что случайно выбранный код окажется "хорошим", т.е. s -композиционным кодом, строго больше нуля, т.е. s -композиционные коды при заданных параметрах существуют.

Для доказательства верхней границы используется энтропийный метод. Для этого рассматривается произвольный q -ичный s -композиционный код C длины n и мощности M . За U обозначим дискретную случайную величину распределенную равномерно на множестве s -элементных подмножеств множества C :

$$\Pr(U = A) = \begin{cases} \binom{M}{s}^{-1} & , \text{ если } A \subseteq C, |A| = s \\ 0 & , \text{ иначе.} \end{cases}$$

Основная идея доказательства заключается в оценке информационной энтропии случайной величины $S(U)$: объединив границы сверху и снизу мы получим верхнюю границу на скорость такого s -композиционного кода.

Заметим, также, что при $q = 2$ из (1) получается следующая граница:

$$\frac{\log s}{4s} \lesssim R \lesssim \frac{\log s}{2s},$$

что в действительности соответствует границам на скорость сигнатурного кода, полученных в работах [1], [4], [5].

Благодарности. Благодарю Бассальго Л.А. за знакомство с этой задачей и за внимательную проверку всех шагов её решения. Работа выполнена при частичной финансовой поддержке Фонда Саймонса.

Список литературы

1. Györfi L., Gyori S., Laczay B. Lectures on Multiple Access Channels. // URL: http://www.szit.bme.hu/~gyori/AF0SR_05/book.pdf
2. Quang A. Some constructions of signature sequences for T-user multiple-access adder channels with correlation type receiver //PROB. CONTROL INFO. THEORY. – 1986. – Т. 15. – №. 3. – С. 253-260.
3. P.Erdos and A.Renyi, "On two problems of information theory", Publ.Math. Inst. Hung. Acad.Sci, V.8, pp. 241-254, 1963.

4. А. Г. Дьячков, В. В. Рыков, "Об одной модели кодирования для суммирующего канала с множественным доступом", Пробл. передачи информ., 17:2 (1981), 26–38
5. Poltyrev G. S. Improved upper bound on the probability of decoding error for codes of complex structure // Problemy Peredachi Informatsii. – 1987. – Т. 23. – №. 4. – С. 5-18.
6. Chang, Shih-Chun, and Jack K. Wolf. "On the T-user M-frequency noiseless multiple-access channel with and without intensity information." Information Theory, IEEE Transactions on 27.1 (1981): 41-48.

Исследование возможностей калибровки сосудистого дерева под конкретного пациента

Сахарова Екатерина Алексеевна¹

¹ Факультет компьютерных наук отделение прикладной математики и
информатики, 4 курс
sakharova.hse@gmail.com

Ключевые слова: гемодинамика, калибровка, уравнение гидродинамики, сосудистое дерево.

Аннотация

Цель данной работы – исследование возможностей адаптации модели кровообращения под конкретного пациента. Для этого рассматривается эксперимент, моделирующий течение жидкости в главных артериях. Параметры данного эксперимента, а также измерения потоков жидкости и давлений известны. Воспроизведение данного эксперимента численной моделью позволило оценить значимость некоторых входных модельных параметров. Уменьшение количества входных параметров упрощает адаптацию модели под конкретного пациента.

Введение

Математическое моделирование и численные расчеты течения крови в теле человека предоставляют информацию, которая может использоваться для правильного диагностирования и лечения. Циркуляция крови описывается уравнениями гидродинамики. Актуальной проблемой является возможность расчета гемодинамики конкретного пациента. Для этого необходимо индивидуально формировать входные данные для модели. В данной работе исследована значимость различных модельных параметров для воспроизведения реального эксперимента.

Модель

В модели кровообращения сосудистая система представляется в виде графа: ребра соответствуют сосудам, точки стыковки ребер – ветвлению сосудов. Кровь считается вязкой несжимаемой жидкостью, протекающей по сети трубок с эластичными стенками. Модель течения крови основывается на законах сохранения массы и импульса [1, 2, 3]:

$$\frac{\partial S}{\partial t} + \frac{\partial(S\bar{u})}{\partial x} = \phi(t, x, S, \bar{u}), \quad (1)$$

$$\frac{\partial \bar{u}}{\partial t} + \frac{\partial(\frac{\bar{u}^2}{2} + \frac{\bar{p}}{\rho})}{\partial x} = \psi(t, x, S, \bar{u}), \quad (2)$$

где

S – площадь поперечного сечения сосуда;

$\bar{u}$ – скорость жидкости, осредненная по сечению;

$\bar{p}$ – трансмуральное давление;

t – координата по времени, $t \in [0, T]$, T – общее время расчета;

x – координата вдоль сосуда, $x \in [0, l]$, l – длина сосуда;

ρ – плотность жидкости;

ϕ, ψ – специальные функции.

Система (1)(2) замыкается уравнением состояния, которое характеризует эластичные свойства стенок сосудов [2]. В литературе известны несколько подходов к его построению. Например,

1. вывод из уравнения равновесия в предположении линейной упругости материала стенки [1]:

$$\bar{p} = \frac{\beta}{S_0}(\sqrt{S} - \sqrt{S_0}),$$

где

$$\beta = \frac{\sqrt{\pi} h E}{1 - \sigma^2};$$

h – средняя толщина стенки;

E – модуль Юнга;

$\sigma = \frac{1}{2}$ – коэффициент Пуассона;

S_0 – площадь поперечного сечения при $\bar{p} = 0$.

2. построение функции эмпирическим путем:

$$\begin{aligned} \bar{p} &= \rho c_0^2 f(S) [2]; \\ f(S) &= \begin{cases} \exp(SS_0^{-1} - 1) - 1, & S \geq S_0 \\ \ln(SS_0^{-1}), & S \leq S_0 \end{cases} \end{aligned} \quad (3)$$

где c_0 – скорость распространения малых возмущений. В этом случае уравнение состояния учитывает нелинейность материала стенки.

Система (1)(2) имеет гиперболический тип, следовательно, на концах сосуда характеристические кривые накладывают особые условия, называемые условиями совместности. Для корректной постановки задачи на каждом конце сосуда необходимы дополнительные условия. Значит, в точках стыковки ребер надо задать систему уравнений, состоящую из закона сохранения массы и одного из условий (а) – (д):

а) условие Пуазейля

$$P_i(S_i) - P_* = R_i u_i S_i,$$

где $i=1..n$, n – количество сосудов, P_i – давление в сосуде № i , R_i – сопротивление в сосуде № i , P_* – давление в узле

б) непрерывность полного давления

$$(P_i + \frac{u_i^2}{2}) - (P_{i+1} + \frac{u_{i+1}^2}{2}) = 0, i = 1..n - 1$$

с) непрерывность давления

$$P_i - P_{i+1} = 0, i = 1..n - 1$$

d) для бифуркаций [3]

$$\begin{cases} P_{t,1} = P_{t,2} + \text{sign}(u_2)f_2(u_2, \alpha_2) \\ P_{t,1} = P_{t,3} + \text{sign}(u_3)f_3(u_3, \alpha_3) \end{cases} \quad (4)$$

Индекс «1» соответствует материнской ветви, а «2» и «3» - дочерним.

$$f_i(u_i, \alpha_i) = 2u_i^2 \sqrt{2(1 - \cos(\alpha_i))},$$

где $i=2,3$; α_i - угол наклона i ветви относительно материнской.

Для расчетов в данной работе использовалась численная модель Симакова-Холодова, разработанная в МФТИ [2].

Адаптация модели под конкретного пациента

В зависимости от используемых граничных условий и уравнения состояния, данная задача может иметь ряд неизвестных параметров: толщина стенки сосуда, углы ветвления сосудов, сопротивления, скорость распространения малых возмущений. Такие параметры индивидуальны для каждого пациента, их трудно измерить, поэтому требуется понять, можно ли избежать их использования без потери качества численного решения.

Индивидуальными являются также геометрия и расположение сосудов. С помощью сегментации снимков магнитно-резонансной томографии (МРТ) конкретного пациента получают трехмерные области сосудистого русла. Путем выделения центральных линий трехмерная сосудистая сеть сводится к одномерной, где для любого сосуда известны длина, площадь поперечного сечения, а для бифуркаций - углы. Граф может состоять как только из основных артерий, так и учитывать мелкие сосуды. Чаще всего для расчетов используют графы от 30 до 60 ребер [1, 4].

Численные эксперименты

В работе [1] был представлен эксперимент: из силиконовых трубок построена модель главных артерий человека. Данная конструкция соединена со специальным прибором, который производит симуляцию периодического выброса сердца (Рис.1). На каждом сосуде измерены давления и скорости

жидкости, схожей по плотности с кровью. Все параметры трубок и материала; результаты измерений приведены в статье [1]. Там же были представлены численные расчеты на графе (Рис.2), соответствующем модели из силиконовых трубок. Результаты имеют малые отклонения от данных, полученных экспериментальным путем.

В данной работе мы повторили расчеты на том же графе, используя параметры и условия, описанные в [1]. Численные результаты соотносятся как с экспериментальными данными, так и с расчетами из [1, 6].

Для оценки значимости модельных параметров и улучшения сходства численного решения с экспериментом, проведены расчеты с различными уравнениями состояния и граничными условиями в узлах. На основании результатов были сделаны следующие выводы:

1. сосуды, радиус которых линейно изменяется по длине, могут считаться трубками с постоянным радиусом ($S_0 = \frac{\int_0^l S(x)dx}{l}$) При этом качество численного решения не ухудшается по сравнению с результатами эксперимента;
2. результаты численных расчетов при использовании условий (b) – (d) отличаются незначительно. Причем эти условия не имеют неизвестных параметров в отличие от условия (a);
3. учет линейных свойств материала в уравнении состояния дает более точные результаты численного решения по сравнению с экспериментальными измерениями.

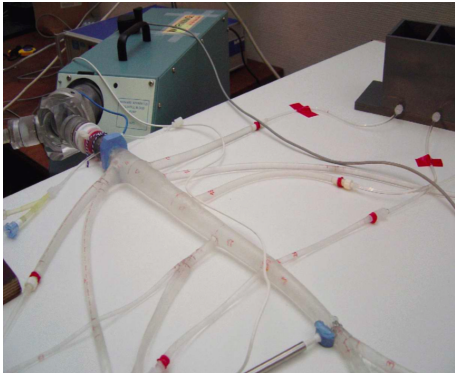


Рис.1. Экспериментальная модель главных артерий человека

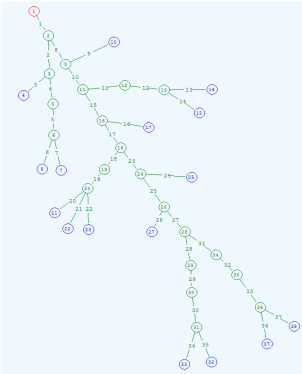


Рис.2. Граф, построенный по экспериментальной модели

Заключение

Уменьшение количества входных параметров упрощает адаптацию модели кровообращения конкретного пациента. В данной работе исследовано влияние некоторых параметров на гемодинамику и показано, какие из них можно исключить. Полученные численные результаты соответствуют эксперименту. Описанные методы настройки графа могут применяться для расчета кровотока реального пациента.

Список литературы

1. *Jordi A., Koen M., Joaquim P.* Pulse wave propagation in a model human arterial network: assessment of 1-D numerical simulation against in-vitro measurements — J Biomech, 2007. 40(15):3476-86.
2. *Vassilevski Yu., Simakov S., Salamatova V., Ivanov Y., Dobroserdova T.* Blood Flow Simulation in Atherosclerotic Vascular Network Using Fiber-Spring Representation of Diseased Wall — Math. Model. Nat. Phenom. Vol. 6, No. 5, 2011, pp. 333-349
3. *Lamponi, D.* One Dimensional and Multiscale Models for Blood Flow Circulation — PhD thesis, EPFL, Lausanne, Thesis n.3006, 2004
4. *Matthys K, Alastruey J, Peiró J, Khir A, Segers P, Verdonck P, Parker K, Sherwin S.* Pulse wave propagation in a model human arterial network: assessment of 1-D numerical simulations against in vitro measurements — Journal of Biomechanics 2007; 40:3476–3486.
5. *Etienne B., Perumal N., Pablo J.* A benchmark study of numerical schemes for one-dimensional arterial blood flow modelling — Int. J. Numer. Meth. Biomed. Engng.; 2015, e02732

Classification of e-commerce websites by product categories

George Moiseev

Higher School of Economics, Moscow, Russia
Software Engineering Department
Year 4
gvmoiseev@edu.hse.ru

One of the most common problems nowadays is a high amount of mostly unorganized information on the web. With the exponential growth of data around the web the arrangement of the information becomes an important task for assisting users and companies in storing and retrieving the information. One example of such task is an automated e-commerce websites categorization problem which also includes the issue of retrieving such sites from the web. This problem comes from a high need of clustered or categorized e-commerce websites for market researchers who need to take into account different types of statistics in the e-commerce sphere, such as [1], comparison of shopping engines like Google Shopping or Yandex Market [2], information retrieving systems [3] followed by other types of services. Normally, categorization by sold product type attracts the most interest.

At first sight, the raised problem is a part of a wider text categorization issue or a web page classification task and it can be solved by direct borrowing of existing algorithms from machine learning literature dedicated to these issues [4-7]. Nevertheless, the solution is far from being so straightforward. Web pages are highly structured and filled with noisy content such as javascript code, advertisements and copyrights. Without taking these factors into account, they would have negative impact on performance of pure text classification algorithm. It has been proved that exploitation of the structure of a web page (HTML tags, hyperlinks) enhances the quality of classification [12].

Although most web page classification [6-8] algorithms apply noise reducing techniques and use structure of web pages to improve classification, there are still issues to discuss and ways to improve.

Firstly, classifying a website raises some ambiguous questions: which webpages of the website should be downloaded and processed, should hyperlinks from the main page be used and how, should one consider content from all pages of the website in feature selection process. Our method consists in downloading the main page of a website and retrieving useful hyperlinks by eliminating links that satisfy one or more conditions listed below:

1. the link refers to a different website;
2. the hyperlink anchor text contains terms frequently used in anchor texts of web pages from different categories. The list of these terms is a union of sets of the

most frequent anchor texts from each category. Some examples of such terms are: «доставка», «контакты», «помощь» and «корзина».

From the content of retrieved web pages we extract only meta tags and title (which usually gives a good summary of the page) and combine them with information from the main page. This approach eliminates the most possible noise from other pages by extracting only summarized information from meta tags and title. However, using entire pages may have its advantage because it allows considering more information about the web site. Thus both variants are tested and compared with classifying only by the main page content.

The second non trivial question is related to the use of a web page structure in classification process. Our idea is to weight terms against the nearest tag they are nested in and to calculate the weight of tags inversely proportional to their frequency (i.e. the more frequent the tag is, the less valuable enclosed terms are).

The term weighting formula for the i th term in the k th web site is derived from TF-IDF [15] as follows:

$$W_{ik} = \frac{tf_{ik} \log \frac{N}{n_i}}{\sqrt{\sum_{j=1}^N (tf_{ij} \log \frac{N}{n_j})^2}} \quad (1)$$

where n_i is the number of websites where the i th term appears, N – total number of web sites in the sample and tf_{ik} is computed as:

$$tf_{ik} = \sum_t^T w(t) f(i, k, t) \quad (2)$$

where T is the set of all tags of k th web site, $f(i, k, t)$ is the frequency of the i th term in tag t from web site k and $w(t)$ is calculated as follows:

$$w(t) = \frac{1}{\sum_x^T [x=t]} \quad (3)$$

Another significant point to consider is a language-based approach. Because of particularly high interest in research on the Russian online market we mostly focus on classifying Russian websites. Most researches on this topic study only web pages in English or try to develop language-independent method [7]. Concentrating on a single language area of the web allows us to use some language specific features that enhance the quality of classification.

Since the focus of this paper is on the preprocessing and feature engineering stages, we use one of the most popular machine learning algorithm for classification – Support Vector Machine (SVM) [13].

In order to test the effectiveness of proposed methods, several experiments were conducted. The dataset was received from datainsight.ru and completely consists of websites in Russian.

As a result, the experiments show that using the data from hyperlinks retrieved with our method increases the accuracy of classification. The experiments also

revealed that it is preferably to exploit only meta data and title from retrieved pages when diversity of data is high enough. Conversely, when the type of classifying data is more or less limited exploiting entire pages may improve the performance. Also, experiments show that proposed tag weighting method increases F-score on 5%.

References

1. Grandon, E., Pearson, J.: Electronic commerce adoption: an empirical study of small and medium US businesses. *Information & Management*. 42, 197-216 (2004).
2. Yuan, S.: A personalized and integrative comparison-shopping engine and its applications. *Decision Support Systems*. 34, 139-156 (2003).
3. Wai Lam, Ruiz, M., Srinivasan, P.: Automatic text categorization and its application to text retrieval. *IEEE Trans. Knowl. Data Eng.* 11, 865-879 (1999).
4. Sebastiani, F.: Machine learning in automated text categorization. *CSUR*. 34, 1-47 (2002).
5. Apte C., Damerau, F., Weiss, S.: Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*. 12, 233-251 (1994).
6. Onan, A.: Classifier and feature set ensembles for web page classification. *Journal of Information Science*. (2015).
7. Qi, X., Davison, B.: Web page classification. *CSUR*. 41, 1-31 (2009).
8. Zengmin, G., Jianxia, D.: Research on web page classification-based core characteristics and web structure. *International Journal of Wireless and Mobile Computing*. 7, 253 (2014).
9. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. *ACM SIGMOD Record*. 27, 307-318 (1998).
10. Ghani, R., Slattery, S., Yang, Y.: Hypertext categorization using hyperlink patterns and meta data. *ICML 01: Proceedings of the Eighteenth International Conference on Machine Learning*. 178-185 (2001).
11. Oh, H., Myaeng, S., Lee, M.: A practical hypertext categorization method using links and incrementally available class information. *SIGIR 00: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 264-271 (2000).
12. Webpage Classification based on Compound of Using HTML Features & URL Features and Features of Sibling Pages. *International Journal of Advancements in Computing Technology*. 2, 36-46 (2010).
13. Vapnik, V., Cortez, C.: Support vector networks. *Machine Learning*. (1995).
14. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. *10th European Conference on Machine Learning*. pp. 137-142. Springer Verlag, Heidelberg (1998).
15. Aizawa, A.: An information-theoretic perspective of tf-idf measures. *Information Processing & Management*. 39, 45-65 (2003).
16. Van Rijsbergen, C.: *Information retrieval*. Butterworths, London (1979).
17. Jackson, P., Moulinier, I.: *Natural language processing for online applications*. John Benjamins Pub., Amsterdam (2002).

