В 30-е годы прошлого века был опубликован популярный комикс Flash Gordon, в последующем он был неоднократно экранизирован и его главный герой, футболист Флэш Гордон, стал одним из наиболее популярных супергероев.

заняли примерно 2,5 года, контрольный срок запуска в эксплуатацию назначен на 1 января 2012 года, срок эксплуатации определен равным трем годам. При выборе архитектуры главными были два соображения или, скорее, учет двух ограничений. Первое – необходимость преодолеть разрыв в темпах роста производительности процессоров и памяти и следующий из него дисбаланс между скоростью работы процессоров и систем хранения данных. Его еще иногда называют «красным смещением». Второе – во многих случаях приложения класса Data-Intensive по природе своей являются параллельными и для них не подходит архитектура MPP (Massively Parallel Processing). Проектируемая система должна была быть ближе к одноименной аббревиатуре HPC, но расшифровываемой как High Performance Computing («высокопроизводительные компьютерные системы»).

Для преодоления этих двух ограничений следует изменить иерархию памяти компьютера. В современных системах она имеет пять уровней. Верхний уровень – регистры процессора. Ниже расположен второй уровень – кэш-память, она может, в свою очередь, делиться на 2 или 3 уровня, но это не видно вне процессора, в среднем операции над данными здесь выполняются в 2–10 раз медленнее, чем на регистровом уровне. Третий уровень – оперативная память, она на 2 порядка медленнее регистров, на ней ограничивается распределенный доступ (shared memory programming). Четвертый сверху уровень образуется за счет системы обмена сообщениями между узлами, здесь обмен происходит со скоростью на 4 порядка медленнее, чем между регистрами процессора. Самый нижний уровень – дисковый, он еще на 7–8 порядков ниже. Слабость такой архитектуры, во-первых, в ограничении распределенного доступа тремя верхними уровнями и, во-вторых, в разрыве между четвертым и пятым уровнями. Появление дисков SSD позволяет закрыть разрыв, они на 2 порядка быстрее HDD, а конструкция Gordon дает возможность распространить подход shared memory и на эти диски.

В итоге прямоадресуемое пространство памяти становится адекватным задачам, попадающим в категорию Big Data.

Далеко не все технические характеристики Gordon открыты, однако по тому, что доступно, можно прийти к следующим выводам.

Архитектура системы представляет собой двухуровневый кластер. Основной структурной единицей являются суперузлы (Supermode), состоящие из 32 процессоров Intel Westmere (Intel Sandy Bridge Compute node), количество ядер и тактовая частота не объявлены, в SMP они объединяются посредством ПО vSMP. Каждый суперузел комплектуется двумя узлами дисков SSD, в состав узла входит до 16 дисков NAND Flash с суммарной емкостью от 256 Гбайт до 4 Тбайт и установившейся скоростью обмена – 170 Мбит/с на диск и 2,7 Гбит/с на узел. Прямоадресуемое пространство DRAM – до 2 Тбайт, а включая флэш-память – 10 Тбайт. Система Gordon может включать до 32 суперузлов, то есть всего 1024 вычислительных узла. Между собой они связаны посредством сети Dual rail QDR InfiniBand с топологией 3D тор (4x4x4) и суммарной скоростью обмена 9,2 Тбайт/с. Дисковая система на HDD – до 4,5 Пбайт, работает под управлением файловой системы Lustre и обеспечивает скорость обмена 100 Гбайт/с. Возможны варианты суперузлов меньшего размера, например, 16 процессоров и без поддержки SMP.

Строго по плану 15 декабря Gordon был представлен публике, и с 1 января начнется его эксплуатация. Директор SDSC Майкл Норман сказал: «Мы являемся свидетелями новой эры суперкомьютинга, ориентированного на большие данные. В Gordon я вижу новый корабль, который позволит нам отправиться в путешествие к еще не открытым континентам науки. В его флэш-память можно записать такое количество прямоадресуемых данных, которое превратит его в самое большое и самое быстрое хранилище данных, обеспечивающее требуемую аналитическую обработку». В рамках национальной программы XSEDE (Extreme Science and Engineering Discovery Environment) предполагается построить 16 экземпляров Gordon.

Рандомизированные алгоритмы при обработке данных

Текст Олег Граничин

Развитие средств контроля и вычислительной техники позволяет в настоящее время перейти к решению многих практических задач «на лету», встраивая «умные» блоки в контуры управления простых и сложных систем, в технологические процессы, в разнообразные системы поддержки принятия решений и т. п. Но во многих задачах ресурсы существенно ограничены, а данных недостаточно. В последнее время в научной литературе активно развиваются рандомизированные методы решения

Рандомизация часто позволяет «обогатить» данные наблюдений и разработать «быстрые» алгоритмы, существенно сокращающие количество операций и «подавляющие» влияние систематических ошибок (эффект смещения при произвольных помехах), точность которых обычно несущественно зависит от размерности данных.

