



Исследование применимости специализации алгоритма Витерби скрытой марковской моделью

Автор: Иван Владимирович Тюляндин, 19.М07-мм

Руководитель: к.ф.-м.н., доцент С.В. Григорьев

Консультант: к.ф.-м.н., ст. преп. СПбГУ Д.А. Березун

Рецензент: ст. преп. СПбПУ М.Х. Ахин

Санкт-Петербургский Государственный Университет
Кафедра системного программирования

9 июня 2021

- Алгоритмы методами линейной алгебры (ЛА) на больших данных
- Любые улучшения критичны
 - ▶ оборудование
 - ▶ изменение алгоритма
- Часть данных может быть зафиксирована
- Применение специализации

Специализация

Два типа параметров

- статические — т.е. зафиксированные
- динамические — все остальные

Специализация

Техника преобразования программ для оптимизации использования статических данных с целью уменьшить количество вычислений

Применение специализации к алгоритмам, выраженными методами ЛА, не изучено

- применяется во многих областях
- выражается методами ЛА
- два параметра: скрытая марковская модель (СММ) и последовательность наблюдений
- на практике СММ зафиксирована, меняется только последовательность

Цель работы

Исследовать применимость специализации к алгоритму Витерби, который описан методами ЛА, при условии, что СММ является статическим параметром

- сделать обзор предметной области
 - ▶ рассмотреть алгоритм Витерби и его существующие реализации
 - ▶ описать технику специализации
- разработать специализированный алгоритм Витерби, реализовать и протестировать корректность реализации
- провести эксперименты по сравнению производительности специализированного алгоритма с неспециализированной версией и существующими реализациями

Скрытая марковская модель

Скрытая марковская модель

Вероятностный детерминированный автомат, каждое состояние которого создает наблюдение

- $S_{1..N}$ — N состояний
- $O_{1..K}$ — K возможных наблюдений
- $B_{1..N}$ — вероятности для состояний $S_{1..N}$ быть стартовыми
- $T_{1..N,1..N}$ — матрица переходов, $T_{i,j}$ — это вероятность перехода из S_i в S_j
- $E_{1..N,1..K}$ — матрица наблюдений, $E_{i,j}$ — это вероятность создать наблюдение O_j в состоянии S_i

СММ и последовательность наблюдений

Алгоритм Витерби, выраженный с помощью полукольца *Min-plus*

Для всех вероятностей p из СММ:

$$t(p) = \begin{cases} p > 0 : & -1 * \log_2(p) \\ p = 0 : & +\infty \end{cases}$$

Алгоритм Витерби методами ЛА

Для всех наблюдений o :

$$P(o) = \begin{pmatrix} t(E[1, o]) & \dots & +\infty \\ \vdots & \ddots & \vdots \\ +\infty & \dots & t(E[N, o]) \end{pmatrix}$$

Алгоритм Витерби методами ЛА

Обработка первого наблюдения из последовательности Obs :

$$Probs_1 = P(Obs[1]) \times B$$

Оставшаяся часть Obs :

$$Probs_t = P(Obs[t]) \times T^T \times Probs_{t-1}$$

Алгоритм специализации

Как использовать данные СММ?

Для всех наблюдений o можно вычислить:

$$P(o) \times B, \text{ далее как } PB(o)$$

$$P(o) \times T^T, \text{ далее как } PT(o)$$

Алгоритм Витерби первого уровня специализации

$$Probs_1 = PB(Obs[1])$$

$$Probs_t = PT(Obs[t]) \times Probs_{t-1}$$

Матричное умножение ассоциативно

Обработка o_1 и o_2 , если столбец $Probs_0$ известен?

$$\begin{aligned} Probs_2 &= PT(o_2) \times Probs_1 \\ &= PT(o_2) \times (PT(o_1) \times Probs_0) \\ &= (PT(o_2) \times PT(o_1)) \times Probs_0 \end{aligned} \tag{1}$$

Можно вычислить $PT(o_2) \times PT(o_1)$. Два наблюдения одним умножением, т.е. второй уровень!