Верификация вложенных сетей Петри с помощью метода развёртки

Ермакова Вера

Образовательная программа «Системная и программная инженерия»

2 курс магистратуры

1 Вступительная часть

Мультиагентные системы подробно изучаются в течение последних десятилетий и могут рассматриваться как одно из самых передовых исследований в области компьютерных наук на сегодняшний день. Они используются в различных практических областях и сферах, таких как: искусственный интеллект, облачные сервисы, сетевые системы, системы дополненной реальности с интерактивными объектами окружающей среды, сбор информации.

Тем не менее, мультиагентные системы являются очень сложными из-за их распределенной структуры. Они состоят из взаимодействующих агентов, имеющих общую среду и автономное поведение. При разработке такой системы очень важно проверить, соответствует ли эта система заданной спецификации. Именно поэтому мы создаем для системы модель: чтобы проверить, работает ли модель должным образом и только после этого мы реализуем её. Для того, чтобы выполнить эту проверку, используется подход верификации.

Одним из формализмов, успешно представляющих распределенное поведение систем, являются сети Петри. Тем не менее, из-за плоской структуры классических сетей Петри, они не слишком хороши для моделирования сложных мультиагентных систем. Для таких систем существует специальное расширение сетей Петри, называемое вложенными сетями Петри [5]. Вложенные сети Петри, естественным образом представляют собой структуру мультиагентных систем, так как фишки в системной сети и есть сети Петри, и могут иметь свое собственное поведение.

Для проверки свойств модели используется один из наиболее распространенных способов верификации - проверка модели. Основная идея верификации модели заключается в создании графа достижимости и проверке свойств на этом графе. Тем не менее, ключевой проблемой для верификации систем с высокой степенью параллелизма при использовании подхода проверки модели является большое количество интерливингов параллельных процессов (возможных последовательностей событий в системе). Это приводит к так называемой

проблеме взрывообразного числа состояний. Это означает, что, если мы добавим один компонент к системе, число состояний вырастет значительно.

Для решения этой проблемы был введен подход развертки [3, 6]. В работе [2] показано применение развертки для вложенных сетей Петри, а также представлен процесс ветвления консервативных вложенных сетей Петри покомпонентным образом. Было доказано, что развертки вложенных сетей Петри отвечают фундаментальному свойству разверток, и, таким образом, могут быть использованы для верификации консервативных вложенных сетей Петри также, как классические методы разверток. Здесь рассматривается особый подкласс вложенных сетей Петри - консервативные безопасные вложенные сети Петри. Это означает, что фишки сети, представляющие агенты, не могут уничтожаться или создаваться, но могут изменить расположение в системе сети и свои внутренние состояния. Таким образом, число агентов постоянно и каждый агент представлен в единственном экземпляре в системе.

Тем не менее, для этого конкретного класса вложенных сетей Петри можно построить эквивалентные классические сети Петри. Очевидный вопрос состоит в следующем: насколько подход развертки, предложенный в работе [2] лучше с точки зрения сложности верификации.

2 Актуальность проблемы и проделанная работа

Проблема трансляции консервативных безопасных вложенных сетей Петри в классические сети Петри ранее не рассматривались. Однако, так как консервативные безопасные вложенные сети Петри ограничены, а число достижимых состояний для них конечно, то ясно, что существует поведенчески эквивалентный класс сетей Петри.

Так как безопасные консервативные вложенные сети Петри имеют конечное число состояний, будет очевидно предположить, что они могут быть переведены в классические сети Петри.

Для того, чтобы сделать правильную трансляцию мы должны установить ряд требований к ней. Главной целью построения модели является возможность провести моделирование. Моделирование предполагает поведенческую эквивалентность: способность повторить все возможные ходы одной модели другой моделью. Поведенческая эквивалентность гарантируется путем установления сильной бисимуляционной эквивалентности состояний двух моделей.

Второе требование касается построения графа достижимости. Это значит, что нам нужно точное соответствие между состояниями нашей модели.

Если эти два требования выполнены, мы можем построить алгоритм трансляции, который позволяет нам использовать целевую модель, имеющую те же поведенческие свойства, что у оригинала для верификации и анализа.

3 Полученные результаты

Мы хотели проверить, позволяет ли применение развертки, предложенной в работе [2] уменьшить сложность верификации вложенных сетей Петри по сравнению с трансляцией вложенных сетей Петри в классические сети Петри.

Так как мы хотели доказать наши ожидания эмпирически, мы должны были транслировать и проверить большое количество вложенных сетей Петри. Был разработан алгоритм трансляции вложенных сетей Петри в сети Петри, а также была доказана поведенческая эквивалентность исходной и целевой сети. Кроме того, мы решили разработать программу, которая поможет выяснить, насколько более выгодным является использование разверток непосредственно для вложенных сетей Петри.

Мы построили программу с тремя основными функциями:

- Трансляция вложенных сетей Петри в классические сети Петри.
- Создание развёрток для сетей Петри.
- Создание разверток для консервативных безопасных вложенных сетей Петри.

Процесс трансляции большой вложенной сети Петри в сеть Петри затруднён из-за значительного роста числа переходов и состояний при различных типах элементарных сетей.

Мы провели эксперименты и сравнили результаты, полученные на различных сетях, и эмпирически доказали, что прямой метод развертывания намного эффективнее с точки зрения времени от 1.5 до 2 раз.

Был введён алгоритм трансляции из вложенных сетей Петри в классические сети Петри, который может быть применен для верификации мультиагентных систем.

Для дальнейшей работы мы планируем расширить алгоритм перевода для многоуровневых вложенных сетей Петри. Кроме того, мы планируем увеличить сложность сетей и число экспериментов.

Список использованной литературы

1. Dworzański L. W., Lomazova I. A. On compositionality of boundedness and liveness for nested Petri nets //Fundamenta Informaticae. – 2012. – Т. 120. – №. 3-4. – С. 275-293.

2. Frumin, D. Branching Processes of Conservative Nested Petri Nets / D. Frumin, I. Lomazova // Fundamenta Informaticae. – 2014. – Т. 133. – №. 2-3. – С. 197-209.
3. McMillan K. L. Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits //Computer Aided Verification. – Springer Berlin Heidelberg, 1992. – С. 164-177.
4. Khomenko, V. Canonical prefixes of Petri net unfoldings / V. Khomenko., M. Koutny, W. Vogler // Acta Informatica. – 2003. – Т. 40. – №. 2. – С. 95-118.
5. Lomazova, I. A. Nested Petri Nets - a Formalism for Specification and Verification of Multi-Agent Distributed Systems // Fundamenta informaticae. – 2000. – Т. 43. – №. 1-4. – С. 195-214.
6. Nielsen, M. Petri nets, event structures and domains / M. Nielsen.,G. Plotkin, G. Winskel // Theoretical Computer Science. – 1981. – Т. 13. – №. 1. – С. 85-108.
7. Дворянский, Л. В., Ломазова, И. А. Имитационное моделирование и верификация вложенных сетей Петри с использованием CPNTools // Вестник Воронежского государственного технического университета. – 2011. – Т. 7. – №. 11.2. – С. 62-64.
8. Ломазова И.А. “Вложенные сети Петри: моделирование и анализ распределенных систем с объектной структурой”. Научный мир, Москва, 2003 —208 с.

Визуализация областей интегрирования в различных системах координат

Кушнир Ирина Вячеславовна

Прикладная математика и информатика, 122 группа
ivkushnir94@gmail.com

Ключевые слова: интегральное исчисление, визуализация области интегрирования, VisualMath.ru

1 Аннотация

В настоящее время набирает популярность методика смешанного обучения (blended learning), основанная на синтезе классической модели преподавания и использовании современных компьютерных технологий. Во многих математических разделах при преподавании абстрактных понятий возникает потребность демонстрации визуальных образов, но обычно для таких целей преподаватель использует либо доску с мелом, либо книжные иллюстрации, либо популярные математические пакеты, такие как Matlab, Mathcad и пр.

Разрабатываемый нашей командой инструмент VisualMath.ru базируется на концепции смешанного обучения и разрабатывается с целью развития у студентов геометрической интуиции и пространственного воображения при изучении объектов высшей математики. VisualMath.ru представляет собой JavaScript онлайн-платформу, доступную из браузера, с комбинируемыми модулями визуальных примеров, используя которую, преподаватель получает возможность дополнить классический процесс обучения демонстрацией интерактивных динамических моделей.

Данная работа посвящена моделированию областей, полученных при пересечении алгебраических поверхностей, а также заданных неявными функциями. Моделирование осуществляется с помощью специальной JavaScript библиотеки Grafar, и основной темой для визуализации является «Области интегрирования в разных системах координат» [1]. Разделы математического анализа, в которых изучаются поверхности второго порядка, более остальных нуждаются в визуальной презентации, и разрабатываемые по ним программы занимают одно из ключевых мест в онлайн-презентациях VisualMath.ru.

Библиотека Grafar была специально спроектирована для нужд математической визуализации и имеет возможности задания как явных, так и неявных функций более двух переменных. Для реализации трёхмерной графики используется технология WebGL. Для ускоренного отображения объектов применяется реактивное программирование [2], при котором пересчитываются только изменённые параметры, а остальные используют сохранённые

кэшированные данные. Также в библиотеке предусмотрено построение дискретных и непрерывных поверхностей, которые визуальны выглядят как сеточные и объёмные фигуры соответственно.

С помощью методов Gfafar'a визуализируются трёхмерные и двумерные фигуры, которые демонстрируют динамику изменения интегральных областей при переводе из декартовой системы координат в цилиндрическую и сферическую системы. Важно отметить, что в online-курсах, школах, лекциях практически полностью отсутствует элемент визуализации по этой теме, в то время как решение очень многих задач осуществляется через понимание геометрии математических объектов.

Несколько результатов работы программ представлены в следующем разделе. К условиям задач прикладываются фрагменты браузерных страниц с визуализированными моделями.

2 Примеры

2.1 Пример 1

Рассматривается задача поиска объёма тела, ограниченного следующими поверхностями:

$$x^2 + y^2 + z^2 = a^2,$$

$$x^2 + y^2 + z^2 = b^2,$$

$$x^2 + y^2 = z^2, z \geq 0, 0 < a < b$$

Эта задача решается с помощью вычисления тройного интеграла. Исходные поверхности и итоговая область интегрирования представлены на рис.1.

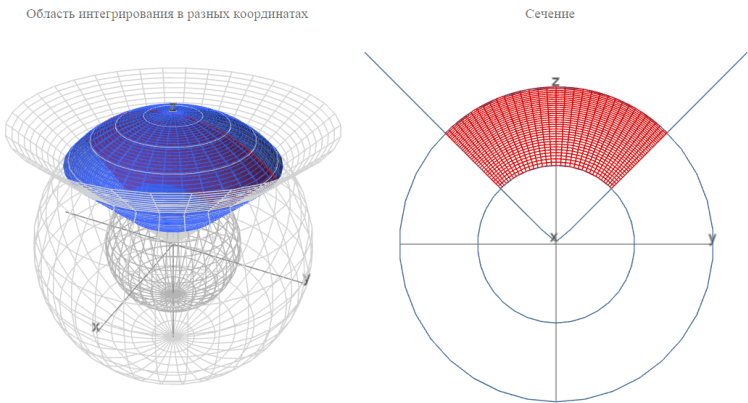


Рис. 1. Пример 1

Благодаря комбинированию дискретного и непрерывного типов параметризации осуществляется построение фигур разной текстуры: исходные поверхности строятся с помощью дискретной параметризации, результирующая фигура (область интегрирования) – с помощью непрерывной параметризации. В дополнение, двумерный график сечения позволяет провести более детальное изучение полученной области. Трёхмерная и двумерная фигуры являются динамичными и изменяются в зависимости от изменения параметров (а и b в данном примере) для более качественной демонстрации геометрического строения фигуры.

2.2 Пример 2

Другая задача поиска объёма тела, ограниченного следующими поверхностями:

$$z = 2a - x^2 - y^2, a > 0$$

$$x^2 + y^2 = z^2$$

Исходные поверхности и результирующая область интегрирования представлены на рис.2.

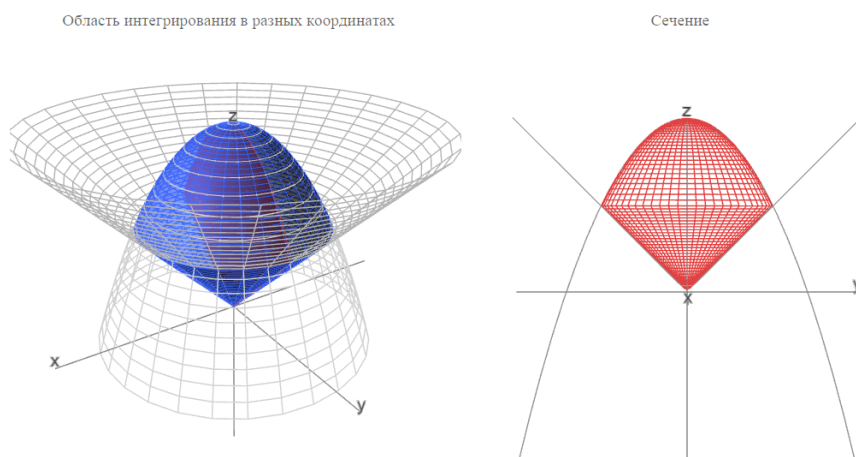


Рис. 2. Пример 2

2.3 Пример 3

Задача поиска объёма тела, заданного следующей функцией:

$$\left(\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} + \alpha^2\right)^2 = \frac{4z^2}{c^2}, \alpha < 1$$

Фигура представлена на рис. 3.

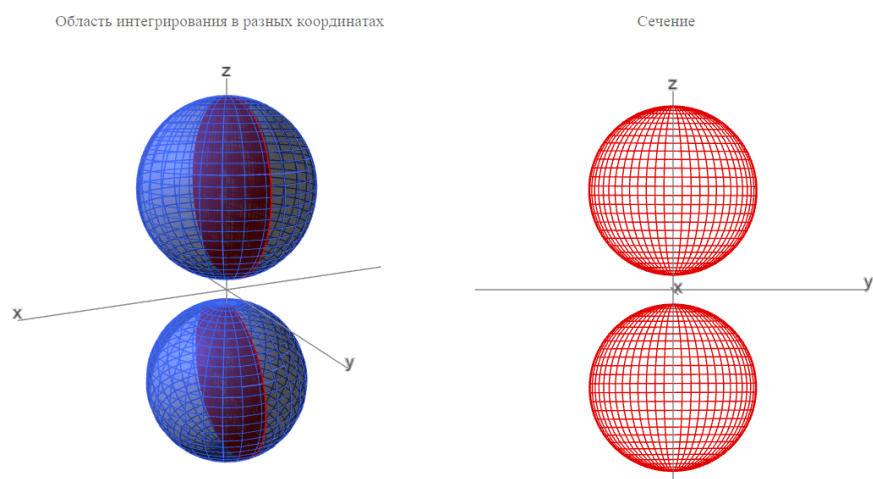


Рис. 3. Пример 3

3 Заключение

Алгебраические поверхности и фигуры, полученные при их пересечении, являются одной из ключевых тем в онлайн-презентациях проекта VisualMath.ru, так как связанные с ними темы математического анализа вызывают много трудностей у студентов. Разрабатываемые в рамках проекта визуальные модели проектируются таким образом, чтобы максимально понятно и наглядно показать природу алгебраических поверхностей и тем самым помочь студентам научиться соотносить формулы с соответствующими пространственными объектами.

Список литературы

1. Никитин, А. А. Сборник листочков с задачами к практическим занятиям по математическому анализу. — М.: Университетская книга, 2016. 51 с.
2. Elliott, C. Functional reactive animation // In ACM SIGPLAN Notices, New York, USA. 1997. С. 263–273.

Компактные разностные аппроксимации для задачи Штурма-Лиувилля

Цымбалов Е.,
студент второго курса магистерской программы «Науки о данных»

Аннотация. Для задачи Штурма - Лиувилля с переменным коэффициентом получены компактные разностные схемы, обладающие свойством l^2 -самосопряженности. Результаты численных экспериментов указывают на повышенную точность и порядок сходимости схем по сравнению с классической дивергентной.

Ключевые слова: компактные разностные схемы, тестовые функции, самосопряженность

1 Введение

Рассматривается компактная разностная аппроксимация линейного обыкновенного уравнения второго порядка с переменным коэффициентом на отрезке $[-\pi; \pi]$:

$$-d_x \theta(x) d_x u = f(x), \quad (1)$$

где $\theta(x), f(x)$ - известные функции, $\theta(x) > 0$ - коэффициент дифференциального уравнения, $f(x)$ - правая часть, с граничными условиями первого рода.

Краевые задачи для линейных дифференциальных уравнений второго порядка с переменными коэффициентами встречаются в разнообразных практических приложениях, таких как стационарные задачи теплопроводности, диффузии, электростатики, упругости в неоднородных средах. Особенно трудны для счета задачи, где коэффициент в главном члене (здесь $\theta(x)$) меняется на несколько порядков и/или имеет кусочно-гладкий вид. В рассматриваемом одномерном случае эффективность схемы не столь актуальна, однако при переходе к многомерным задачам эффективность — соотношение между количеством арифметических операций и погрешностью решения — является важной практической задачей.

В данной работе мы ограничиваемся рассмотрением самосопряженных разностных операторов, так как спектр оных является гарантированно вещественным.

2 Случай переменного гладкого коэффициента

Аппроксимируем дифференциальное уравнение (1) следующим образом:

$$R_{j,+}u_{j+1} + R_{j,0}u_j + R_{j,-}u_{j-1} = p_j f_{j+1} + q_j f_j + r_j f_{j-1}, \quad j = 1 \dots N, \quad (2)$$

где N – число узлов равномерной сетки на отрезке $[-\pi; \pi]$, $h = 2\pi N^{-1}$ – шаг сетки.

Сформулируем ряд требований к коэффициентам разностной схемы (2):

- l^2 – самосопряженность (поскольку (1) имеет L^2 – самосопряженную форму):

$$R_{j+1,-} = R_{j,+}, \quad j = 1 \dots N. \quad (3)$$

- На константе ($u(x) = \text{const}$) оператор (2) должен дать тождественный нуль:

$$R_{j,+} + R_{j,0} + R_{j,-} = 0, \quad j = 1 \dots N. \quad (4)$$

- Все функции $R_{j,+}$ должны быть однородными первого порядка по переменным $\{\theta(x_j)\}_{j=1}^{n-1}$. Общий вид таких функций:

$$R_{j,+} = \int_{-\infty}^{\infty} \sigma(\gamma) \theta^\gamma(x_j) \theta^{1-\gamma}(x_{j+1}) d\gamma, \quad (5)$$

где весовая функция $\sigma(\gamma)$ удовлетворяет нормировочному условию $\int_{-\infty}^{\infty} \sigma(\gamma) d\gamma = 1$. Весовую функцию σ выберем позднее.

