Применение практики к теории

Григорий Ярославцев

Академический Физико-Технологический Университет РАН
http://logic.pdmi.ras.ru/~grigory

17 мая, 2009

План

1. Введение
2. Обзор существующих работ
   - Слабо экспоненциальные алгоритмы
   - Аппроксимируемость и неаппроксимируемость

Использование вычислительных мощностей

Что мы можем сделать, используя свободные вычислительные мощности современных компьютеров?

- Изучение свёртывания белков
- Улучшение климатических моделей
- Поиски внеземного интеллекта
- Поиск чисел Мерсенна
- ...

Мы рассмотрим некоторые применения автоматических методов для задач теоретической информатики.

Слабо экспоненциальные алгоритмы

- Цель — разработка более быстрых алгоритмов для точного решения NP-трудных задач.
- Например, интересно было бы найти алгоритм со временем работы $O(1.9^n \cdot \mathrm{poly}(m))$ для задачи выполнимости булевой схемы.
- Сейчас нет идей, как это можно сделать. Более того, неизвестно никаких интересных следствий из факта существования такого алгоритма.
- Однако известно решение задачи 3-SAT за время $O(1.33^n)$ (Iwama, Tamaki, 2004).

Минимальное вершинное покрытие (MVC)

$N(v)$ — множество соседей вершины v в графе G.

Алгоритм:

- Если степени всех вершин меньше трёх, то решается за полиномиальное время.
- Если есть вершина степени 0, то удаляем её.
- Если есть вершина степени 1, то удаляем её и её соседа, добавляя его к ответу.
- Иначе выбираем вершину v с максиальной степенью.
  - Решаем задачу для графа $G \setminus \{v\}$, находя минимальное покрытие C_1.
  - Решаем задачу для графа $G \setminus \{v \cup N(v)\}$, находя минимальное покрытие C_2.
  - Возвращаем минимум из $C_1 \cup \{v\}$ и $C_2 \cup \{N(v)\}$.

Анализ времени работы алгоритма для MVC

- Рекуррентное соотношение на время работы (n – число вершин в G):
$$T(n) \leq T(n-1) + T(n-4) + \mathrm{poly}(n)$$
Находя вещественный корень $f(x) = 1 - 1/x - 1/x^4$, находим, что $T(n) \leq 1.39^n$.
- Оценим время работы через количество рёбер m:
$$T(m) \leq T(m-3) + T(m-5) + \mathrm{poly}(n)$$
Получающаяся оценка: $T(m) \leq 1.19^m$.
- Комбинация двух оценок:
$$T(m,n) \leq T(m-3, n-1) + T(m-5, n-4) + \mathrm{poly}(n)$$

Анализ времени работы алгоритма для MVC

- Положим $k = \alpha_1 m + \alpha_2 n$. Тогда:

$$T(k) \leq T(k - 3\alpha_1 - \alpha_2) + T(k - 5\alpha_1 - 4\alpha_2) + \text{poly}(k)$$

  Теперь $T(k) \leq O(c^k)$, где c — вещественное решение уравнения $1 - 1/x^{3\alpha_1+\alpha_2} - 1/x^{5\alpha_1+4\alpha_2} = 0$.
- Например, при $\alpha_1 = 1/2$ и $\alpha_2 = 1$ оценка на время работы получается $O(1.21^{m/2+n})$.
- Так можно интерполировать между двумя предыдущими оценками.

Оптимизация оценок от нескольких переменных

- Приближённое решение при помощи записи в виде квазивыпуклой программы (Eppstein, 2006).
- Можно следить за количеством вершин степени i, введя переменные n_i.
- Результаты:
  - $O(1.52^n)$ для задачи о доминирующем множестве (Fomin, Grandoni, Kratsch, 2005)
  - $O(1.23^n)$ для задачи о минимальном вершинном покрытии (Fomin, Grandoni, Kratsch, 2006)

MAX 3-SAT

- Случайный набор значений переменных удовлетворяет хотя бы $7/8$ клозов в 3-КНФ формуле (в каждом клозе должны встречаться 3 различных переменных!).
- Задача о нахождении $(7/8+\varepsilon)$-приближения (для любого $\varepsilon > 0$) для этой задачи *NP*-трудна. (Håstad, 2001).