Этот подход можно применить для повышения уровня

Анализ операций

Пусть длина *Obs* обозначается как *lo*, количество состояний СММ как *N*, количество возможных наблюдений СММ как *K*

Неспециализированная версия

Матричных умножений:

$$1 + 2 * (lo - 1)$$

Специализированная версия уровня *M*

Матричных умножений:

$$(lo - 1) / M + (lo - 1) \bmod M$$

Дополнительной памяти: K^M матриц, каждая $N \times N$

Нужно и для CPU, и для GPGPU

Библиотеки SUITESPARSE:GRAPHBLAS и cuASR

Проведено тестирование

- на корректность реализации алгоритма Витерби
- на сохранение семантики
- отсутствие утечек памяти

- Ubuntu 20.04
Intel Core i7-6700 3.40 ГГц, 64 Гб RAM
NVIDIA GeForce GTX 1070, 8Гб, 1920 CUDA-ядер
- Сравнение специализированной версии с неспециализированной и CUDAMPF
- 24 CMM из репозитория CUDAMPF, 4 набора данных
- Медиана из 10 запусков

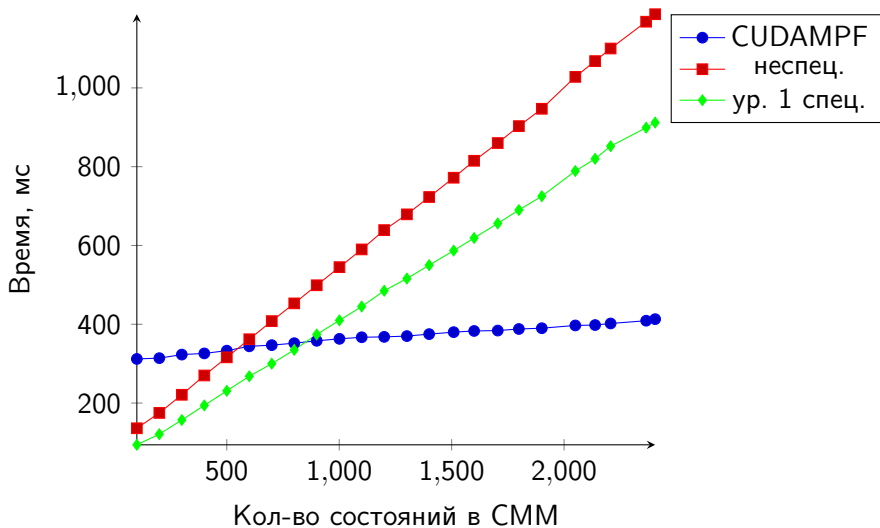


Рис. 1: GraphBLAS, набор данных из БД PFAM, меньше — лучше

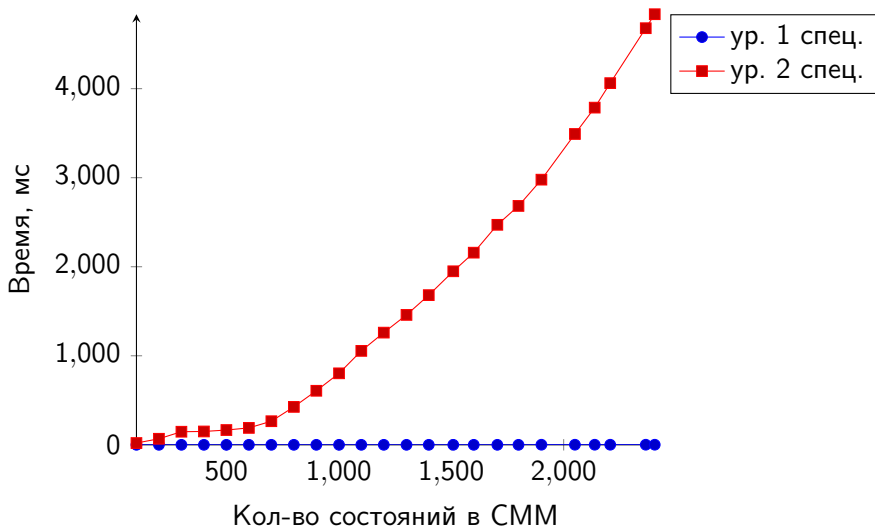


Рис. 2: GraphBLAS, время на специализацию, меньше — лучше

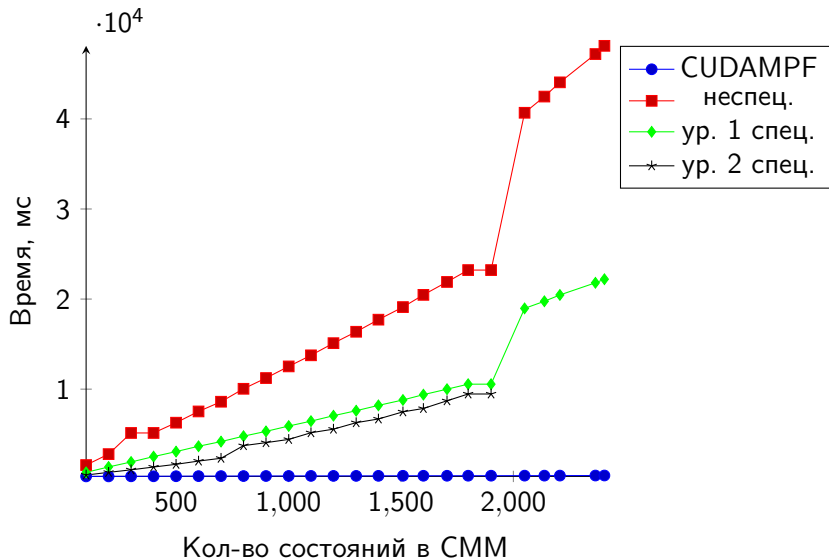


Рис. 3: cuASR, набор данных из БД PFAM, меньше — лучше

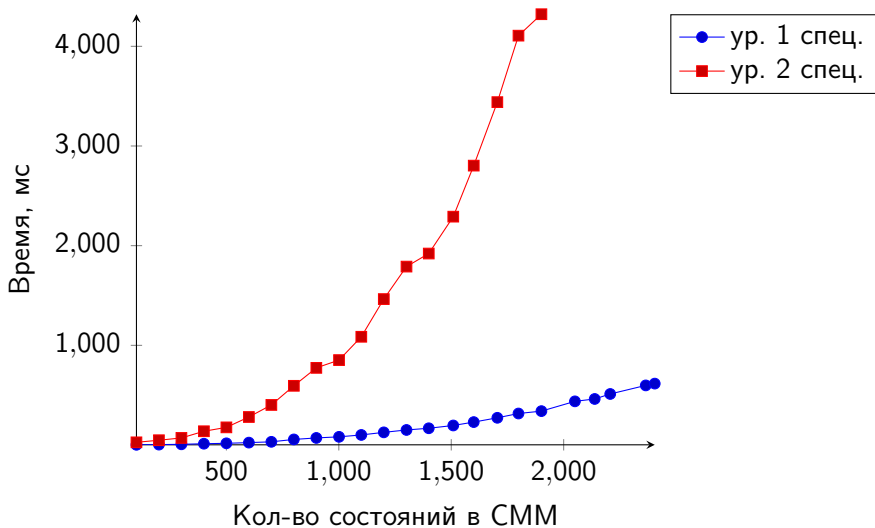


Рис. 4: сиASR, время на специализацию, меньше — лучше

- Выполнен обзор предметной области
- Реализованы и протестированы две реализации специализированного алгоритма Витерби
 - ▶ SUITESPARSE:GRAPHBLAS для выполнения на CPU
 - ▶ cuASR для выполнения на GPGPU
- Проведены эксперименты на данных из репозитория CUDAMPF