- Коэффициент дифференциального уравнения (1) локально аппроксимируется в окрестности центральной точки шаблона (например, при $x=0$) экспонентой: $\theta(x) \approx A \exp(ax + bx^2)$. Если известны значения функции $\rho(x)$ в трех точках шаблона, то можно приравнять:

$$\rho(-h) = A \exp(-ah + bh^2), \quad \rho(0) = A, \quad \rho(h) = A \exp(ah + bh^2).$$

В этом случае можно выразить коэффициенты разностной схемы:

$$R_+ = \int_{-\infty}^{\infty} \sigma(\gamma) \theta_0^\gamma \theta_+^{1-\gamma} d\gamma = A \int_{-\infty}^{\infty} \sigma(\gamma) \exp\left[(1-\gamma)(ah + bh^2)\right] d\gamma, \quad (6)$$

$$R_- = \int_{-\infty}^{\infty} \sigma(\gamma) \theta_0^\gamma \theta_-^{1-\gamma} d\gamma = A \int_{-\infty}^{\infty} \sigma(\gamma) \exp\left[(1-\gamma)(-ah + bh^2)\right] d\gamma, \quad R_0 = -(R_+ + R_-).$$

- Равенство (2) должно быть верным для четырех пар тестовых функций

$$\langle u_k(x), f_k(x) \rangle, k=1 \dots 4, \text{ где } u_k(x) = x^k, \quad (7)$$

а правые части получаются по формуле (1):

$$f_k(x) = kA \exp(ax + bx^2) x^{k-2} [(k-1) + x(a + 2bx)].$$

Вместо интеграла по параметру γ с функцией σ общего вида, в формуле (6) возьмем сумму двух вариантов определения коэффициентов (со степенями γ_1 и γ_2):

$$R_+ = \alpha \theta_+^{\gamma_1} \theta_{1-}^{1-\gamma_1} + \beta \theta_+^{\gamma_2} \theta_{-}^{1-\gamma_2}, \quad R_- = \alpha \theta_+^{\gamma_1} \theta_{-}^{1-\gamma_1} + \beta \theta_+^{\gamma_2} \theta_{-}^{1-\gamma_2}, \quad R_0 = -(R_- + R_+).$$

Мы получили 4 уравнения ($k=1, 2, 3, 4$) на 5 коэффициентов: p, q, r, α, β . Решив уравнения, полученные подстановкой тестовых функций (7) в (2), получим коэффициенты для семейства разностных схем (2), зависящие от параметров γ_1, γ_2 .

Полученная компактная схема в численных экспериментах сравнивается с классической дивергентной схемой

$$\theta(x_{j-1/2})u(x_{j-1}) - [\theta(x_{j-1/2}) + \theta(x_{j+1/2})]u(x_j) + \theta(x_{j+1/2})u(x_{j+1}) = f(x_j)h^{-2} \quad (8)$$

Полученную схему легко обобщить на случай аппроксимации дифференциального уравнения с младшим членом (потенциалом) $q(x)$:

$$-d_x \rho(x) d_x u + q(x)u = f. \quad (9)$$

Численные эксперименты для уравнения (9) вместо (8) показывают схожие результаты в этом случае; амплитуда колебаний потенциала $q(x)$ на относительные результаты погрешности схем не влияет.

3 Случай кусочно-постоянного коэффициента

3.1 Осреднение коэффициента в точке разрыва полусуммой

Аппроксимируем задачу Штурма-Лиувилля на отрезке $[-\pi; \pi]$ со скачком в коэффициенте в точке $x = 0$:

$$\begin{aligned} \theta_- \Leftarrow x < 0 \\ -d_x \theta d_x u = f, \quad \theta = \frac{\theta_+ + \theta_-}{2} \Leftarrow x = 0 \\ \theta_+ \Leftarrow x > 0 \end{aligned} \tag{10}$$

Значение коэффициента $\theta(0)$ не влияет на решение дифференциальной задачи (10). Стыковочные условия (при $x = 0$) для решения дифференциального уравнения (10):

$$\begin{aligned} u_{-0} &= u_{+0}; \\ \theta_-(d_x u)_- &= \theta_+(d_x u)_+ \end{aligned} \tag{11}$$

Компактную разностную схему в точке скачка $x = 0$ строим снова на трехточечном шаблоне:

$$au(h) + bu(0) + cu(-h) = pf(h) + qf(0) + rf(-h), \tag{12}$$

где a, b, c, p, q, r – неопределенные коэффициенты, определяемые с точностью до общего множителя. Выберем в качестве нормировки этого набора коэффициентов условие $b = -(\theta_- + \theta_+)$.

Для определения остальных пяти коэффициентов необходимо потребовать выполнения равенства (12) на следующих пяти тестовых функциях:

Таблица 1. Тестовые функции для вычисления коэффициентов (12)

#	u	f	Уравнение на коэффициенты (12)
(13)	1	0	$a + b + c = 0$
(14)	$\theta_+ x \Leftarrow x < 0$ $\theta_- x \Leftarrow x \geq 0$	0	$a\theta_- x - c\theta_+ x = 0$
(15)	x^2	$-2\theta_- \Leftarrow x < 0$ $-(\theta_+ + \theta_-) \Leftarrow x = 0$ $-2\theta_+ \Leftarrow x > 0$	$(a + c)h^2 = -2[p\theta_+ + \frac{q}{2}(\theta_+ + \theta_-) + r\theta_-]$
(16)	$x^2 \operatorname{sign}(x)$	$2\theta_- \Leftarrow x < 0$ $(\theta_- - \theta_+) \Leftarrow x = 0$ $-2\theta_+ \Leftarrow x > 0$	$(a - c)h^2 = -2[p\theta_+ + \frac{q}{2}(\theta_+ - \theta_-) - r\theta_-]$
(17)	x^3	$-6\theta_- \Leftarrow x < 0$ $0 \Leftarrow x = 0$ $-6\theta_+ \Leftarrow x > 0$	$(a - c)h^3 = -6h[p\theta_+ - r\theta_-]$

Решая систему линейных алгебраических уравнений относительно неизвестных a, b, c, p, q, r , получаем следующее решение:

$$a = \theta_+, b = -(\theta_- + \theta_+), c = \theta_-, p = r = -h^2 / 6, q = -2h^2 / 3.$$

Заметим, что сумма коэффициентов правой части схемы равна $-h^2$ и не зависит от функции θ .

3.2 Осреднение коэффициента в точке разрыва средним геометрическим

Рассмотрим другое осреднение коэффициента θ в точке разрыва:

$$\begin{aligned} \theta_- \Leftarrow x < 0 \\ \theta = \sqrt{\theta_- \theta_+} \Leftarrow x = 0 \\ \theta_+ \Leftarrow x > 0 \end{aligned} \tag{18}$$

Тогда уравнение (15) заменяется на

#	u	f	Уравнение
(19)	x^2	$-2\theta_- \Leftarrow x < 0$ $-2\sqrt{\theta_+ \theta_-} \Leftarrow x = 0$ $-2\theta_+ \Leftarrow x > 0$	$(a+c)h^2 = -2[p\theta_+ + q\sqrt{\theta_+ \theta_-} + r\theta_-]$

По сравнению с решением в предыдущем разделе здесь изменятся только коэффициенты p и r :

$$p = -\frac{h^2}{6\theta_+}[\theta_+ + 2\theta_- + 2\sqrt{\theta_+ \theta_-}]; r = -\frac{h^2}{6\theta_-}[\theta_- + 2\theta_+ + 2\sqrt{\theta_+ \theta_-}].$$

3.3 Однопараметрическое осреднение значения коэффициента в точке разрыва

Рассмотрим более общий случай осреднения коэффициента в точке разрыва:

$$\begin{aligned}
 &\theta_- \Leftarrow x < 0 \\
 &\theta = (\theta_- - \theta_+) \left(\frac{2\sqrt{\theta_+ \theta_-}}{\theta_- + \theta_+} \right)^\omega \Leftarrow x = 0 \\
 &\theta_+ \Leftarrow x > 0
 \end{aligned}
 \tag{20}$$

Тогда уравнение (15) заменяется на

#	<i>u</i>	<i>f</i>	Уравнение
(21)	x^2	$-2\theta_- \Leftarrow x < 0$ $-2(\theta_- - \theta_+) \left(\frac{2\sqrt{\theta_+ \theta_-}}{\theta_- + \theta_+} \right)^\omega \Leftarrow x = 0$ $-2\theta_+ \Leftarrow x > 0$	$(a+c)h^2 = -2[p\theta_+ +$ $+ \frac{q}{2}(\theta_- - \theta_+) \left(\frac{2\sqrt{\theta_+ \theta_-}}{\theta_- + \theta_+} \right)^\omega + r\theta_-]$

По сравнению с решением в предыдущей секции, изменятся только формулы для коэффициентов правой части, а именно:

$$\begin{aligned}
 p &= \frac{h^2}{6} \left[\left(\frac{2\sqrt{\theta_+ \theta_-}}{\theta_- + \theta_+} \right)^{-\omega} - 2 - \frac{\theta_-}{\theta_+} + \frac{\theta_-}{\theta_+} \left(\frac{2\sqrt{\theta_+ \theta_-}}{\theta_- + \theta_+} \right)^{-\omega} \right]; \\
 q &= -\frac{2h^2}{3} \left(\frac{2\sqrt{\theta_+ \theta_-}}{\theta_- + \theta_+} \right)^{-\omega}; \\
 r &= \frac{h^2}{6} \left[\left(\frac{2\sqrt{\theta_+ \theta_-}}{\theta_- + \theta_+} \right)^{-\omega} - 2 - \frac{\theta_+}{\theta_-} + \frac{\theta_+}{\theta_-} \left(\frac{2\sqrt{\theta_+ \theta_-}}{\theta_- + \theta_+} \right)^{-\omega} \right].
 \end{aligned}$$

4 Численные эксперименты

4.1 Гладкий переменный коэффициент

Численный эксперимент по сравнению компактной разностной схемы (2) и классической дивергентной (8) проведем на решениях (1):

$$\begin{aligned}
 u &= \sin^2(x); \; \theta(x) = x^3 + 1; \\
 f &= -\cos(2x)(2x^3 + 2) - 3x^2\sin(2x).
 \end{aligned}
 \tag{22}$$

$$\begin{aligned} u &= \sin(x) \cos^2(x); \theta(x) = K \cos^2(x) + 1; \\ f &= -\sin(x)[9K - 24K \sin^2(x) + 15K \sin^4(x) - 9 \sin^2(x) + 7]. \end{aligned} \tag{23}$$

В (23) числовой параметр K регулирует амплитуду колебаний коэффициента $\theta(x)$.

Таблица 2. Результаты численных экспериментов для решений (22), (23). Здесь и далее погрешность указана в чебышевской (C) норме.

	Схема	(22)	(23), K= 10	(23), K = 1	(23), K= 0.01	(23), K=0.0001
Погрешность (C) при N = 100	Классик	1.31-3	7.65-4	7.76-4	7.98-4	7.99-4
	Компакт	2.86-4	1.05-3	3.45-4	4.78-6	1.39-6
Погрешность (C) при N = 1000	Классик	1.29-5	7.49-6	7.61-6	7.83-6	7.83-6
	Компакт	2.80-6	1.02-5	3.33-6	3.31-8	9.06-10
Порядок сходимости	Классик	2.0086	2.009	2.009	2.009	2.009
	Компакт	2.0102	2.014	2.017	2.176	3.422
Отношение погрешностей при N = 1000 (классик / компакт)		4.6	0.7	2.3	236.5	8645.6

4.2 Кусочно-постоянный коэффициент

Рассматриваются следующие варианты схем:

- классическая дивергентная;
- классическая дивергентная с простым условием стыковки: правая часть – нулевая, левая: $a = \theta_+, b = -(\theta_- + \theta_+), c = \theta_-$;
- компактная со средним арифметическим усреднением;
- компактная со средним геометрическим усреднением;
- компактная с простым условием стыковки;
- компактная с однопараметрическим усреднением ($\omega = -0.1$);

Схемы сравниваются на следующем решении:

$$u = \cos(\theta x + \pi / 3); \theta(x) = \begin{cases} \theta_-, x < 0 \\ \theta_+, x > 0 \end{cases} \tag{24}$$

Таблица 3. Результаты численных экспериментов для решения (24) при различных $\theta_{\pm}$.
 Жирным шрифтом выделены схемы с наименьшей погрешностью.

	Схема	$\theta_- = 1;$ $\theta_+ = 2$	$\theta_- = 1;$ $\theta_+ = 17$	$\theta_- = 3;$ $\theta_+ = 11$	$\theta_- = 2;$ $\theta_+ = 4$
Погрешн. (C) при N = 100	Классик	2.6176+0	4.5609+0	7.1472+0	4.0907+0
	Классик + стык.	2.6329+0	4.5872+0	7.4481+0	4.2128+0
	КС ср. арифм.	4.1172-2	7.9520-1	1.0491+0	1.6685-1
	КС ср. геом.	8.9445-2	1.3258-1	1.0761+0	3.6166-1
	КС + стык.	1.1034-1	1.3585+0	2.2213+0	4.5521-1
	КС однопарам.	1.3324-2	6.4384-1	1.0734+0	1.0073-1
Погрешн. (C) при N = 1000	Классик	2.5985+0	3.9375+0	6.4356+0	3.9899+0
	Классик + стык.	2.6000+0	3.9400+0	6.4652+0	4.0022+0
	КС ср. арифм.	4.1119-3	6.6151-2	9.8657-2	1.6457-2
	КС ср. геом.	8.7045-3	1.7859-2	1.0269-1	3.4937-2
	КС + стык.	1.0727-2	1.0204-1	1.8717-1	4.3059-2
	КС однопарам.	1.1455-4	1.7000-2	1.9315-2	1.0104-3
Порядок сходимос- ти	Классик	3.4225-3	6.8199-2	4.9082-2	1.1736-2
	Классик + стык.	5.8712-3	7.0574-2	6.6174-2	2.4025-2
	КС ср. арифм.	1.0006+0	1.0877+0	1.0295+0	1.0066+0
	КС ср. геом.	1.0126+0	8.4433-1	1.0185+0	1.0158+0
	КС + стык.	1.0132+0	1.1319+0	1.0795+0	1.0259+0
	КС однопарам.	2.0509+0	1.5619+0	1.7439+0	1.9894+0

5 Заключение

Для задачи Штурма-Лиувилля представлены новые компактные разностные схемы. Результаты численных экспериментов указывают на превосходство полученных схем над классическими по точности и порядку сходимости.

Полученные результаты создают задел для решения более сложных задач, таких как уравнение поперечных колебаний стержня:

$$\rho \frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial x} [R^2(x) \rho \frac{\partial^3 u}{\partial x \partial t^2}] + \frac{\partial^2}{\partial x^2} [E R^2(x) \frac{\partial^2 u}{\partial x^2}] = f,$$

где ρ - плотность, R – радиус, E – модуль Юнга стержня.

6 Список литературы

1. Самарский А.А. Введение в теорию разностных схем. – М.: Наука, 1971.

2. Гордин В.А. Математика, компьютер, прогноз погоды и другие сценарии математической физики. – М.: Физматлит, 2010.
3. Gordin V.A. Tsymbalov E.A. Compact differential schemes for the diffusion and Schrödinger equations. Approximation, stability, convergence, effectiveness, monotony // Journal of Computational Mathematics. - 2014 г. - Т. 32 (3). 348-370.
4. Spatz W.F. High-Order Compact Finite Difference Schemes for Computational Mechanics: PhD Thesis / University of Texas at Austin. - Austin, TX, 1995.

Тезисы подготовлены в ходе проведения исследования (16-05-0069) в рамках Программы «Научный фонд Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ)» в 2016 – 2017 гг. и с использованием средств субсидии на государственную поддержку ведущих университетов Российской Федерации в целях повышения их конкурентоспособности среди ведущих мировых научно-образовательных центров, выделенной НИУ ВШЭ.

Метод конвертации сервисных автоматов в модели на языке BPEL

Ким Полина Владиславовна

Отделение прикладной математики и информатики
на факультете компьютерных наук
группа 121
pvkim@edu.hse.ru

Ключевые слова: сервисные автоматы, BPEL, преобразование автоматов

1 Введение

Язык исполнения бизнес-процессов, представленных в виде веб-сервисов (Web Services Business Process Execution Language, BPEL), предназначен для описания бизнес-процессов (далее процессов или сервисов) и взаимодействия между процессами. Существуют исследования, которые рассматривают задачу проверки корректности взаимодействия процессов и их внутреннего поведения. Исследования обусловлено сложностью внутренней структуры процесса и модели взаимодействия процессов.

Непосредственный анализ BPEL кода требует значительных усилий, некоторые работы [1], [2], [3] предлагают двухэтапный подход для проверки корректности поведения моделей в форме BPEL кода. На первом шаге код преобразуется в графовую модель (сеть Петри, конечный автомат, UML-диаграмму деятельности и т.д.). На втором этапе проводится анализ полученной модели.

Также существует другой подход: конвертация заведомо корректной модели в BPEL код [4], [5]. Существуют различные способы [6], [7], [8] проверки корректности поведения как отдельных сервисов, так и композиции сервисов в форме конечных автоматов или сетей Петри.

Мы рассматриваем в качестве исходной модели для преобразования конечные автоматы, которые, в отличие от сетей Петри, являются более низкоуровневым описанием сервисов и не допускают параллельного исполнения. Таким образом, конечные автоматы являются эффективным средством для представления и анализа поведения бизнес-процессов.

2 Определения

В данной работе в качестве сервисного автомата мы рассматриваем классический конечный автомат. На рис. 1 изображен пример сервисного автомата.

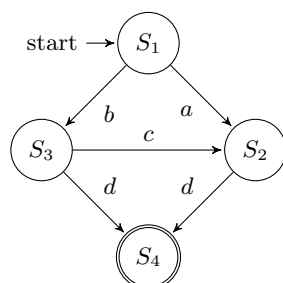


Рис. 1. Пример сервисного автомата S .

Определение 1. Будем называть автоматы A и B эквивалентными, если они описывают одинаковое поведение — порождают один и тот же язык: $L(A) = L(B)$.