В рандомизированных алгоритмах, в отличие от традиционных детерминированных, один или несколько шагов основываются на случайном правиле, при котором среди многих детерминированных правил одно выбирается в соответствии с некоторой случайной схемой, т. е. при использовании рандомизированных алгоритмов на каком-то этапе вместо того, чтобы самим принять решение, мы призываем «судьбу» (случай) выбирать за нас. Но тогда естественно возникает вопрос: зачем прибегать к «судьбе» мудрому человеку? «Судьба» не является специалистом ни в чем, выбор делается случайно. Итак, почему от рандомизированных алгоритмов может быть польза? Сознательное использование рандомизированных методов требует ответов на этот вопрос, часть из которых обсуждается в статье.

1. Возможно ли осмысленное оценивание при произвольных внешних помехах?

Рассмотрим для примера простейшую задачу оценивания:

$y_t = \theta \cdot u_t + v_t$, $t = 1, 2, ..., N$,

в которой можно:

- выбирать входы u_t,
- измерять выходы y_t.

Требуется по последовательности входов и выходов $\{u_t, y_t\}$ определить неизвестный параметр θ при отсутствии каких-либо ограничений на последовательность помех $\{v_t\}$.

Не кажется ли такая постановка задачи абсурдной?

С детерминистской точки зрения – конечно! Не может быть никакого детерминированного алгоритма, дающего хотя бы в каком-то смысле здравый ответ (кроме бессмысленного решения – вся числовая ось). Предложив в качестве ответа любое из чисел или даже какой-то интервал при конечном (или счетном) числе наблюдений, всегда можно будет подобрать такие v_t, что при следующем наблюдении предложенный ответ будет неверным.

Общий алгоритм последовательного оценивания неизвестного параметра θ состоит из двух шагов:

1. Выбор входа u_t.
2. Оценивание параметра θ на основе полученных данных u_t, y_t (вычисление, например, числовой оценки $\hat{\theta}$ или множества $\hat{\Theta}$, содержащего θ).

Если бы в условиях задачи дополнительно можно было бы предположить случайную (вероятностную) природу помех v_t, то при выполнении условий закона больших чисел можно было бы говорить об оценивании неизвестного параметра θ путем простого усреднения данных наблюдения. Результаты моделирования с истинным параметром $\theta = 3$ и наблюдениями, которые производились с равномерно распределенной на интервале [–0.5; 0.5] помехой v_t, приведены в таблице:

t	1	2	3	4	5	6	7
u_t	1	1	1	1	1	1	1
y_t	2.9	2.8	3.2	3.3	2.6	3.4	2.7
$\hat{\theta}_t$	2.9	2.85	2.97	3.05	2.96	3.03	2.99

из которой видно, что $\hat{\theta}_7 = 2.99$ уже близко к истинному параметру $\theta = 3$.

Если наблюдения проводить также со случайной помехой, но у которой среднее значение m было бы неизвестно, то результаты моделирования (например, при m =1) показывают ошибочность работы алгоритма $\hat{\theta}_7 = 3.99$ и существенно превосходят $\theta = 3$.

t	1	2	3	4	5	6	7
u_t	1	1	1	1	1	1	1
y_t	3.9	3.8	4.2	4.3	3.6	3.9	4.2
$\hat{\theta}_t$	3.9	3.85	3.97	4.05	3.96	4.03	3.99

Несмотря на кажущуюся абсурдность постановки задачи оценивания при произвольных внешних помехах, из практических потребностей часто ее все-таки приходится решать.

Альтернативой оказываются рандомизированные алгоритмы, в которых выполнение одного или нескольких шагов основано на случайном правиле (т. е. среди многих детерминированных правил одно выбирается случайно в соответствии с вероятностью P).

В зависимости от специфики конкретной задачи вероятность P или является искусственным элементом, вводимым в алгоритм для улучшения разрешимости проблемы, или в рассматриваемой системе могут присутствовать измеряемые случайные элементы. Выбор этой вероятности P является частью конструирования алгоритма.