Специализированная версия алгоритма Витерби превосходит неспециализированную

SEIM 2021: статья *Viterbi Algorithm Specialization Using Linear Algebra*

Ответы на вопросы рецензента

- 1 Подход не зависит от бионформатики и применим в областях, где СММ может быть фиксирована
- 2 Не было найдено инструментов для специализации с возможностью оптимизации ЛА
- 3 На графиках указано время обработки без учёта специализации
- 4 Точная причина не установлена
- 5 Количество CUDA-ядер у NVIDIA GTX 1070 равно 1920
- 6 Указано время для обработки СММ с кол-вом состояний менее 2000

Пример специализации: возведение в степень

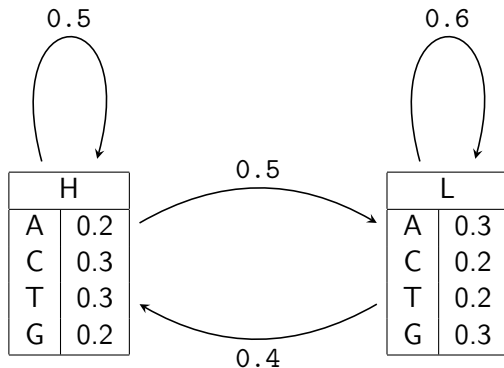
```
function f(x, n)
  if n == 0 then 1
  elif even(x) then f(x, n/2)^2
  else x * f(x, (n-1)/2)^2
```

Предположим $n = 5$, т.е. это статический параметр

```
function f_spec(x) = x * (x^2)^2
```

Специализация бесполезна, если x — статический параметр

СММ: пример



Вероятности быть стартовым состоянием

H	L
0.5	0.5

Переопределяем '+' как *min*, '*' как *plus*, ∞ и 0 как нейтральные элементы

Пример матричного умножения

$$\begin{pmatrix} 0 & 1 \\ +\infty & 2 \end{pmatrix} \times \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} \min(0 + 3, 1 + 4) \\ \min(+\infty + 3, 2 + 4) \end{pmatrix} = \begin{pmatrix} 3 \\ 6 \end{pmatrix}$$

Обработка o_{t-2} , o_{t-1} и o_t , если столбец $Probs_{t-3}$ известен?

$$Probs_t = PT(o_t) \times PT(o_{t-1}) \times PT(o_{t-2}) \times Probs_{t-3} \quad (2)$$

Все произведения $PT(o)$ могут быть вычислены по данным из СММ

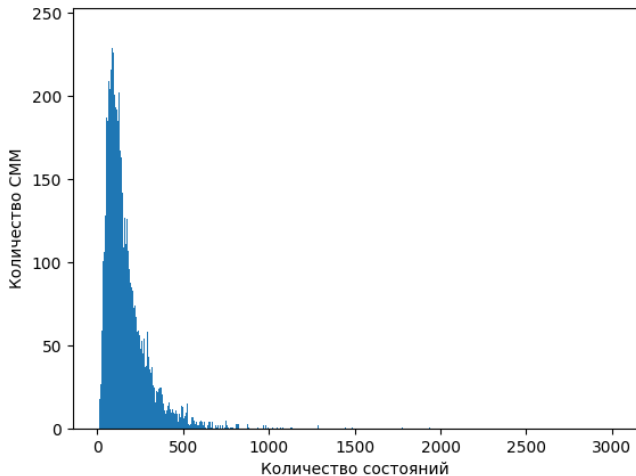


Рис. 5: График зависимости количества СММ от количества состояний в них

	CUDAMPF	неспец.	Ур. 1	Ур. 2
3 x 3500	4854	10765	8062	215329
3 x 7000	9209	21062	16152	387464
Набор из PFAM	8796	15864	12036	298269
50 x 3500	103036	176263	134259	2921104

Таблица 1: GraphBLAS, общее время обработки с учетом затрат времени на специализацию, мс

	CUDAMPF	неспец.	Ур. 1	Ур. 2
3 x 3500	3666	154415	77758	85977
3 x 7000	7053	305578	153216	145176
Набор из PFAM	6777	241687	114214	114766
50 x 3500	78930	2596532	1271980	1025473

Таблица 2: cuASR, общее время обработки с учетом затрат времени на специализацию, кол-во состояний CMM меньше 2000, мс