Определим три правила для редукции (сокращения размера) автомата: для а) последовательности, б) выбора (разветвление) и в) цикла. Каждое правило редукции объединяет часть автомата в BPEL блок — последовательность, цикл или выбор.

Правила редукции иллюстрирует рис. 2, где часть автомата, соответствующий одному из BPEL блоков, будет заменена одним состоянием.

Определение 2. Структурированный цикл — это цикл с одним состоянием входа и с одним состоянием выхода из цикла. Соответственно, неструктурированный цикл — это цикл со многими состояниями входа/выхода.

В общем случае, редукция цикла в автомате является нетривиальной задачей. В [9] был предложен метод трансформации неструктурированных циклов в структурированные. Суть метода заключается в составлении системы уравнений, описывающей каждое состояние UML-диаграммы деятельности, с последующим преобразованием диаграммы, чтобы получился структурированный цикл с булевыми формулами на вершинах, который полностью описывал бы исходный неструктурированный цикл. Мы используем данный подход в нашей работе.

Определение 3. Хорошо структурированным автоматом назовем такой автомат, который можно разделить на BPEL блоки без трансформации исходного автомата.

Другими словами, если автомат с помощью правил редукции можно свернуть в единственную вершину, то такой автомат является хорошо структурированным (рис. 3). Простейшим примером хорошо структурированного автомата является *дерево*.

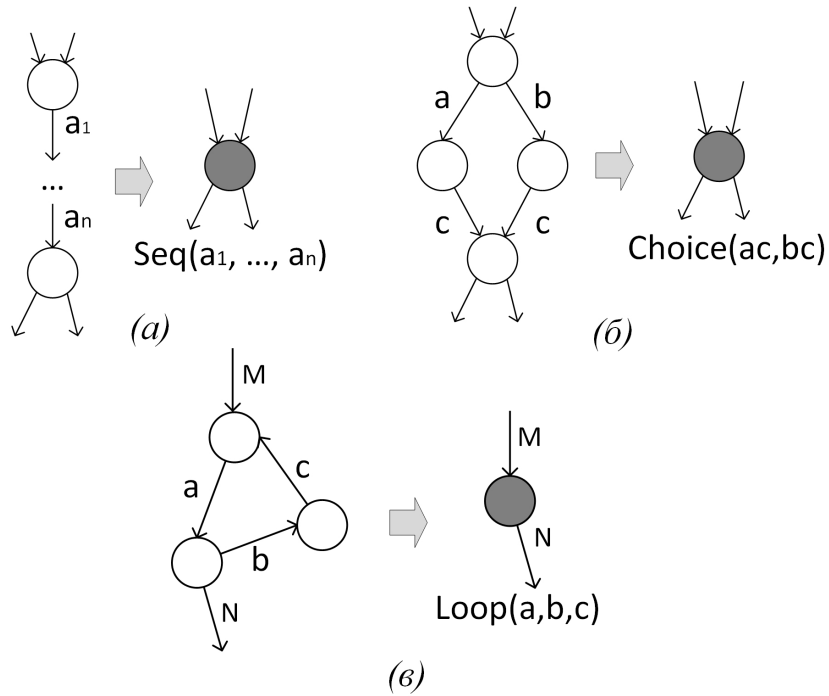


Рис. 2. Правила редукции автомата: а) последовательность, б) выбор, в) цикл.

2.1 Язык BPEL

BPEL [10] — структурный язык, позволяющий моделировать синхронное и асинхронное взаимодействие сервисов.

Каждый процесс состоит из этапов — активностей. В BPEL различают два вида этапов, называемых *активностями*: базовые и структурированные. Примерами базовых активностей могут быть: `<invoke>`, `<receive>`, `<response>`, которые используются для обмена сообщениями между процессами. К структурированным относятся `<sequence>`, `<flow>`, `<if>`, `<pick>`, `<while>`, `<until>`, `<for>`, они соответствуют последовательности активностей, параллельному потоку, выбору и циклам.

3 Трансформация автомата

Лемма 1. *Любой автомат S может быть представлен эквивалентным хорошо структурированным автоматом $\tilde{S}$.*

Доказательство. Рассмотрим автомат S . Найдем все компоненты сильной связности в S и заменим каждую из них на одно состояние, которое будет

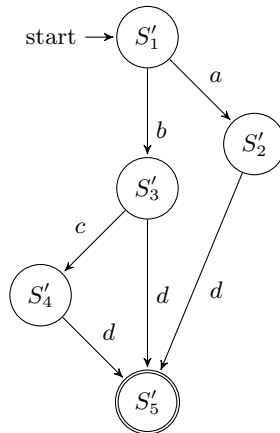


Рис. 3. Хорошо структурированный автомат S' , который эквивалентен автомату S (рис. 1).

включать в себя эту компоненту сильной связности. Таким образом, получим ациклический граф $\tilde{S}$.

Чтобы получить хорошо структурированный автомат $\tilde{S}$ эквивалентный $\tilde{S}$, возьмем язык $L(\tilde{S})$ и построим префиксное дерево $\tilde{S}$, которое будет полностью описывать слова (трассы) из $L(\tilde{S})$. В $\tilde{S}$ любой путь из начального состояния (корня) до конечного (листа) будет описывать одно слово из $L(\tilde{S})$.

Заметим, что хорошо структурированным автоматом, описывающим поведение исходного может быть не только дерево, но и автомат меньшего размера. Поскольку BPEL код состоит из структурных блоков, а внутренние элементы разных блоков в общем случае не соединяются ссылками (которые препятствуют получению «хорошей» структуры кода), задача состоит в преобразовании заданного автомата в эквивалентный хорошо структурированный. Так как хорошо структурированный автомат однозначно разбивается на BPEL блоки, он (полностью) конвертируется в BPEL код.

В [4] предлагают использовать `<link>`, как способ перехода между внутренними элементами разных блоков. Для этого сеть Петри представляется как `<flow>` блок, что не противоречит спецификацией BPEL языка. Предложенный в [4] подход обладает несколькими ограничениями: 1) предложенный алгоритм обрабатывает только структурированные циклы, поведение реальных систем зачастую описывается неструктурированными; 2) если алгоритм встречает элемент, который не подходит под шаблон компонентов последовательности, выбора, цикла или параллельного потока, тогда алгоритм приглашает пользователя самому конвертировать этот компонент сети в BPEL структуру.

Мы расширяем предложенный подход за счет обработки циклов и упрощения результирующего BPEL кода. Будем конвертировать в BPEL хорошо структурированные автомат, что позволит избежать использования `<link>`.

3.1 Метод трансформации автомата в хорошо структурированный автомат

Мы предлагаем разделить трансформацию заданного автомата на два этапа.

1. Определим сильно связанные компоненты графа, например, при помощи алгоритма Косарайю [11]. Если сильно связанная компонента описывает неструктурированный цикл, применим алгоритм [9]. Заменяем каждую компоненту одной вершиной, которая будет содержать цикл, описывающий эту компоненту.
Результатом первого этапа является ациклический автомат.
2. Если полученный ациклический автомат является хорошо структурированным, тогда его можно непосредственно конвертировать в BPEL код. Иначе, составим систему уравнений, описывающую все состояния заданного автомата, и с помощью этой системы построим хорошо структурированный автомат эквивалентный данному.

Будем строить систему уравнений как предложено в [9]. Тогда каждому состоянию автомата будет соответствовать одна переменная. Это уравнение отвечает за текущее состояние автомата. Опишем автомат уравнениями вида: $x_i = \text{if } \langle \text{условие (активность)} \rangle, \text{ then invoke активность, } x_j; i \neq j$, т.к. автомат ациклический.

В левой части уравнения стоит переменная, отвечающая за текущее состояние, а в правой — все возможные переходы из этого состояния.

Если текущее состояние x_i имеет несколько возможных продолжения, тогда оно соответствует BPEL структурам `<if>` или `<pick>`. Для каждого такого разветвления найдем состояние, в котором все продолжения сходятся. Будем использовать правила трансформации системы уравнений из [9], а именно, правилами *факторизации*, *замены переменных* и *разветвления*. Первое x_j , при котором достигается уравнение вида $x_i = \text{if } \langle \text{условие (активность)} \rangle, \text{ then invoke активность, } x_j; i \neq j$, будет состоянием, в котором заканчивается разветвление x_i . Для каждого разветвления будем строить свои внутренние состояния, внутренние состояния каждого блока не будут переходить во внутренние состояния других блоков.

Рассмотрим пример использования описанной системы уравнений. Для сервисного автомата на рис. 1 получим следующую систему:

$$\begin{aligned} s_1 &= \text{if } a, \text{ then invoke } a, s_2; \\ &\quad \text{if } b, \text{ then invoke } b, s_3; \\ s_2 &= \text{if } d, \text{ then invoke } d, s_4; \\ s_3 &= \text{if } c, \text{ then invoke } c, s_2; \\ &\quad \text{if } d, \text{ then invoke } d, s_4; \\ s_4 &= \text{End}; \end{aligned}$$

Заменяв s_2 в уравнении для s_3 и объединив два условия перехода для s_3 , получим

$$\begin{aligned} s_3 &= \text{if } c, \text{ then invoke } c; \\ &\quad \text{if } d, \text{ then invoke } d, s_4; \end{aligned}$$

Следовательно, в состояние s_4 сходятся все разветвления, выходящие из s_3 . Аналогично, для s_1 состояние s_4 так же будет завершать разветвление. Используя полученные блоки построим новый хорошо структурированный автомат (рис. 3), эквивалентный заданному.

4 Заключение

Был предложен метод конвертации сервисных автоматов в BPEL код. Отличительная особенность метода: он позволяет автоматически конвертировать любой конечный детерминированный автомат в BPEL структуру. Данный метод реализован в виде плагина для ProM [12]. Дальнейшая работа заключается в сравнении предложенного метода с алгоритмом [4], где модели будут построены по реальным логам систем.

Список литературы

1. Niels Lohmann Peter Massuthe Christian Stahl Daniela Weinberg. Analyzing Interacting BPEL Processes // Business Process Management, 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006, Proceedings. Vol. 4102 of Lecture Notes in Computer Science. Springer-Verlag, 2006. . P. 17–32.
2. Mateescu Radu, Rampacek Sylvain. Formal Modeling and Discrete-Time Analysis of BPEL Web Services / Ed. by Jan L. G. Dietz, Antonia Albani, Joseph Barjis. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008. P. 179–193. ISBN: 978-3-540-68644-6.
3. Fu Xiang, Bultan Tevfik, Su Jianwen. Analysis of Interacting BPEL Web Services. ACM Press, 2004. P. 621–630.
4. van der Aalst Wil M. P., Bisgaard Lassen Kristian. Translating Unstructured Workflow Processes to Readable BPEL: Theory and Implementation // Inf. Softw. Technol. 2008. Vol. 50, no. 3. P. 131–159.
5. Chun Ouyang Marlon Dumas Stephan Breutel Arthur H. M. ter Hofstede. Translating Standard Process Models to BPEL // Proceedings of the 18th International Conference on Advanced Information Systems Engineering. Vol. 4001 of Lecture Notes in Computer Science. Springer International Publishing, 2006. P. 417–432.
6. Massuthe Peter, Reisig Wolfgang, Schmidt Karsten. An Operating Guideline Approach to the SOA // 2nd South-East European Workshop on Formal Methods 2005 (SEEFM05). Ohrid, Republic of Macedonia, 2005.
7. Müller Richard Stahl Christian Aalst Wil M. P. Westergaard Michael. Service Discovery from Observed Behavior while Guaranteeing Deadlock Freedom in Collaborations / Ed. by Samik Basu, Cesare Pautasso, Liang Zhang, Xiang Fu. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013. P. 358–373. ISBN: 978-3-642-45005-1.
8. Lohmann Niels, Massuthe Peter, Wolf Karsten. Operating Guidelines for Finite-State Services // ICATPN. Vol. 4546 of Lecture Notes in Computer Science. Springer, 2007. P. 321–341.
9. Koehler Jana, Hauser Rainer. Untangling Unstructured Cyclic Flows - A Solution Based on Continuations // CoopIS/DOA/ODBASE (1). Vol. 3290 of Lecture Notes in Computer Science. Springer, 2004. P. 121–138.

10. BPEL standard specification. URL: <http://docs.oasis-open.org/wsbpel/2.0/0S/wsbpel-v2.0-0S.html>.
11. Sedgewick Robert. Algorithms in C. Part 5. , Graph algorithms. Boston (Mass), San Francisco (Calif.), New York (N. N.) : Addison-Wesley, 2002. ISBN: 0-201-31663-3.
12. Verbeek H.M.W. Buijs J.C.A.M. Dongen B.F. van Aalst W.M.P. van der. ProM 6: The Process Mining Toolkit // Proc. of BPM Demonstration Track 2010 / Ed. by M. La Rosa. Vol. 615 of CEUR Workshop Proceedings. 2010. P. 34–39.

Генерация предикторов для аттракторной модели внимания

Д.А. Филин, 3 курс ФКН ПМИ

Аннотация.

В данной работе рассматривается вопрос о генерации множества предикторов для ресурсной сети, используемой при распознавании и категоризации образов. Ресурсная сеть является аналогом известной нейронной сети Хопфилда, однако имеется ряд небольших отличий, связанных с реализацией данной сети под конкретную задачу. За основу данной работы взят алгоритм, основанный на принципах работы человеческого мозга при попытках распознать тот или иной образ [1]. Утверждается, что категоризация образа достигается посредством движения глаза по нему и выделения отличительных признаков, характерных для того или иного объекта. На каждом этапе алгоритма происходит “притяжение” образа к одному из заранее известных аттракторов, знания о которых накапливаются в процессе обучения. В результате нескольких движений глаза (в дальнейшем саккад) человек в состоянии категоризировать предложенный объект. В данной работе опыт проводится на так называемых stick-animal изображениях, то есть изображениях животных, состоящих из прямых линий. Ставится задача генерации отличительных признаков для каждого типа животных для обучения ресурсной сети. Для этого вводится понятие предиктора. Предиктор представляет собой структуру, состоящую из трех элементов: начальный признак, саккада, конечный признак. Именно предикторы играют ключевую роль в процессе распознавания изображений. В качестве примера можно рассмотреть пару конечностей животного (в частности собаки). Внешне передние и задние конечности совпадают. Поэтому, чтобы категоризировать эти объекты необходима саккада, например в правую сторону. Тогда если в область нашего зрения попадет хвост, мы можем понять, что это были задние конечности, если же мы видим продолжение туловища или же пустое место, то изначально наблюдались передние конечности (Рис. 1).

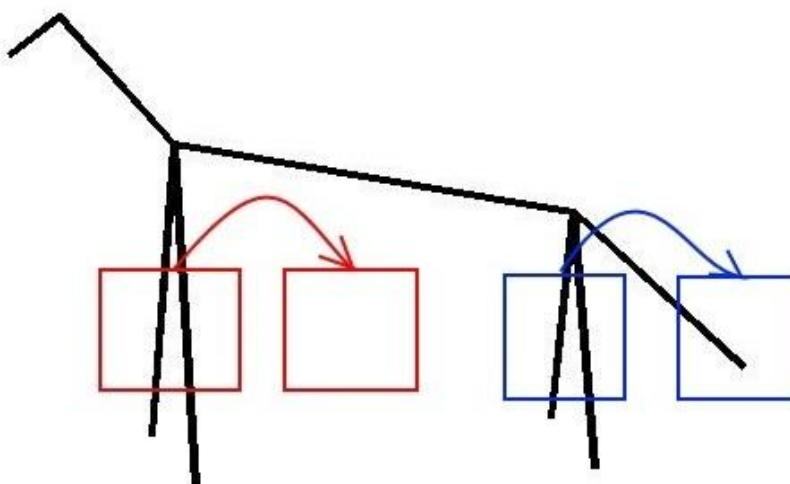


Рисунок 1 – Пример различия конечностей посредством саккады

Генерация предикторов.

Итак, поняв, что представляют из себя предикторы можно преступить к их генерации. Если выбрать определенную саккаду (то есть приращение по осям Ox, Oy на которую будем сдвигаться из одной области в другую) и сгенерировать всевозможные пары вида $(F_{src}, F_{src} + \delta)$, то очевидно получим много шума. Поэтому, как было сказано выше, нас будут интересовать места, определяющие исключительность того или иного объекта. Этими местами являются соединения конечностей. Так, например, у жирафа расстояние от головы до места соединения шеи, туловища и ног намного больше чем у какого-либо другого животного. И это является его отличительной чертой. Итак, встает задача о нахождении точек соединения на изображении. С точки зрения геометрии изображения данные области есть ни что иное как углы. Поэтому задача генерации предикторов сводится к задаче поиска углов на изображении.

Кратко о детекторе Харриса.

Для реализации поставленной цели существует множество средств. Однако не стоит забывать, что для обучения ресурсной сети необходимо использовать большое количество изображений, поэтому алгоритм должен работать достаточно быстро. В качестве инструмента был выбран детектор Харриса [2]. Данный детектор углов обладает такими преимуществами как простота реализации и инвариантность к линейным операциям на изображении. Пусть изображение задано

некоторой функцией интенсивности $I(x, y)$. Ограничимся черно-белыми изображениями, поэтому значение функции в каждой точке будет лежать в отрезке от 0 до 255. Допустим мы рассматриваем некоторую область $N \times N$ пикселей. Как изменится значение функции для каждой точки в этом окне, если мы немного сдвинем данную область (Рис. 2)? Для понимания работы алгоритма следует осознать несколько базовых концепций.

- Если изображение в данной области почти монотонно, то при сдвиге данной области на некоторую δ значение функции почти не изменится.
- Если в данной области лежит линия пересечения двух плоскостей с разными цветами, то движение окна вдоль данной прямой приведет опять же к малым изменениям, тогда как движение в перпендикулярном направлении даст большее изменение.
- В случае угла значение функции будет всегда сильно изменяться независимо от направления сдвига.