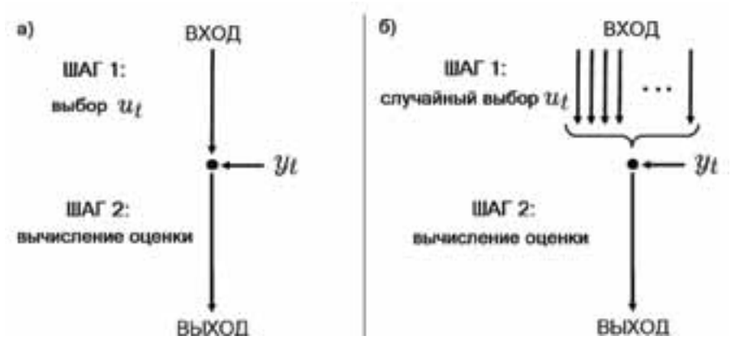
Рассмотрим следующее правило случайного выбора для первого шага рандомизированного алгоритма последовательного оценивания неизвестного параметра θ,
т. е. на первом шаге случайно выбирается один из 2^7 возможных наборов входов (управлений).

$$u_t = \begin{cases} +1, & \text{с вероятностью } \frac{1}{2}, \\ -1, & \text{с вероятностью } \frac{1}{2}. \end{cases}$$

На втором шаге по известным парам значений (u_t, y_t) формируем величины $\bar{y}_t = u_t \cdot y_t$. Для «новой» последовательности наблюдений справедлива похожая на исходную модель $\bar{y}_t = \theta \cdot \bar{u}_t + \bar{v}_t$, в которой
$\bar{u}_t = u_t \cdot u_t$, $\bar{v}_t = u_t \cdot v_t$.

Пусть, как и ранее при моделировании, v_t – случайные помехи с неизвестным математическим ожиданием. Если v_t – внешние помехи, то естественно считать, что они независимы с нашим рандомизированным правилом выбора входов на шаге 1.
Следовательно,
$E\, v_t = E\, u_t \cdot v_t = E\, u_t \cdot E\, v_t = 0 \cdot E\, v_t = 0$,
т. е. «в новой модели» наблюдений задача об оценивании неизвестного параметра θ, не имевшая решения, превращается при использовании случайного правила выбора входов на шаге 1 рандомизированного алгоритма в «стандартную» задачу об оценивании неизвестного параметра θ, наблюдаемого на фоне независимых центрированных помех.

Детерминированный (а) и рандомизированный (б) алгоритмы



Ниже сведены соответствующие результаты имитационного моделирования.

Как видно, результаты существенно лучше, чем ранее, но в отличие от результатов первой таблицы качество оценок получилось ниже, так как «новые ошибки» $\bar{v}_t$ имеют большую дисперсию по сравнению с v_t.

t	1	2	3	4	5	6	7
u_t	–1	1	–1	1	1	1	–1
y_t	–2.1	3.8	–1.8	4.3	3.6	4.4	–2.3
$\bar{u}_t$	1	1	1	1	1	1	1
$\bar{y}_t$	2.1	3.8	1.8	4.3	3.6	4.4	2.3
$\hat{\theta}_t$	2.1	2.95	2.57	3.00	3.12	3.33	3.19

Можно получить и более строгий математический результат о гарантированном множестве возможных значений неизвестного параметра θ для произвольных внешних помех v_t. Опуская технические детали, опишем метод построения доверительного интервала.

1. Пусть M = 8 и выберем случайно семь (= M – 1) разных групп по четыре индекса.
2. Вычислим семь соответствующих частичных сумм.
3. Доверительный интервал для θ с вероятностью $p = 75\%$ (= $1 - 2 \cdot 1/M$) сформируем как минимальный отрезок, содержащий все семь полученных на втором этапе чисел.

По описанному методу получаем интервал $\hat{\Theta} = [2.875; 3.4]$, который содержит неизвестный параметр θ с вероятностью $p = 75\%$, что вполне соотносится с условиями задачи, так как при моделировании использовалось фактическое значение $\theta = 3$.

Взяв вместо семи пятнадцать ($M = 16$) частичных сумм s_i, можно вдвое сократить вероятность ошибки, получив $p = 87.5\%$, но при этом, вообще говоря, получится больший доверительный интервал $\hat{\Theta}$ (см. статью автора «Неасимптотическое доверительное множество для параметров линейного объекта управления при почти произвольных помехах» в журнале «Автоматика и телемеханика», 2012, № 1, с. 24–35).

Итак, для, казалось бы, абсурдной задачи об оценивании параметра при произвольных внешних помехах, с которой принципиально не может справиться ни один детерминированный алгоритм, внесение рандомизации в процесс выбора входов дает получить вполне осмысленные результаты, позволяя говорить о вероятностной успешности рандомизированного алгоритма с некоторым параметром (вероятностью) p.

Достижение успешных результатов с высокой степенью вероятности, в отличие от детерминированного случая, соответствует компромиссу: если полностью гарантированный результат получить невозможно, то лучше иметь 75%-ную гарантию, чем не иметь ничего.