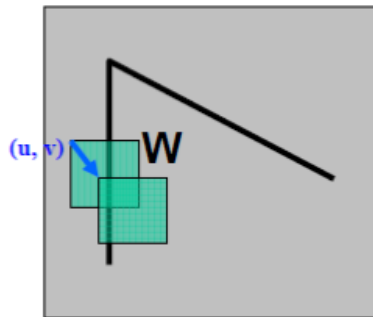


Рисунок 2 – Пример сдвига выбранной области на (u, v)

Введем следующую функцию:

$$E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

То есть мы рассматриваем квадрат изменения функции изображения в некотором окне W . Функцию $w(x, y)$ возьмем как бинарную, то есть

$$w = \begin{cases} 1, & (x, y) \in W \\ 0, & \text{иначе} \end{cases}$$

Далее воспользуемся разложением функции $I(x + u, y + v)$ в ряд Тейлора в точке (x, y) :

$$I(x + u, y + v) \approx I(x, y) + I_x(x, y)u + I_y(x, y)v$$

Тогда исходная функция примет вид:

$$\begin{aligned} E(u, v) &= \sum_{x,y} w(x, y) [I_x(x, y)u + I_y(x, y)v]^2 \\ &= \sum_{x,y} w(x, y) (u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2) \end{aligned}$$

Тогда в матричном виде:

$$E(u, v) = [u \ v] * \left(\sum_{x,y} w(x, y) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \left(\sum_{x,y} w(x, y) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right)$$

Как уже было сказано в случае угла малые изменения аргумента приводят к большим изменениям функции. Это значит, что собственные числа матрицы M должны быть большими по модулю. Так как вычисление собственных чисел довольно ресурсоемкая задача было предложено использовать следующую их оценку:

$$R = \det(M) - k * \text{trace}^2(M), k = [0.04; 0.06]$$

Итак, для поиска углов с помощью данного детектора необходимо задать следующие параметры: размер анализируемых детектором окон, размер ядра оператора Собеля (с помощью данного оператора находятся частные производные по изображению) и параметр k . При выполнении подобных задач размер ядра принято брать равным 3, значение k равным 0.04, а вот выбор размеров области сильно зависит от конкретной задачи. В данной проблеме этот параметр необходимо установить как можно меньше, так как изображения достаточно низкого разрешения и прямые имеют небольшую толщину. Иначе, если взять большой размер окна, почти все точки внутри него будут удовлетворять угловым, а нам интересна лишь одна точка в каждом месте соединения. Поэтому после отработки алгоритма необходимо провести чистку, а именно если в какой-то окрестности, определенной как угол, все-таки содержится более одной точки, то нужно оставить лишь одну. Размер данной окрестности был выбран опытным путем и составил 10x10 пикселей.

Результаты.

В результате работы алгоритма получилось сгенерировать базу предикторов, необходимых для обучения ресурсной сети. В дальнейшем планируется создание и обучение самой ресурсной сети и использование ее в задачах категоризации изображений. Результат работы алгоритма можно увидеть на рисунке 3.

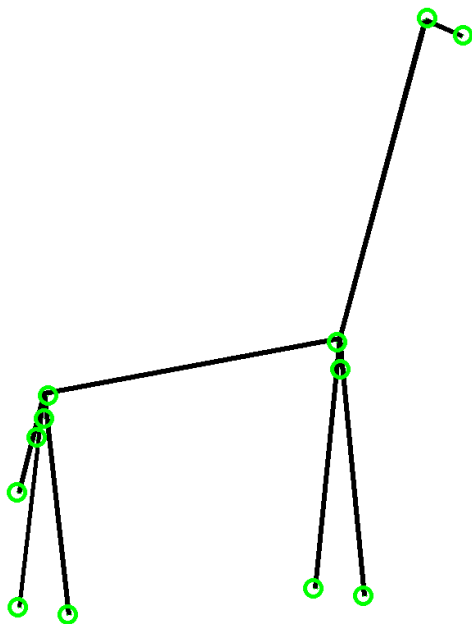


Рисунок 3 – Результат работы алгоритма поиска предикторов

Литература.

1. Nicola Catenacci Volpi, Jean Charles Quinton, Giovanni Pezzulo How active perception and attractor dynamics shape perceptual categorization: A computational model // Neural Networks. 2014. №60.
2. A COMBINED CORNER AND EDGE DETECTOR // The British Machine Vision Association and Society for Pattern Recognition URL: <http://www.bmva.org/bmvc/1988/avc-88-023.pdf> (дата обращения: 06.03.2016).

Граничные условия, имитирующие задачу Коши для разностных уравнений в частных производных

Шемендюк Александр Андреевич

Прикладная математика и информатика, БПИИ 131
alex.shemendyuk@gmail.com

Аннотация.

Алгоритм построения граничных условий полного поглощения волн, выходящих из вычислительной области, применяется к различным разностным аппроксимациям уравнений математической физики. Представлены результаты работы алгоритма и численные эксперименты с такими граничными условиями для волнового уравнения, аппроксимируемого явными и неявными разностными схемами.

Введение.

Обычно, физическая задача включает в себя дифференциальное уравнение (или систему уравнений) и граничное условие. Иногда вычислительная область выбрана так, что на границе не происходит никаких особенных физических явлений. Например, в задаче прогноза погоды, если мы рассматриваем её в ограниченной по горизонтали и (или) вертикали области, происходят известные физические процессы переноса вещества, теплообмена и прочие, а на границе нужно проводить стыковку решения с решением в фоновой области (например, на всем глобусе).

Здесь граничное условие Дирихле для разностных уравнений привело бы к полному отражению от границы в прогностическую область выходящих волн. В природе такого отражения нет. Таково же и граничное условие Неймана. Нужны неотражающие условия.

Такие граничные условия, как правило (если граница не характеристическая), существуют. Условия эти обычно получаются нелокальными по времени (требуется помнить всё, что было в прошлые моменты времени вблизи границы), поэтому появляется потребность в приближенной их локализации (помним лишь несколько прошедших моментов).

Мы создаем «альбом»: для разных дифференциальных уравнений и аппроксимирующих схем - шаблоны неотражающих граничных условий с приближенной локализацией и соответствующие погрешности.

Алгоритм.

Рассмотрим волновое уравнение

$$\partial_t^2 u = c^2 \partial_x^2 u \quad (1)$$

и его аппроксимацию по схеме leap-frog [1]:

$$u(t - \tau, x) - 2u(t, x) + u(t + \tau, x) = \nu^2 [u(t, x - h) - 2u(t, x) + u(t, x + h)], \quad (2)$$

где $\nu = c\tau/h$ - параметр Куранта.

После Z-преобразования (дискретный аналог преобразования Лапласа) получим обыкновенное разностное уравнение второго порядка:

$$z\tilde{u}(x + h) - (\nu^{-2}(z - 1)^2 + 2z)\tilde{u}(x) + z\tilde{u}(x - h) = -g(z, x), \quad (3)$$

где $g(z, x)$ содержит начальные данные.

Характеристическое уравнение для (3) имеет два корня, причем $|\lambda_1(z)| < 1 < |\lambda_2(z)|$ при больших $|z|$. Решение однородного уравнения (3) записывается как линейная комбинация двух степенных функций – первая растет налево и убывает направо, а вторая - противоположно:

$$Y(mh, z) = C_1(z)\lambda_1^m(z) + C_2(z)\lambda_2^m(z).$$

Для неоднородного уравнения по этой фундаментальной системе решений строится функция Грина с граничными условиями убывания на $\pm\infty$.

Граничное условие полного поглощения должно быть таким, чтобы удовлетворялось условие ограниченности продолженного решения как на $+\infty$, так и на $-\infty$. Оно ищется в виде:

$$1 + A(z)\lambda_2(z) = 0 \iff A(z) = -\lambda_1(z). \quad (4)$$

Тогда получаем при $x > 0$ имитацию задачи Коши с продолженным нулем (то есть $g(z, x) \equiv 0$ при $x < 0$) [2], $\lambda_1(z)$ - мероморфная в окрестности бесконечности функция [3].

Применяя обратное Z-преобразование, т. е. раскладывая $A(z)$ в ряд Лорана, получаем [2] граничное условие:

$$u(n\tau, 0) + \sum_{k=0}^n a_k u(t - k\tau, h) = 0, \quad (5)$$

где a_k - коэффициенты разложения Лорана функции $A(z)$.

Для любого момента времени в (5) используются все n предыдущих. При больших временах интегрирования, т. е. при больших n , это дорогой алгоритм. Разностное уравнение (2) аппроксимирует (1) с погрешностью, поэтому и граничное условие может быть приближенным. Локализация: используем в шаблоне граничного условия приграничные точки $x_j = jh$, $j = 0, 1, \dots, k$ и несколько последних моментов времени:

$$\sum_{j=0}^k \sum_{l=0}^{d_j} y_{jl} u(t - l\tau, jh) = 0, \quad y_{00} = 1. \quad (6)$$

Количество точек по времени, которое используется в граничном условии для j -ой приграничной точки сетки, равно степени d_j . Для вычисления коэффициентов y_{jl} используем векторную аппроксимацию Паде – определим коэффициенты многочленов Лорана $Y_j(z) = \sum_{l=0}^{d_j} y_{jl} z^l$, таких что:

$$\sum_{j=0}^k Y_j(z) \lambda_2^j(z) = O(z^{-N}), \quad Y_0(\infty) = y_{00} = 1.$$

По разложению λ_1 в ряд Лорана для заданных степеней d_j векторной аппроксимации Паде, решаем систему линейных уравнений относительно неизвестных коэффициентов y_{jl} .

Заключение.

Для волнового уравнения приведены результаты численных экспериментов с неотражающими граничными условиями (6) для различных разностных схем.

В каждом случае перечислен набор шаблонов приближенной локализации, их сравнение и выбор наилучшего. Критерий: исследуется энергия волны в рассматриваемой области – чем быстрее она убывает, тем лучше работает граничное условие (6).

Список литературы

1. Гордин, В. А. Краевое условие полного поглощения волн, выходящих за пределы прогностической области, для разностного уравнения в частных производных. / в сб. «Труды Гидрометцентра СССР», N242, С. 104–120 (1982).
2. Гордин, В. А. Математика, компьютер, прогноз погоды и другие сценарии математической физики— М.: ФИЗМАТЛИТ, 2013. 628 с.
3. Гордин, В. А. Применение векторной аппроксимации Паде при численном решении эволюционных прогностических уравнений. // Метеорология и гидрология. 1982. № 11. С. 24–37.

Работа подготовлена в ходе проведения исследования (16-05-0069) в рамках Программы «Научный фонд Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ)» в 2016 – 2017 гг. и с использованием средств субсидии на государственную поддержку ведущих университетов Российской Федерации в целях повышения их конкурентоспособности среди ведущих мировых научно-образовательных центров, выделенной НИУ ВШЭ.

Симуляция бизнес-процессов и генерация искусственных логов событий

Шугуров Иван Сергеевич

4 курс, БПИ122

Департамент программной инженерии
shugurov94@gmail.com

Keywords: генерация логов событий, симуляция бизнес-процессов, сети Петри, BPMN (Business Process Model and Notation), process mining

1 Введение

Постоянно растущая сложность современных информационных систем порождает необходимость в создании новых методов их формальной спецификации, моделирования и анализа. Это необходимо для обеспечения надёжности информационных систем и для возможной оптимизации процессов, как самой информационной системы, так и бизнес-процессов, осуществляющихся при помощи информационной системы. Одной из научных дисциплин, пытающихся решить подобные задачи, является process mining [1].

Целью process mining является разработка подходов и методов моделирования и оптимизации бизнес-процессов, основанных на анализе поведения информационных систем, который осуществляется путём автоматизированного исследования логов событий, создаваемых во время работы информационной системы. Преимуществом process mining по сравнению с классическими подходами к моделированию процессов является то, что модели строятся по реальным логам системы, а не по предположениям эксперта о том, как устроен процесс. Это позволяет лучше понимать суть процессов, понимать, что происходит в реальности, предоставляя возможность для эффективного улучшения процессов.

Большинство алгоритмов process mining требуют лог событий как один из входных параметров. Несмотря на то, что целью process mining является анализ реальных логов событий, они зачастую не подходят для верификации новых алгоритмов на начальных этапах разработки. Ещё одной проблемой является то, что зачастую трудно найти реальные логи с характеристиками, необходимыми исследователю. В то время как существует достаточное количество инструментов, позволяющих визуально создавать и редактировать модели процессов, наблюдается дефицит генераторов искусственных логов событий общего назначения. Таким образом, генерация искусственных логов событий по созданной модели, обладающей необходимыми характеристиками, позволяет осуществить более простое и эффективное тестирование и оценку правильности и эффективности разрабатываемых алгоритмов. В данном докладе представлена програм-

ма, которая позволяет генерировать искусственные логи событий по моделям процессов.

2 Симуляция и генерация искусственных логов

Реализованная программа поддерживает две популярные нотации, используемые для моделирования бизнес-процессов: сети Петри и BPMN. Сети Петри [2] это популярный в академической среде язык моделирования, имеющий строгое математическое определение. Из-за этого они являются основным способом представления процессов в process mining. В противоположность сетям Петри, язык BPMN [3] больше распространён в индустрии, так как является более гибким и выразительным языком – доступно большое количество элементов, в том числе и высокоуровневых (таких, которые компактно описывают сложные действия и которые зачастую проблематично выразить при помощи сетей Петри). Однако в последнее время результаты работы алгоритмов process mining всё чаще представляются в виде BPMN моделей [4], [5]. Пример BPMN модели представлен на рис.1, лог нескольких воспроизведений этой модели представлен в таблице 1.

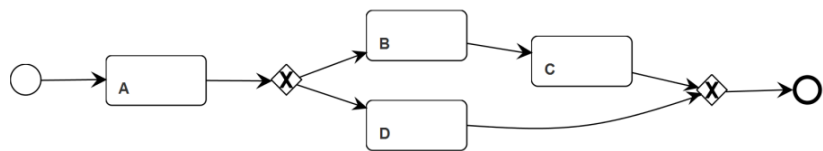


Рис. 1. Пример BPMN модели

Генератор логов был реализован как расширение к программной платформе ProM Framework. ProM это платформа с открытым исходным кодом для имплементации алгоритмов process mining в стандартизированном окружении. Для хранения сгенерированных логов событий был использован XML-подобный формат XES, который является стандартным форматом хранения логов для задач process mining. Согласно стандарту события описываются такими характеристиками как название события, время события, идентификатор ресурса, участвовавшего в выполнении активности и так далее. Однако часть из этих характеристик может быть и пропущена.

При любом выбранном типе генерации программа предоставляет пользователю ввести (1) желаемое количество логов, (2) количество выполнений процесса в каждом логе и (3) максимальное количество шагов при конкретном воспроизведении модели. Количество шагов ограничивается, чтобы бороться с возможными бесконечными циклами в моделях.

Таблица 1. Пример лога событий

Воспроизведение	Событие	Название	Время
3	1	A	01.03.2016

			22:46:17
3	2	B	02.03.2016 00:16:56
3	3	C	02.03.2016 01:38:26
6	1	A	01.03.2016 22:31:14
6	2	D	01.03.2016 23:36:05

2.1 Генерация логов по сетям Петри

Программная реализация предоставляет три возможных варианта генерации логов по сетям Петри.

«GENA: Basic log generator» позволяет воспроизводить сети Петри согласно их стандартному определению. Дополнительной функцией данного вида генерации является возможность умышленного добавления «шума» – отклонения логов от реального воспроизведения модели. Данная функция является полезной при тестировании алгоритмов, задачей которых является работа с неидеальнымилогами, или алгоритмов, которые определяют степень соответствиялога и модели. Отклонения могут быть составлены как из событий, которые соотносятся с элементами модели, так и отсутствующими записями о событиях, так и событиями, которые не соответствуют никаким элементам модели. Возможно настраивать степень искажения.

«GENA: Log generator with static priorities» – каждому переходу модели задаётся его приоритет. Для того, чтобы выбрать, какой переход сработает следующим, вначале находят набор переходов, который могут сработать согласно стандартным правилам сетей Петри, затем из них выбирается переход с наивысшим приоритетом.

«GENA: Time-Driven log generator» позволяет проводить полноценную симуляцию моделей с заданием времён срабатываний переходов, задании того, какие ресурсы необходимы для срабатывания перехода, синхронизация переходов по требуемым ресурсам. Так же предоставляется большое количество возможных опций искажения генерируемых логов.

2.2 Генерация логов по BPMN моделям

Программная реализация предоставляет пять возможных варианта генерации логов по BPMN моделям.

«GENA: Basic BPMN log generator» предоставляет базовую функциональность генерации логов по BPMN моделям. Данные тип генерации поддерживает элементы, необходимые для задания порядка выполнения элементов: activities, start and end events, parallel and exclusive gateways, flows; перспективы ресурсов:

pools, lanes, message flows; иерархической структуры: sub-processes, cancel events.

«GENA: Extended BPMN log generator» позволяет пользователю приоритизировать выбор, который производится на exclusive gateway. Благодаря этому возможно точнее смоделировать поведение в случаях, когда возможные пути выполнения процесса не равновероятны.

«GENA: BPMN log generator with time» позволяет задавать, сколько времени необходимо для выполнения activity как через интерфейс программы, так и при помощи скриптов на Python. При генерации времени используется равномерное распределение с заданными минимальными и максимальными значениями.

«GENA: BPMN log generator with data» предоставляет поддержку двух дополнительных элементов BPMN, отвечающих за моделирование потока данных: data objects и data associations. При использовании этого типа генерации изменяется поведение activity, exclusive choice gateway и sub-processes, если они имеют ассоциации с data objects. Activities могут читать и изменять значения в data objects. Exclusive gateways используют data objects для выбора между альтернативными путями выполнения процесса. Для предоставления гибкой модификации поведения подобных элементов была добавлена возможность задания поведения при помощи скриптов на языке Python.

Так же программа предоставляет возможность совмещать генерацию с данными и генерацию со временем.

3 Апробация программы

Генератор искусственных логов уже несколько лет используется сотрудниками лаборатории процессно-ориентированных информационных систем и студентами, пишущими курсовые работы в лаборатории. Программа расширяется по мере изменения запросов сотрудников лаборатории. Помимо этого, для проверки работоспособности генерации логов по BPMN диаграммам было проведено тестирование на моделях реальных бизнес-процессов, представленных в коллекции «Signavio BPMN Reference Models». Для тестирования были выбраны 1000 моделей, удовлетворяющих ограничениям программы. Тестирование на этих моделях показало, что генератор работает корректно и надёжно, и что скорость генерации позволяет создавать искусственные логи большого объёма, что является важной характеристикой в эпоху больших данных.

4 Заключение

По результатам тестирования и использования можно сделать вывод, что программа справляется со своими задачами. При разработке программы были использованы лучшие практики проектирования ПО, так что в случае необходимости добавления новых нотаций или изменении уже реализованных способов генерации это удаётся сделать просто и быстро. Детали реализации генера-

ции логов по сетям Петри были представлены на конференции для молодых исследователей [8]. Ознакомиться с программой можно по ссылке [9].

Список литературы

1. Van Der Aalst W. Process mining: discovery, conformance and enhancement of business processes. – Springer Science & Business Media, 2011.
2. Reisig W. Understanding Petri Nets. – Springer, Berlin, 2013. – Т. 4.
3. Спецификация BPMN / Object Management Group Online – Режим доступа: <http://www.omg.org/spec/BPMN/2.0/PDF/> свободный. (Дата обращения: 09.03.2016).
4. Kalenkova A., De Leoni M., van der Aalst W. M. P. Discovering, Analyzing and Enhancing BPMN Models Using ProM //BPM (Demos). – 2014. – С. 36.
5. Kalenkova A. A. et al. Process mining using BPMN: relating event logs and process models //Software & Systems Modeling. – 2015. – С. 1-30.
6. Verbeek H. M. W. et al. Prom 6: The process mining toolkit //Proc. of BPM Demonstration Track. – 2010. – Т. 615. – С. 34-39.
7. Günther C. W., Verbeek H. M. W. Xes-standard definition. – 2014.
8. Shugurov I., Mitsyuk A. A. Generation of a set of event logs with noise //Proceedings of the 8th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYR-CoSE 2014). – 2014. – С. 88-95.
9. Страница о генераторе логов / Лаборатория ПОИС – Режим доступа: <https://pais.hse.ru/research/projects/gena> свободный. (Дата обращения: 09.03.2016).

Разработка игрового искусственного интеллекта с имитацией реакции человека на появление противника в трехмерном лабиринте на Unreal Engine 4

Токмаков Михаил Александрович¹

образовательная программа "Науки о данных"
matokmakov@gmail.com

Ключевые слова: игровой искусственный интеллект; системы, основанные на правилах

1 Расширенная аннотация

Основной задачей современной видеоигры является предоставление игроку возможности полностью погрузиться в виртуальный мир. Для того чтобы достичь этого необходимо заставить игрока поверить в реальность происходящего. Именно поэтому разработчики стараются сделать виртуальный мир максимально правдоподобным. Одной из важнейших составляющих этого мира в 3D-шутерах являются компьютерные агенты (так называемые БОТы).

В соответствии с популярным определением, приведенным в [3], правдоподобность поведения компьютерного агента во многом зависит от следующих факторов: личностные качества, мотивация, человекоподобное поведение, под которым подразумевается наличие целей, планов, человеческие ограничения, реакция на изменение окружающей среды, социальные отношения.

Далеко не все из перечисленного может предложить игроку современная индустрия видеоигр. Разработчики все чаще ставят на первый план графическую и сюжетную составляющие, в то время как моделирование реалистичного игрового искусственного интеллекта становится второстепенной задачей. Чаще всего используется уже существующие модели компьютерных игроков, которые во многих случаях не позволяют добиться достаточного уровня реализма. Во многих играх мы можем наблюдать БОТов застревающих в текстурах, стреляющих в пустоту, и самое худшее – превосходство игроков-компьютеров над пользователем [6],[7],[8].

Данная работа является попыткой создать паттерн игрового искусственного интеллекта для 3D-шутера, который будет сочетать в себе человеческое и машинное распознавание образов для прицеливания, и модели принятия решения для обхода лабиринта и поведения во время сражения. Для определения траекторий передвижения БОТа по карте к данным, описывающим

траектории реальных игроков, применяются методы математической статистики и анализа данных. Функция, распознающая образы игроков использует знание геометрии лабиринта. При моделировании поведения игрока используются теории графов и принятия решений.

Подобные исследования уже проводились ранее, однако их цели были иными. На соревновании 2K BotPrize [11] команды представляли на суд жюри компьютерных игроков. Побеждали те команды, чьи боты проходили тест Тьюринга [1]. Судьи играли в 3D-шутер Unreal Tournament 2004 вместе с БОТаами, и каждый судья в процессе игры должен принять решения относительно каждого игрока – является ли он компьютерным агентом или же человеком. Победа присуждалась в том случае, если БОТу удавалось обмануть не менее половины судей. Таким образом представленные модели не были правдоподобными, а скорее создавали видимость правдоподобия. Так же ежегодно проводится конференция Artificial Intelligence in Creative Industries [12] на которой участники представляют множество работ по данной тематике. Мы попытались отойти от попытки имитировать человека без понимания принципов принятия решения в пользу более подробного проектирования действий БОТа.

Началом данного исследования послужил сборник статей Believable Bots [10]. Его подзаголовком является основная идея заложенная в данную работу: «Могут ли компьютеры играть как люди». Этот вопрос волнует как обычных пользователей так и программистов, занимающихся разработкой моделей игровых искусственных интеллектов. На основании работ из данного сборника и существующих на данный момент моделей мы построили модель, частично дающую ответ на данный вопрос.

Построена и реализована модель игрового агента для шутера в трехмерном лабиринте. Модель позволяет имитировать передвижение человека в лабиринте на основе собранных данных с реальных игроков; заменяет алгоритм автоматического распознавания противника на подключаемый внешний модуль распознавания противников и включенный в модель недетерминированный алгоритм прогнозирования положения противника относительно опасных зон. Передвижение реализованного БОТа совмещает обе указанные выше особенности.

Прицеливание в игре адаптировано по отношению к качеству распознавания противника и моторным реакциям человека, что учитывается как в скорости реагирования на распознавания, так и в траектории прицеливания, которая определяется в зависимости от положения противника и направления прицела игрока и разделяется на два основных этапа – наведение и коррекцию.

Модуль принятия игровых решений на текущем этапе практически целиком состоит из детерминированных правил для осуществления переходов между основными игровыми состояниями – передвижение, поиск, стрельба – и задается выборами обрабатываемой ветви дерева решений. Из-за этого компьютерный агент не всегда принимает оптимальное решение относительно дальнейших действий.

Модель подключается к локальному серверу с тестовыми уровнями, на которых и собираются данные с реальных игроков. Требуют доработки алгоритмы обучения по собранным данным, улучшение модели взаимодействия контроллеров, а также реализацией алгоритмов принятия решений с большей свободой при переключении между основными состояниями.

Список литературы

1. *Тьюринг, А. М.* Вычислительные машины и разум, В сб.: Хофштадер Д., Деннет Д. Глаз разума. — Самара: Бахрах-М, 2003. — с. 47-59
2. *Voronoi, G.* Nouvelles applications des parametres continus a la theorie des formes quadratiques. Journal fur die reine und angewandte Mathematik, 1907 — с. 133
3. *Loyall, A.* Believable agents: building interactive personalities. — Ph.D. Dissertation: Stanford University, 1997 — с. 2-3
4. Официальная документация Unreal Engine 4.
<https://docs.unrealengine.com/latest/INT/Engine/Blueprints/>
5. Разработчики пакета Mixiamo. <https://www.mixamo.com/3d-characters>
6. Рецензия на игру Call of Duty: Ghosts. http://www.thg.ru/game/obzor_call_of_duty_ghost_test/onepage.html
7. Рецензия на игру F.E.A.R.2. http://bestgamer.ru/reviews/project_origin.htm
8. Рецензия на игру Men of Valor. http://www.igromania.ru/articles/44798/Rukovodstvo_i_prohozhdenie_po_Men_of_Vvalor.htm
9. Сайт конференции AIST. <http://aistconf.org/2015/extra/ru>
10. Сборник статей Believable Bots. <http://www.springer.com/gp/book/9783642323225>
11. 2K BotPrize Competition <http://botprize.org/index.html>
12. Game/AI conference <http://gameaiconf.com/>

Метод автоматического построения UML-диаграмм последовательности с задаваемым уровнем детализации на основе журналов событий

Давыдова Ксения Валерьевна

факультет компьютерных наук департамент программной инженерии, группа
БПИ132
kvdavydova@edu.hse.ru

Ключевые слова: журнал событий, UML-диаграмма последовательно-
сти, автоматическое выведение моделей.

1 Введение

Разработка программного обеспечения (ПО) является быстро развивающейся областью, в которой задействовано много участников и ресурсов. Размеры и сложность ПО делают актуальным вопрос корректности и достоверности ПО по сравнению с теми спецификациями, по которым они разрабатываются. Кроме того, так называемые гибкие (agile) методологии зачастую вообще избегают создания классических моделей ПО. Вне зависимости от того, по какой методологии ведется разработка ПО, автоматическое выведение моделей (reverse engineering) является актуальной задачей. Для каскадной и итеративной методологий важен вопрос, действительно ли разработанная система соответствует первоначальным моделям. В гибких методологиях разработки моделей обычно нет изначально, однако было бы полезно получить эти модели на основе исполнения кода готовой программы и посмотреть на них с позиции архитектора.

Большое число работ, связанных с автоматическим выведением моделей в виде UML-диаграмм последовательности, заключается в анализе исходного кода. На их основе разработано много инструментов. Однако такой подход обладает недостатками: не всегда имеется доступ к исходному коду; если рассматривается сервис-ориентированная система, то ее компоненты могут быть гетерогенными и т.д. Тем не менее, почти все информационные системы имеют возможность записывать результаты своей работы в журналы событий. Было бы хорошо иметь инструмент, который на основе журналов событий формировал бы UML-диаграмму последовательности. Предлагаемый подход заключается в задании набора параметров, который позволяет получать диаграммы с разными уровнями детализации, что позволяет избегать «спагетти-моделей» (при выборе более высокого уровня абстракции или используя подходы для объединения некоторых элементов диаграммы), повышает читаемость диаграммы, учитывает декомпозиционную составляющую ПО.

2 Входные данные

Таблица 1. Фрагмент журнала событий L1. SOA-система банкинга

CaseID	Domain	Service	Operation	Action	Payload	Timestamp
23	Account	Operations	GetLastOperations	REQ	user=a, today=23.07.2015, client=Alex, manager=Julia	17:32:15 135
23	Account	CardInfo	GetCardID	REQ	user=a	17:32:15 250
23	Account	CardInfo	GetCardID	RES	res=15674839	17:32:15 297
23	Card	Operations	GetOperations	REQ	days=30	17:32:15 378
23	Utils	Calendar	GetDate	REQ	days=30	17:32:15 409
23	Utils	Calendar	GetDate	RES	res=23.06.2015	17:32:15 478
23	Card	Operations	GetOperations	RES	res={BP Billing Transfer, Retail}	17:32:15 513
23	Card	OperationData	GetPlaceAndDate	REQ	op=BP Billing Transfer	17:32:15 589
23	Card	OperationData	GetPlaceAndDate	RES	res=RUS SBERBANK ONLAIN PLATEZH, date=20.07.2015	17:32:15 601
23	Card	OperationData	GetPlaceAndDate	REQ	op=Retail	17:32:15 638
23	Card	OperationData	GetPlaceAndDate	RES	res=RUS MOSCOW OAO MTS, date=05.07.2015	17:32:15 735
23	Account	Operations	GetLastOperations	RES	res=succ	17:32:15 822
25	Account	Operations	GetLastOperations	REQ	user=a, today=23.07.2015, client=Alex, manager=Julia	17:40:18 345
25	Account	CardInfo	GetCardID	REQ	user=k	17:40:18 408
25	Account	CardInfo	GetCardID	RES	res=error	17:40:18 489
25	Account	Operations	GetLastOperations	RES	res=no bounded cards	17:40:18 523

Основными входными данными для рассматриваемого подхода и реализующего его инструмента является журнал событий, представленный в одном из стандартных форматов. Журнал событий, который рассматривается в работе, представляет собой множество трасс исполнения, где каждая трасса - это последовательность упорядоченных по времени событий. Формальное определение журнала события приведено ниже.

Определение 1. *Журнал событий. Пусть E - множество событий. Событие - это набор атрибутов $e = (a_1, a_2, \dots, a_n)$, где n - количество атрибутов. $\sigma = \langle e_1, e_2, \dots, e_k \rangle$ - это трасса (т.е. набор событий, относящихся к одному $CaseId$). $L = P(T)$ - журнал событий (т.е. мульти-множество трасс).*

Пример фрагмента журнала события для системы с сервис-ориентированной архитектурой (SOA) представлен в таблице 1. Здесь столбцы соответствуют атрибутам, а строки - событиям. Среди всех атрибутов наиболее важными являются CaseID, задающий трассу, Timestamp, задающий порядок событий, и Activity, который зависит от заданного уровня абстракции. Рассматривая журнал L1, в качестве атрибута Activity может быть выбран атрибут Domain, Service или Operation. Здесь сервисы, которые состоят из операций, объединены в домены. В работе рассматривались журналы событий,

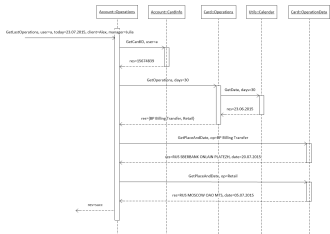


Рис. 1. Первый подход: отображение атрибутов журнала событий на компоненты диаграммы

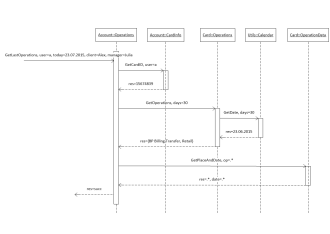


Рис. 2. Второй подход: объединение компонентов диаграммы на основе регулярных выражений

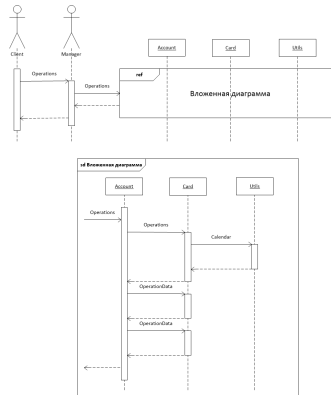


Рис. 3. Третий подход: декомпозиция диаграмм с использованием вложенных фрагментов

сформированные системами с SOA-архитектурой, предполагается, что новый вызов исходит из последнего не закончившегося вызова, кроме первого.

3 Ожидаемые результаты

Взаимодействие между компонентами программной системы может быть представлено с разными уровнями детализации. Возможность определения уровня детализации пользователем позволяет в зависимости от текущих требований получать как высокоабстрактные, так и детализированные на отдельных участках модели.

В работе предлагается три независимых подхода, в результате применения которых можно получить диаграммы с разными уровнями абстракции. На рисунках 1, 2 и 3 изображены диаграммы, полученные в результате применения каждого из подходов к журналу событий L1.

4 Предлагаемые подходы

Первый подход состоит в том, что можно выбрать, какие атрибуты журнала событий пользователь хочет видеть в виде объектов линий жизни (Lifelines), а какие в виде параметров сообщений на UML-диаграмме последовательности. В зависимости от выбора атрибутов, будут формироваться диаграммы разной степени детализации. Для данного подхода вводится функция, которая отображает множество атрибутов журнала событий на множество объектов линий жизни. На рисунке 1 значения атрибутов Domain и Service отображены на композитные объекты линии жизни с заголовками вида «Domain::Service». Также вводится функция, которая отображает другие атрибуты журнала событий на параметры сообщений (Messages). На рисунке 1 значения атрибутов Operation и Payload отображаются на параметры сообщений.

Второй подход заключается в том, что схожие участки диаграммы могут быть объединены при помощи регулярных выражений. Это позволяет существенно сократить размер модели за счет обобщения схожих поведений. В действительности, различные параметры считаем неразличимыми при регулярного выражения. Для данного подхода вводится функция «CheckInclusionCriteria», выполняющая отображение отфильтрованного декартова произведения журнала событий на самого себя [8] на множество 0,1 так, чтобы элемент произведения представлял двойку «текущее событие - следующие событие». Тогда двойка, удовлетворяющая критериям регулярного выражения помечается как включаемая (1), а неудовлетворяющая - как исключаемая (0). Наконец, вводится функция, которая отображает атрибуты журнала событий, прошедшие процесс объединения, на параметры сообщений UML диаграммы последовательности.

Третий подход основан на возможности строить вложенные диаграммы. Он позволяет иметь и высокоуровневую, и детализированную диаграммы. На рисунке 3 на первой диаграмме изображаются внешние для рассматриваемой системы участники, а на вложенной диаграмме отображается взаимодействие элементов системы. В зависимости от выбора атрибутов для отображения на диаграммах получаются диаграммы с разными уровнями детализации. Для данного подхода применяется функция «CheckInclusionCriteria» из предыдущего подхода для определения событий журнала событий, которые можно объединить, если считать некоторые атрибуты неразличимыми. Также применяется функция, которая выполняет отображение одной UML-диаграммы последовательности на другую. Данное отображение осуществляется при помощи блока *ref*, который скрывает детали на диаграмме высокого уровня и позволяет обратиться к ним на соответствующей вложенной диаграмме последовательности.

5 Апробация

В качестве апробации предложенных подходов была разработана библиотека журналов событий на языке C#, в которой предлагаются различные

компоненты, отображающие журнал событий на объектную модель журнала событий. Разные компоненты журнала событий показывают разную продуктивность. Если события упорядочены, можно применять упорядоченную коллекцию для представления трассы, если в журнале есть чередующиеся события, то эффективнее применять неупорядоченные коллекции. Для некоторых задач требуются все атрибуты журнала событий, и тогда необходимо предоставлять доступ ко всем из них, в других необходим доступ только к некоторым атрибутам (например, в нашем случае). Библиотека содержит компоненты, реализующие разные «политики», которые приведены в таблице 2. Также разработан плагин к фреймворку VTMine4Visio, в котором реализованы два первых подхода к автоматическому построению UML диаграмм последовательности.

Таблица 2. «Политики» для библиотеки журналов событий

	Признаки
Сортировка	Кеширование атрибутов
Трасса сортирована	Все атрибуты кешируются
Трасса не сортирована	Некоторые атрибуты кешируются
	Никакие атрибуты не кешируются

Для использования плагина автоматического построения UML-диаграмм последовательности необходимо указать файл журнала событий. Далее для применения первого подхода следует указать атрибуты журнала событий, которые отобразятся в виде линий жизни, и атрибуты, которые отобразятся в виде параметров сообщений. Этим задается уровень абстракции, который необходим пользователю. Для применения второго подхода необходимо указать атрибуты, как и в первом подходе, а также указать атрибуты, которые можно объединить при помощи регулярных выражений. После выбора атрибутов программа строит диаграммы последовательности заданного уровня абстракции в виде документа Microsoft Visio 2010. Фреймворк VTMine4Visio является плагином к Microsoft Visio 2010, так как последний является профессиональным инструментом для создания диаграмм и схем. Также, большим его преимуществом является поддержка языка UML.

Список литературы

1. W.M.P. van der Aalst Discovery, Conformance and Enhancement of Business Processes. - Springer, 2011.
2. OMG Unified Modeling Language (OMG UML) Version 2.5 // OMG: Object Management Group URL: <http://www.omg.org/spec/UML/2.5/PDF> (дата обращения: 22.02.2016).
3. S. Roubtsov, A. Serebrenik, A. Mazoyer, M.G.J. van den Brand, E. Roubtsova I2SD: Reverse Engineering Sequence Diagrams from Enterprise Java Beans with Interceptors.