Конечно, не во всех задачах компромисс возможен. Во многих случаях нужен ответ, гарантированный на 100%. Но, «защищая» рандомизированные алгоритмы, надо отметить, что уровень достоверности p обычно является параметром алгоритма, который может быть настроен пользователем. Параметр p ослабляет понятие детерминированной разрешимости, для которой вероятность успеха может быть только 0 или 1 – образно выражаясь, результат «черный» или «белый». Переходя к рандомизированным алгоритмам, p становится непрерывным параметром, пробегающим интервал [0; 1], задавая тот или иной оттенок «серого».

Замечание. Альтернативный вероятностный подход к решению задачи оценивания – байесовский, при котором присутствующим в системе помехам v_t априори приписывается вероятностная природа Q, но его невозможно применить при произвольных внешних помехах (в худшем случае), так как все выводы имеют вероятностную основу предположений о системе. По смыслу байесовский и рандомизированный подход совершенно различны с практической точки зрения. В байесовском Q описывает вероятность того или иного значения помехи v_t по сравнению с другими, т. е. выбор Q является частью модели задачи.

В отличие от этого вероятность P в рандомизированном подходе является тем, что мы искусственно выбрали и используем. P существует только в нашем алгоритме, и, следовательно, нет традиционной проблемы плохой модели, как это может случиться с Q при байесовском подходе.

2. Рандомизация как способ снижения вычислительной сложности.

Многие авторы выступают за использование рандомизированных алгоритмов для уменьшения сложности вычислений по сравнению с более традиционными детерминированными подходами. Идея состоит в следующем: предположим, что детерминированный алгоритм требует чрезмерных вычислений для обработки всей доступной информации. Тогда можно сознательно отказаться от части информации и перейти к решению упрощенной задачи с частичной информацией. При этом, однако, детерминированной разрешимости, может быть, достичь невозможно, но, как и ранее, можно рассмотреть рандомизированный подход к определению решения с высокой вероятностью успеха.

Конечный результат: компромисс между полной гарантией успеха и вычислительной реализуемостью (возможностью получить ответ за реальное ограниченное время).

В рамках этой логики впечатляющие результаты были получены для рандомизированных подходов в основных областях теории управления и систем. Это относится, например, к выпуклым проблемам робастного управления, включая в том числе широкий класс задач управления, сводящихся к параметрозависимым линейным матричным неравенствам (LMIs). Как известно, многие из этих проблем являются NP-трудными в детерминированной постановке. В связи с массовым переходом к обработке потоков двумерных (2D) и трехмерных (3D) данных резко увеличились объемы обрабатываемой информации. Сложность традиционных методов квантования сигналов возрастает по экспоненциальному закону с ростом размерности. Квантование 1D-сигналов при $N = 10$ отсчетов соответствует 100 в случае 2D, а в 3D – 1000, что уже чрезвычайно велико. В современных приложениях для цифровых фото- и видеокамер традиционное требование к необходимой частоте (скорости, Nyquist Rate) измерения настолько высоко, что слишком большое количество получающихся данных надо существенно сжимать перед хранением или пересылкой. В других приложениях, включая системы отображения (медицинские сканеры, радары) и быстродействующие аналого-цифровые конвертеры, увеличение частоты измерений оказывается очень дорогостоящим.

Новая парадигма Compressive Sensing («опознание со сжатием») позволяет достаточно точно восстанавливать «разреженную» (sparse) информацию – вектор θ размерности d по вектору наблюдений y размерности m при условии $m \ll d$. При этом используются рандомизированные измерения, по результатам которых информация восстанавливается с помощью методов l_1-оптимизации.

Успеху рандомизированных алгоритмов способствует еще и то, что их производительность и вероятность их успеха может быть достаточно точно оценена аналитическими методами.

Рождение рандомизированных методов восходит к появлению метода статистического моделирования Монте-Карло, предложенного в Лос-Аламосе Н. Метрополисом и С. Уламом в ходе работ над Манхэттенским проектом.

Детальному анализу возможностей рандомизированных алгоритмов в задачах оценивания и оптимизации при произвольных помехах посвящена книга автора и Б. Т. Поляка «Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах» (М.: «Наука», 2003). При решении задач глобальной оптимизации в настоящее время активно используются алгоритмы имитации отжига и генетические алгоритмы, в основе которых также лежат рандомизированные правила.

Надеюсь, что представленные в статье мысли о рандомизированных методах будут стимулировать дальнейшие размышления и могут помочь читателям получить более глубокое понимание потенциала и ограничений рандомизации.