4. T. Ziadi, M. Aurélio Almeida da Silva, L. Messan Hillah, M. Ziane A Fully Dynamic Approach to the Reverse Engineering of UML Sequence Diagrams.
5. R. Delamare, B. Baudry, Y. Le Traon Reverse-engineering of UML 2.0 Sequence Diagrams from Execution Traces. - 2006.
6. L.C. Briand, Y. Labiche, Y. Miao Towards the Reverse Engineering of UML Sequence Diagrams. - 2003.
7. Y. Guéhéneuc, T. Ziadi Automated Reverse-engineering of UML v2.0 Dynamic Models.
8. S. A. Shershakov. VTMine Framework as Applied to Process Mining Modeling. // International Journal of Computer Communication Engineering, 2015.

Визуализация динамических графов референций

Иван Родин

Факультет компьютерных наук, прикладная математика и информатика,
БПМ123
ivrodin@edu.hse.ru,

Ключевые слова: анализ текстов, визуализация динамических графов

Аннотация

Мы представляем инструмент для визуализации динамического графа референций. Динамический граф референций представляет собой граф, построенный на ключевых фразах, полученных из размеченного по времени корпуса естественного языка. Этот инструмент может быть использован для анализа связанных пар скрытых тем, изменений в значении этих тем и отношения между ними в течение различных периодов времени.

Введение

Визуализация текстов – довольно неоднозначное понятие. Один из подходов к пониманию это термина состоит в генерации сцен по текстовому описанию, как например в работе [1]. Однако для нас интерес представляет графическое изображение значимых элементов текста, таких как ключевые фразы или термины. Полученные схемы могут стать инструментом краткого изложения и суммаризации информации. Как показывает исследование [2] число техник для анализа и визуализации стремительно возрастает.

В сущности, проблема визуализации данных может быть разложена на две составляющие: визуализация статической и динамической текстовой информации. Наиболее известный способ графического представления статических текстовых данных это облако тэгов [3]. Многие техники визуализации расширяют эту идею. Например, [4] и [5].

Методы, основанные на составлении облака тэгов, также весьма успешно применяются при визуализации динамических текстовых данных. К примеру, визуализация ThemeRiver, предложенная в работе [6]. В исследовании [7] представлена ещё одна оригинальная техника визуализации динамических текстовых корпусов, называемая «схемой метро».

Графы тэгов — это ещё одно расширение концепции облака тэгов. Чтобы составить граф тэгов, необходимо определить своего рода отношение между тэгами. Например, в работе [8], тэги представляют вершины графа, и они соединены ребром, если они встречаются вместе. В [9] представлен способ

графического представления для динамических графов совместной встречаемости.

В данной работе мы предлагаем подход к анализу временных текстовых данных, основанный на графе референций. Ключевое отличие такого графа от графа совместной встречаемости состоит в его ориентированности. Это позволяет анализировать, как один термин «ссылается» на другой, и более подробно описывать паттерны отношений между терминами. Наш метод визуализации также позволяет анализировать изменение значимости конкретных терминов во времени.

Основная часть

Чтобы построить динамический граф референций, можно использовать Алгоритм 1.

Algorithm 1: Построение динамического графа референций

Input : Индексированный по времени корпус текстовых документов и список ключевых фраз

Output: Динамический граф референций

- 1 Разбить корпус на T последовательных временных периодов $\tau = 1..T$.
- 2 Задать список N ключевых фраз (концептов) $w_i, i = 1..N$.
- 3 **for** $\tau = 1..T$ **do**
- 4 Извлечь информацию о статических взвешенных ориентированных графах G_τ , которая должна включать: 1) значимость (значение величины поддержки) каждого концепта в период: $S_\tau(w_i)$, или число документов, в которых появился данный концепт; 2) Значения уровней доверия связей $C_\tau(w_i, w_j), C_\tau(w_j, w_i)$ для всех пар концептов w_i, w_j .
- 5 **end**
- 6 Определить нижние границы для значений support и confidence.
Визуализировать граф G_1 for $\tau = 1$ с учетом следующих ограничений:
- 7 **for** $\tau = 2..T$ **do**
- 8 Удалить ребра из $G_{\tau-1}$ которые не встречаются в G_τ
- 9 Удалить вершины из $G_{\tau-1}$ которые не встречаются в G_τ
- 10 Добавить вершины из G_τ которые не встречаются в $G_{\tau-1}$
- 11 Добавить ребра из G_τ которые не встречаются в $G_{\tau-1}$
- 12 **end**

Инициализация. Входными данными для визуализации является корпус, размеченный по времени. Необходимо разбить документы на несколько временных периодов, предполагая, что число документов в каждом периоде сопоставимо. Далее нужно задать список концептов. Они могут быть определены как самые частотные термины из корпуса. Для некоторых случаев имеет смысл задать список концептов вручную.

Построение G_τ . Концепты определяют множество вершин графа G_τ . Взвешенные ориентированные ребра задаются мерой смежности этих концептов. При построении графа так же учитываются информация о поддержке каждого из концептов во все временные периоды.

Вычисление поддержки концепта. Определим значимость концепта w_i во временной период τ как $S_\tau(w_i) = |D_\tau(w_i)|$, где $D_\tau(w_i)$ множество документов в период τ , в которых концепт w_i встречается больше, чем σ раз. По умолчанию мы используем для построения значение $\sigma = 0$, однако в некоторых случаях порог может быть увеличен.

Вычисление уровня надежности. Если известны значения уровней поддержки для всех концептов w_i в τ , мы можем вычислить значения уровней надежности для отсылок между всеми парами концептов с ненулевыми значениями поддержки. Мы полагаем, что концепт w_i “отсылает” к концепту w_j во временной период τ с уровнем надежности $C_\tau(w_i, w_j)$, если $C_\tau(w_i, w_j) = \frac{|D_\tau(w_i) \cap D_\tau(w_j)|}{|D_\tau(w_i)|} \in [0; 1]$.

Как только все данные о концептах и связях между ними за период τ получены, мы можем приступить к построению статического ориентированного взвешенного графа G_τ , для этого периода. Вершина из концепта w_i к концепту w_j с весом $C_\tau(w_i, w_j)$ в графе означает, что w_i отсылает к w_j с уровнем надежности $C_\tau(w_i, w_j)$. Стоит отметить, что значения $C_\tau(w_i, w_j)$ близкие к 1 означают, что если w_i встречается в документе, то и w_j с высокой долей вероятности должно в нем встретиться. Иными словами, w_j обычно появляется в контексте w_i .

Данные

Мы собрали коллекцию статей на тему экономики из четырех Российских новостных порталов (“Известия”, “Коммерсант”, “Московский Комсомолец”, “Независимая Газета”), опубликованных в 2014 году. Этот корпус (далее “RuNeWC”) содержит 4061 статью. Он разбит на 26 временных периодов, каждый из которых имеет длину 2 недели.

Концепты извлекались автоматически, с использованием стратегии, предложенной в [10], которая состоит из 2 шагов. На первом шаге извлекаются потенциальные претенденты — слова и словосочетания, которые удовлетворяют заданным речевым паттернам, предложенным в [11] (например, ПРИЛ.+СУЩ., СУЩ.+СУЩ.). На следующем шаге из 250 самых частотных фраз и 100 слов формируется список концептов. На финальном этапе вручную удаляются концепты, которые не являются семантически значимыми (прим.: корреспондент Известий). Для визуализации графа мы установили минимальный уровень поддержки равный 28 и уровень уверенности 0.9. Пример визуализации динамического графа для корпуса RuNeWC представлен на рис. 1

Чтобы представить еще более понятный пример, мы создали корпус, основанный на четырех книгах из цикла популярного фэнтези-романа “Песнь льда и пламени” написанного Джорджем Мартиным. В качестве концептов мы задали список персонажей саги.

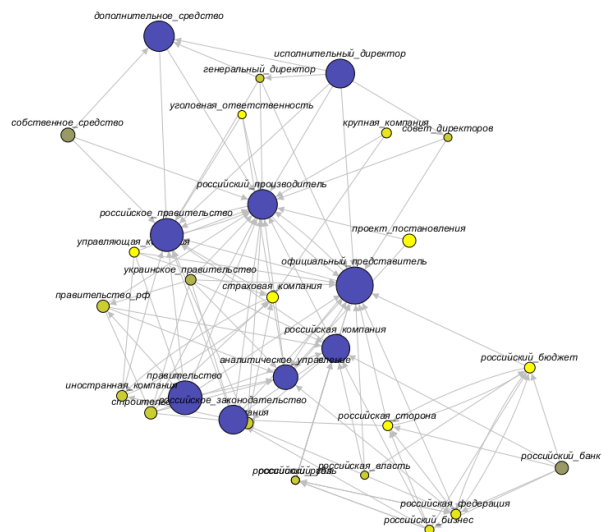


Рис. 1. Граф референций для RuNeWC за период $\tau = 9$ (17-18 недели 2014)

Сюжет в первых четырех книгах развивается последовательно. Мы разбили каждую главу на несколько равных частей (2–7, в зависимости от размеров главы). Каждая из этих частей была представлена в корпусе как отдельный документ. Далее множество всех документов было разбито на группы из 50 последовательных документов, которые сформировали 14 временных периодов. Мы не рассматривали каждую главу как отдельный временной период, так как время между действующими персонажами одной главы распределено неравномерно. Деление одной главы на несколько документов произведено для того, чтобы увеличить точность оценок.

В нашем эксперименте мы считаем, что концепт встречается в документе, если он появился в нем хотя бы два раза (т.е. $\sigma = 1$). Граница для уровня поддержки была взята равной 5, а минимальное значение уровня надежности для связи между концептами 0.3 (определяет, будет ли отображаться ребро в графе).

В результате мы получили визуализацию динамического графа референций, которая может служить схематическим пересказом сюжета “Песни льда и пламени”. Примеры визуализации для корпуса ASOIAF представлены на Рис. 2,3,4,5¹.

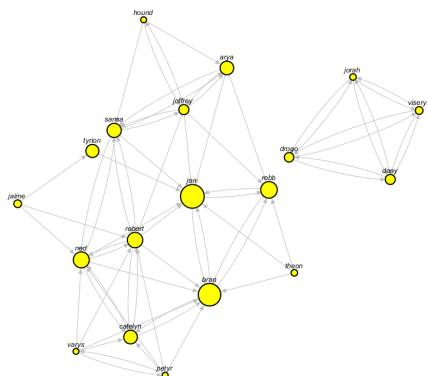


Рис. 2. Референсный граф для корпуса ASOIAF в первый временной период. Примечательно, что граф состоит из двух связанных компонент, соответствующих двум континентам, на которых разворачивается действие книг.

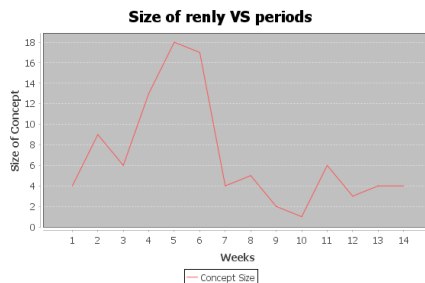


Рис. 3. Изменение поддержки концепта *Renly* с течением времени.

Визуализация

Существует несколько средств для визуализации динамических графов, таких как GraphStream², KeyLines³, Gephi⁴. Для нашего задания мы использовали GraphStream, так как это хорошо документированная библиотека для Java, которая была специально разработана для редактирования и визуализации динамических графов.

Наше ПО позволяет визуализировать динамические графы референций основываясь на их текстовом описании. Оно обеспечивает зависимость укладки графа референций G_τ для $\tau > 1$ от укладки графа $G_{\tau-1}$. Наша визуализация также учитывает информацию о поддержке концептов: чем больше поддержка, тем соответствующая вершина графа. “Возраст” вершины (число периодов, в которых концепт появлялся без перерывов) также учитывается: новые вершины графа окрашены желтым цветом, но с течением времени их цвет плавно переходит в синий. Если концепт появляется в графе впервые, соответствующая вершина получает широкую черную границу.

Наша визуализация также имеет графический пользовательский интерфейс. В любой момент времени пользователь может приостановить показ анимации чтобы исследовать структуру графа. Все вершины показываемого графа можно перемещать мышкой. Наше ПО также обрабатывает клики: если пользователь кликнул по вершине графа, только вершины и ребра

¹ Видео с визуализацией доступно по ссылке <https://youtu.be/UaUGVPTdM-w>

² <http://graphstream-project.org/>

³ <http://cambridge-intelligence.com/keylines/>

⁴ <https://gephi.org/>

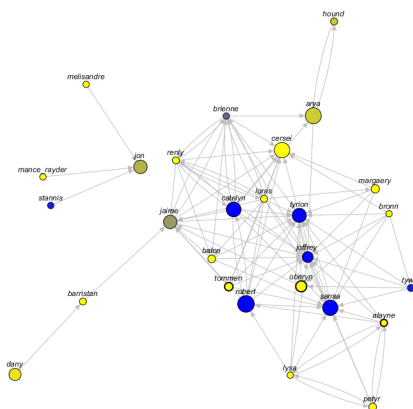


Рис. 4. Динамический граф в период $\tau = 11$, который соответствует действиям, описанным в конце третьей книги. Цветовая дифференциация вершин отображает возраст концептов в графе.

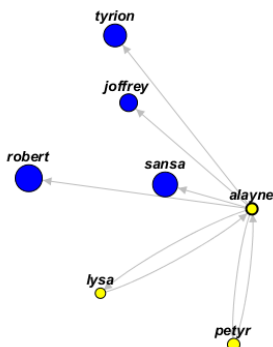


Рис. 5. Если пользователь выбрал конкретную вершину, то отображаются только смежные с ней

смежные с ней отображаются. Более того, информация о динамике изменения поддержки соответствующего концепта за все временные периоды отображается в отдельном окне.

Выводы

Метод визуализации текстовых корпусов с использованием графа референций представляется нам довольно простым, понятным, и в то же время эффективным. Предложенная визуализация позволяет анализировать такие ключевые показатели как: самые частотные термины, связи между ними, изменение частоты появления концептов во времени, изменение отношений между концептами, “возраст” концептов. При этом сам процесс визуализации контролируется пользователем.

Стоит отметить дальнейшие направления работы:

Мы собираемся интегрировать ПО по визуализации с ПО, извлекающим концепты и информацию о связях между ними из корпуса, это позволит пользователям легко строить графы референций для их собственных корпусов. Система поиска концептов также может быть улучшена: один из возможных вариантов улучшения системы — использовать метод аннотированного суффиксного дерева [12]. Представленная система может быть улучшена путем добавления системы, извлекающей контекстные синонимы, такие как *Петир – Лорд Бейлиш – Мизинец* в корпусе ASOIAF. Исследование динамических графов может оказаться эффективным инструментом для аналитики. Извлечение и выделение сильно связанных компонент позволит пользователям распознавать значимые регулярные тренды.

Благодарности. Автор выражает благодарность участникам научно-учебной группы “Методы анализа и визуализации веб-корпусов” за множество ценных комментариев к работе, а также руководителю группы Миркину Борису Григорьевичу за неоценимый вклад в создание представленной системы.

Список литературы

1. *Chang A. X., Savva M., Manning C. D.* Semantic parsing for text to 3d scene generation //ACL 2014. – 2014. – С. 17.
2. *Kucher K., Kerren A.* Text visualization browser: A visual survey of text visualization techniques //Poster Abstracts of IEEE VIS. – 2014. – Т. 2014.
3. *D. Coupland.* 1996. Microserfs, Flamingo.
4. *Coppersmith G., Kelly E.* Dynamic wordclouds and vennclouds for exploratory data analysis //Sponsor: Idibon. – 2014. – С. 22.
5. *Wang H. et al.* A system for real-time twitter sentiment analysis of 2012 us presidential election cycle //Proceedings of the ACL 2012 System Demonstrations. – Association for Computational Linguistics, 2012. – С. 115-120.
6. *Havre S., Hetzler B., Nowell L.* ThemeRiver: Visualizing theme changes over time //Information Visualization, 2000. InfoVis 2000. IEEE Symposium on. – IEEE, 2000. – С. 115-123.
7. *Shahaf D. et al.* Information cartography: creating zoomable, large-scale maps of information //Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. – ACM, 2013. – С. 1097-1105.
8. *Lloyd L., Kechagias D., Skiena S.* Lydia: A system for large-scale news analysis //String Processing and Information Retrieval. – Springer Berlin Heidelberg, 2005. – С. 161-166.
9. *Berendt B. and Subasic. I.* Stories in time: A graphbased interface for news tracking and discovery. — In WI-IAT09, 2009.
10. *Hulth A.* Improved automatic keyword extraction given more linguistic knowledge //Proceedings of the 2003 conference on Empirical methods in natural language processing. – Association for Computational Linguistics, 2003. – С. 216-223.
11. *Mitrofanova O., Zakharov V.* Automatic Analysis of Terminology in the Russian Corpus on Corpus Linguistics //NLP, Corpus Linguistics, Corpus Based Grammar Research. – 2009. – С. 249.
12. *Dubov M.* Text Analysis with Enhanced Annotated Suffix Trees: Algorithms and Implementation //Analysis of Images, Social Networks and Texts. – Springer International Publishing, 2015. – С. 308-319.

Визуальное распознавание противника в трехмерном лабиринте на Unreal Engine 4

Токмакова Лада Романовна¹

образовательная программа "Математические методы оптимизации и стохастики"

lrtokmakova@yandex.ru

Ключевые слова: визуальное распознавание образов, обнаружение прямых линий, преобразование Хафа

1 Расширенная аннотация

Существует множество алгоритмов обнаружения противников для игрового искусственного интеллекта (ИИ) в трехмерных шутерах. В современных компьютерных играх в большинстве случаев используются методы, основанные на автоматическом распознавании цели, например, рейкастинг (ray casting) [7,12]. Используемые в играх запрограммированные сценарии (scripting) [17], которые активируются, когда игрок находится в соответствующей зоне, также вносят искусственную составляющую в игровой процесс. В таких моделях игроки-компьютеры заранее находятся в выигрышной ситуации по сравнению с игроками-людьми, потому что ИИ не требуется тратить время на обнаружение противника или процесс прицеливания.

Реализация противников с имитацией поведения человека в коммерческих шутерах редкое, однако, встречающееся явление в современной игровой индустрии [11, 15, 16], что подтверждает актуальность данных исследований.

Разработчики решают проблему мгновенной реакции ИИ на игрока-человека в определенной локации следующим образом: при срабатывании рейкастинга они устанавливают некоторую задержку на прицеливание или обнаружение. Такое решение не является реалистичным, потому что человек не всегда может увидеть противника в случаях, когда из укрытия видна только часть его тела, а в данной ситуации обнаружение гарантировано. В связи с этим было проведено исследование, основная цель которого заключается в том, чтобы смоделировать поведение игрока, управляемого компьютером, приближенное к человеческому.

В данной работе описывается реализация визуального распознавания для игрового агента. *Объектом* исследования является визуальный ряд, получаемый в игре от первого лица – скриншоты трехмерного лабиринта, которые отображают то, что видит бот в игре. Стоит отметить, что заранее известны типы объектов на игровой карте: коробки, стены, колонны и дверные проемы. Эти знания помогают сформулировать предположение, что многие контуры на карте, кроме противников, состоят из прямых линий.

Таким образом, *предметом* исследования является наличие или отсутствие кластеров на изображении, соответствующих возможному расположению игроков в лабиринте, в том числе, частично закрытых контурами с границей из прямых линий.

За последние десятилетия появилось множество методов визуального распознавания [10, 13], часть из которых может быть применена для решения задачи поиска противника на скриншотах. *Методами* исследования являются методы компьютерной обработки изображений. Реализация анализа объектов исследования осуществляется с помощью инструментов Image Processing Toolbox для среды программирования Matlab. *Цель* работы заключается в том, чтобы построить модель зависимости времени реакции на появление противника относительно оценки качества распознавания областей на скриншоте, которые могут соответствовать найденным противникам, с учетом знания геометрии трехмерного лабиринта и протестировать полученную модель. Для обработки объектов исследования были решены следующие *задачи*:

- Реализация метода для распознавания противников на скриншоте трехмерного лабиринта;
- Оценка полученных результатов распознавания числовым параметром;
- На основе оценки результата выявление соответствия между качеством распознавания и временной задержкой перед проверкой объекта средствами движка Unreal Engine 4 [18, 19].

Результатом работы является контроллер обнаружения противников для проекта в Unreal Engine 4, содержащий исполняемый файл программы для распознавания, написанной на языке Matlab.

Алгоритм модуля распознавания можно разбить на несколько основных этапов:

- Выделение контуров на изображении;
- Фильтрация полученного изображения;
- Поиск на изображении прямых линий (с помощью преобразования Хафа);
- Исключение объектов, образованных прямыми линиями;
- Поиск центров найденных связных областей.

Точность обнаружения вычисляется при сравнении полученного результата модуля распознавания с эталонами изображения противника (вид бота с разных сторон).

Эксперимент показывает, что в выборке из 150 скриншотов ошибка первого рода равна 30,4%, а ошибка второго рода равна 5,33%. Нам удастся обнаруживать противника за время, меньшее скорости человеческой реакции на распознавание противника, но при этом качество распознавания относительно ошибок второго рода требует доработки.

В рамках продолжения работы планируется усовершенствовать реализованный алгоритм с целью достижения более высокой скорости и улучшения качества работы модуля распознавания.

Список литературы

1. *Говорухин, В. Н., Цибулин, В. Г.* Введение в Maple. Математический пакет для всех — М.: Мир, 1997
2. *Гонсалес, Р., Вудс, Р., Эддинс, С.* Цифровая обработка изображений — М.: ТЕХНОСФЕРА, 2006
3. *Прэтт, У.* Цифровая обработка изображений — М.: Мир, 1982
4. *Шапиро, Л., Стокман, Дж.* Компьютерное зрение — М.: БИНОМ. Лаборатория знаний, 2006
5. *Яне, Б.* Цифровая обработка изображений — М.: ТЕХНОСФЕРА, 2007
6. *Canny, J.* A Computational Approach to Edge Detection — IEEE Transaction on Pattern Analysis and Machine Intelligence, 1986
7. *Goldstein, R. A., Nagel, R.* 3-D visual simulation
8. *Lim, Jae, S.* Two-Dimensional Signal and Image Processing — Englewood Cliffs. New Jersey: Prentice Hall, 1990
9. *Parker, James, R.* Algorithms for Image Processing and Computer Vision — New York: John Wiley & Sons, 1997
10. *Szeliski, R.* Computer Vision: Algorithms and Applications — Springer London Dordrecht Heidelberg New York, 2011
11. *Янович, И.* Обзор на Call of Duty: Ghosts <http://kanobu.ru/articles/syuzhetnaya-kampaniya-call-of-duty-ghosts-367451/>
12. <http://embor.embopress.org/content/embor/7/5/467.full.pdf>
13. <http://graphicon.ru/ru/conference>
14. <http://aistconf.org/2015/ru>
15. <http://splintergame.ru/load/7-1-0-6>
16. <https://www.callofduty.com/content/dam/atvi/callofduty/ghosts/manuals/Ghosts-Manual-PC-en.pdf>
17. <http://web.stanford.edu/~ouster/cgi-bin/papers/scripting.pdf>
18. <https://www.unrealengine.com>
19. <http://unreal-engine4.ru>
20. <http://matlab.exponenta.ru/imageprocess/book3/index.php>

Преобразование каузальных сетей в бесконфликтные сети потоков работ

Т. Р. Бадретдинов

Национальный исследовательский университет «Высшая школа экономики»
Программная инженерия, 2 курс
trbadretdinov@edu.hse.ru

Ключевые слова: Process Mining, Causal nets, Workflow nets

1 Введение. Постановка задачи

1.1 Мотивация

В сфере анализа процессов уже устоялись некоторые традиции [1]. Проверенные сообществом модели представления бизнес-процессов, такие как сети Петри, пользуются большой популярностью у научного сообщества. Однако, у моделей подобного рода существует ряд недостатков. В этой работе мне хотелось бы сделать первую попытку в устранении недостатков стандартных структур с сохранением их достоинств.

1.2 Сети Петри. Сети потоков работ

Сети Петри (Petri nets) являются одним из традиционных способов представления процессов [2]. Сеть состоит из позиций, переходов и токенов. Состояние сети определяется её разметкой, то есть размещением токенов по позициям. При выполнении перехода (события) токены забираются из каждого входного места и передаются всем выходным местам.

Сети потоков работ (WF-nets, workflow nets) являются частным случаем сетей Петри, в котором есть только одно начальное и одно конечное места. Таким образом, в начале работы сети в ней есть только один токен, стоящий в начальном месте. В случае правильной работы по завершении процесса в сети должен остаться только один токен, находящийся в конечном месте.

Одним из существенных недостатков сетей потоков работ является возможность возникновения deadlock'a (тупика) во время её выполнения.

1.3 Каузальные сети

Каузальные сети (C-nets, Causal nets) являются относительно новой структурой представления процессов, предложенной Аалстом и другими учёными в 2011 году [3]. Особенностью этой структуры является декларативная семантика, обеспечивающая надёжность модели, в отличие от более традиционных моделей, например сетей Петри.

Существенными недостатками таких сетей являются относительная сложность конструкций и невозможность простым способом исполнить их.

1.4 Постановка задачи

Цель работы — предложить алгоритм, который объединит простоту и выполнимость сетей потоков работ и безопасность каузальных сетей. Более формально, алгоритм должен избавить сеть от deadlock’ов и переполнений конечной позиции путём наложения ограничений на выполнение некоторых переходов.

2 Решение задачи

2.1 Преобразование C-net в WF-net

Прежде всего необходимо получить сеть потоков работ, для которой требуется обеспечить гарантию корректного выполнения. Для получения Workflow-net из существующей Causal-net существует алгоритм [3].

2.2 Построение графа достижимости

Для дальнейшей работы требуется построить граф достижимости сети, то есть совокупность всех возможных маркировок (состояний) этой сети. За первую вершину возьмём начальную маркировку WF-сети. Далее для каждой вершины — начиная с первой — достраиваем граф, добавляя новые возможные разметки.

Algorithm 1 Рекурсивное построение графа достижимости по сети потоков работ.

```
1: procedure BUILDVERTEX(vertex, WFNet<P, T, F>, G<V, E>)
2:   for each t in T do
3:     if t can be activated then
4:       newMarking ← getNewMarking(vertex.getMarking(), t)
5:       create the vertex newVertex with marking newMarking
6:       create the edge edge from vertex to newVertex
7:       E.add(edge)
8:       if newVertex not in V then
9:         V.add(newVertex)
10:      buildVertex(newVertex, WFNet, G)
```

В приведённом выше алгоритме сеть потоков работ задана множеством *P* мест, множеством *T* переходов и множеством *F* дуг. Граф достижимости определён множеством вершин *V* и множеством *E* рёбер. Для получения графа требуется выполнить этот алгоритм для вершины, представляющей начальную маркировку.

2.3 Раскраска графа достижимости

Следующим шагом необходимо раскрасить полученный граф. Для этого потребуется два цвета: красный и зелёный. Вершина графа, раскрашенная в красный цвет, ведёт к неправильной конечной маркировке, противоречащей определению. Вершины графа, раскрашенные в зелёный цвет, ведут хотя бы к одному варианту хорошего завершения работы сети (завершение с корректной конечной маркировкой).

На рис. 1 представлена раскраска графа для простейшей сети потоков работ.

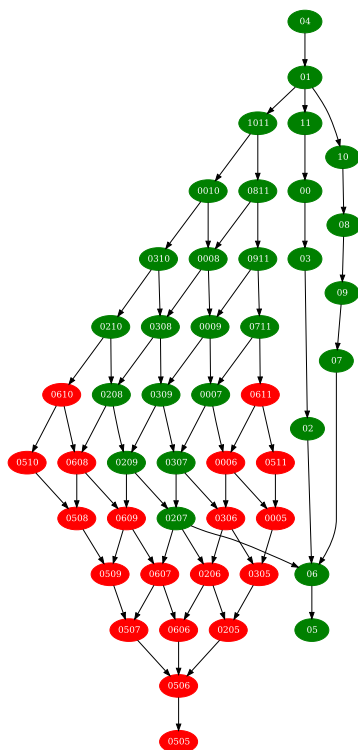


Рис. 1. Раскраска графа достижимости. В узлах указана текущая маркировка.

2.4 Поиск потенциально некорректных переходов

Для поиска ограничений необходимо сначала найти те переходы в сети, выполнение которых может привести к проблемам. Имея раскрашенный граф достижимости сделать это очень просто. С помощью поиска сверху вниз найдём такие переходы, в которых после зелёной вершины находится

красная, то есть те переходы, после выполнения которых не остаётся возможности корректно завершить работу сети. Ниже приведено формальное описание алгоритма.

Algorithm 2 Finds transitions that lead to invalid marking

```
1: procedure TRAVERSEVERTEX(Vertex, Transitions, MG<V, E, M>)
2:   mark Vertex as visited
3:   children ← Vertex.getChildren()
4:   for each child in children do
5:     if M[Vertex] ∧ not M[child] then
6:       transition ← MG.getEdge(Vertex, child)
7:       Transitions.add(transition)
8:     else
9:       traverseVertex(child, Transitions, MG)
```

В приведённом выше алгоритме раскрашенный граф достижимости задан множеством *V* вершин, множеством *E* рёбер и отображением *M* вершин на их раскраску (зелёная или красная). Для получения множества переходов требуется выполнить этот алгоритм для вершины, представляющей начальную маркировку.

3 Заключение

Полученные результаты можно применить для получения новых структур, которые учитывают найденные ограничения во время своего исполнения. Примером такой модели может стать сеть Петри с данными, в которой в качестве переменных будет храниться информация об активации переходов, а на некоторые переходы будут накладываться условия. Полученная сеть будет безопасной.

Список литературы

1. *van der Aalst, W.* Process Mining: Discovery, Conformance and Enhancement of Business Processes – Springer, 2011

2. *Reisig, W.* Understanding Petri Nets – Springer, 2013

3. *van der Aalst, W., Adriansyah A., and van Dongen B. F.* Causal Nets: A Modeling Language Tailored towards Process Discovery // CONCUR 2011 – Concurrency Theory. – Aachen, 2011. – P. 28–42.

Оценка объема рынка DIY в России

Кошелева Евгения¹, Петров Михаил², Бенькович Никита²,
Баранецкая Дарья², and Быков Николай²

¹ Науки о данных, м15

evkosheleva@edu.hse.ru,

² Прикладная математика и информатика 133,142,121,142

{mvpetrov_1,ndbenkovich,dabaranetskaya,nsbykov}@edu.hse.ru

Факультет Компьютерных Наук, ВШЭ, 2016

Ключевые слова: оценка объема рынка, прогнозирование объема рынка, регрессионный анализ, DIY

1 Аннотация

Проблема оценки рынка в экономике не нова и используется в разных исследованиях. В статье [1] это используется для оценки объема рынка техники по добыче промышленной и энергетической древесины по Ленинградской области, в [2] это является одним из параметров в исследовании сорока регионов Японии, в [3] оценка объема рынка влияет на выбор производителей и экспортеров на рынке. В нашей работе мы делаем первые шаги по изучению рынка DIY («Do it yourself», аббревиатура, используемая для обозначения сетевой розничной торговли товарами для ремонта и строительства), и первая задача стояла в оценке его объема, без каких-либо ограничений по использованию тех или иных данных. Данная работа выполнена в рамках проектно-учебной группы студентов ФКН НИУ ВШЭ, совместно с компанией «Леруа Мерлен», и посвящена одной из задач анализа данных в области маркетинга.

2 Введение

Основной проблемой «чистых вычислений» по чекам с продаж является следующий факт: из сегментов рынка, то есть гипермаркетов, специализированных магазинов, рынков и т. д., лишь гипермаркеты подают соответствующие статистические данные. Процент доли гипермаркетов от всего рынка продаж неоднозначен. По этой причине оценить весь объем рынка более точно представляет некоторые трудности. Подсчет оборота всех точек напрямую весьма дорогостоящее мероприятие, хотя нам и были предоставлены соответствующие данные по трем городам России. Путем прямого обхода адресов в них была собрана необходимая информация. Однако, перед нами стояла задача оценки рынка по всей России, и, очевидно, что в таких масштабах нужны более эффективные средства.

3 Первый этап

На первом этапе работы, изучив доступные данные, мы выработали базовый подход к решению данной задачи. Учитывая, что мы имеем дело с оценочными данными и прогнозированием, построение регрессии кажется самым логичным подходом. Идея заключается в построении алгоритма подсчета объема рынка каждого региона России по каким-то известным параметрам этого региона. Тем самым, взяв объем рынка в качестве целевого признака, задача сводится к построению обучающей и тестовой выборки и вычислению коэффициентов регрессии.

4 Второй этап

На следующем этапе была выработана стратегия построения обучающей выборки. Имея цифры объема рынка лишь в трех городах, использование объема в чистом виде, как значение целевого признака, наверняка бы привело к высокой погрешности. По запросу, нам были предоставлены данные опросников по 16 городам о средних тратах людей на ремонт в год. Также нам предоставили информацию о проценте домохозяйств, делающих ремонт в год, соответственно в квартире, на даче и в доме. Мы оценили объем рынка региона следующим образом:

- 1) численность региона разделенная на среднее число людей в домохозяйстве дает среднее число домохозяйств в регионе;
- 2) полученное число умножаем на количество домохозяйств делающих ремонт;
- 3) умножив ответ на средние траты домохозяйства на ремонт в год получим объем трат домохозяйств по региону.

5 Решение и погрешность

Таким образом, задача свелась к прогнозированию трат на ремонт по регионам, а уже по полученным данным, с помощью 1)-3), к подсчету объема рынка. В качестве параметров модели мы выбрали количество гипермаркетов, рынков, моллов, небольших магазинов, и специализированных магазинов, полученные из API 2ГИС.

Было предложено 4 гипотезы об оптимальном наборе параметров:

- 1) количество магазинов разного типа;
- 2) количество людей, приходящихся на каждую категорию торговых точек, на расходы на ремонт (численность/количество гипермаркетов);
- 3) количество денег, приходящихся на каждую категорию торговых точек, на расходы на ремонт (численность*среднюю зарплату/количество гипермаркетов);
- 4) площади жилищного фонда, приходящегося на каждую категорию торговых точек, на расходы на ремонт (площадь жилищного фонда/количество гипермаркетов);

Гипотезы 2)-4) не дали адекватного результата, так как многие значения регионов показали отрицательные значения. Это происходит из-за того что данные по магазинам представлены по городам, а впоследствии используется численность всего региона, что приводит к некорректности коэффициентов. За основу была взята гипотеза 1, к которой мы применили обычную линейную регрессию без регуляризации.

С помощью кроссвалидации была подсчитана средняя ошибка, которая составила 12%, а на обучении - 32%. Несмотря на столь высокие абсолютные значения погрешности и ошибки на обучении, в решении задач анализа рынков - это вполне приемлимые значения, так как существует теневой сегмент рынка (бабушки, продающие рассаду на обочинах, нелегальные предприятия), который в реализации модели не учитывается, но может по факту составлять довольно большой процент рынка.

На этапе реализации было выделено 82 субъекта РФ, и в каждом взято по 1-3 города с населением более 100 тыс. человек. Таким образом, коэффициенты регрессии, полученные на выборке из 16 городов, применялись при подсчете средних расходов на ремонт в других крупных городах РФ. Также стоит учесть, что при расчетах 1)-3) используется численность всего региона, а не конкретного города, в предположении, что закупки, так или иначе, производятся в более крупных городах региона.

В качестве основного результата, было получено ранжирование городов по объему рынка DIY в деньгах, которое соотносится с экспертным мнением специалистов, и, соответственно, общий объем рынка DIY в России, посчитанный с погрешностью приблизительно 20%.

6 Заключение

Стоит заметить, что для тринадцати субъектов РФ данных 2GIS по структуре рынка не существует, в них были использованы средние значения признака. На данный момент, модель полностью ориентирована на рынок B2C, и учитывает лишь ремонт жилых помещений. Дальнейшая работа будет направлена на адаптирование модели на рынок B2B.

Список литературы

1. *Gerasimov, Yur. Karjalainen, Timo.* Estimation of Machinery Market Size for Industrial and Energy Wood Harvesting in Leningrad Region // *Croatian Journal of Forest Engineering*, 2012, Vol. 33, №. 1, Str. 49-60
2. *Donald R. Davis, David E. Weinstein* Market Size, Linkages, and Productivity: A Study of Japanese Regions // *NBER Working Paper* №. 8518. 2001.
3. *Marc J. Melitz, Gianmarco I. P. Ottaviano* Market Size, Trade, and Productivity // *Review of Economic Studies*, 2007, Vol. 75, Issue. 1, P. 295-316.
4. Международная картографическая компания 2ГИС. [Электронный ресурс] <http://info.2gis.ru/moscow>
5. Федеральная служба государственной статистики. [Электронный ресурс] <http://www.gks.ru/>