Гаджеты для сведений

Рассмотрим стандартное сведение 3-SAT к MAX 2-SAT (Garey, Johnson, Stockmeyer, 1976):

- Клозу $c_i = (l_1 \vee l_2 \vee l_3)$, где l_1, l_2, l_3 — литералы, сопоставим гаджет из 2-CNF клозов:

$$(l_1), (l_2), (l_3), (y_i), (\neg l_1 \vee \neg l_2), (\neg l_1 \vee \neg l_3), (\neg l_2 \vee \neg l_3),$$
$$(l_1 \vee \neg y_i), (l_2 \vee \neg y_i), (l_3 \vee \neg y_i)$$

- Если набор выполняет клоз c_i, то ровно 7 из 10 клозов гаджета могут быть выполнены.
- В противном случае ровно 6 из 10 клозов могут быть выполнены.

MAX 2-SAT и MAX 3-SAT

Предложение

Если есть $(1-\varepsilon)$-приближённый алгоритм для задачи MAX 2-SAT, то путём применения предыдущего сведения из него получается $(1-7\varepsilon)$-приближённый алгоритм для задачи MAX 3-SAT.

Доказательство

- *Пусть m_3 — число клозов в записанной 3-КНФ формуле F, m_3^* — максимальное число клозов, которые могут быть выполнены в F, $\hat{m}_3$ — максимальное число клозов, которые могут быть выполнены путём запуска сведения.*
- *По предположению из условия $1-\varepsilon \leq \frac{6m_3+\hat{m}_3}{6m_3+m_3^*}$*
- *Используя то, что $m_3^* \leq m_3$, получаем $\frac{\hat{m}_3}{m_3^*} \geq 1-7\varepsilon$.* □

Формализация понятия гаджета

Следствие

MAX 3-SAT не имеет $(7/8+\varepsilon)$-приближённого решения, если $P \neq NP$, следовательно задача MAX 2-SAT не имеет $(55/56+\varepsilon)$-приближённнного решения, если $P \neq NP$.

Определение

Пусть $\alpha, l, n \geq 1$, $f : \{0,1\}^k \to \{0,1\}$ и $\mathcal{F}$ — семейство функций из $\{0,1\}^{k+n}$ в $\{0,1\}$. α-гаджет, сводящий f к $\mathcal{F}$, задаётся множеством дополнительных переменных $y_1, \ldots, y_n$ и весами $w_j \geq 0$ вместе с ограничениями $C_j \in \mathcal{F}$, где $j = 1, \ldots, l$. Для всех $a \in \{0,1\}^k$,

- *Если $f(a) = 1$, то $(\forall b \in \{0,1\}^n) \sum_j w_j C_j(a,b) \leq \alpha$ и $(\exists b \in \{0,1\}^n) \sum_j w_j C_j(a,b) = \alpha$*
- *Если $f(a) = 0$, то $(\forall b \in \{0,1\}^n) \sum_j w_j C_j(a,b) \leq \alpha - 1$*

Анализ формализации понятия гаджета

- Веса можно добавлять, так как это эквивалентно размножению ограничений
- Сведение 3-SAT к MAX 2-SAT является 7-гаджетом, сводящим $f(x_1, x_2, x_3)$ к семейству функций, представимому 2-клозами.
- В таком сведении $n = 1, l = 10$ и $w_j = 1$ для всех j.

Поиск гаджетов и линейное программирование

- Фиксируем n и $(C_1, \ldots, C_l) \in \mathcal{F}^l$.
- Тогда задача поиска w_j для минимизации α сводится к решению задачи линейного программирования, имеющей неравенства для всех $a \in \{0,1\}^k$.
- Это сведение почти тривиально за исключением того, что в случае $f(a) = 1$ нужно записать условие существования такого b, что достигается равенство суммы α.
- Можно перебрать все функции B из множества выполняющих наборов f в $\{0,1\}^n$.
- Тогда для любого a такого, что $f(a) = 1$ мы просто полагаем $b = B(a)$ в равенстве $\sum_j w_j C_j(a, b) = \alpha$ задачи линейного программирования.
- Получающая задача линейного программирования имеет большой размер.

Нахождение оптимальных гаджетов

- Возможно, что при увеличении n гаджеты будут улучшаться.
- При выполнении некоторых естественных условий доказано, что достаточно рассматривать $n \leq 2^s$, где s — число выполняющих наборов f.
- Эти условия выполняются для 2-КНФ и многих других семейств ограничений.
- С помощью компьютера найден оптимальный 3.5-гаджет для сведения 3-SAT к MAX 2-SAT:

$$(l_1 \vee l_3), (\neg l_1 \vee \neg l_3), (l_1 \vee \neg y_i), (\neg l_1 \vee y_i), (l_3 \vee \neg y_i), (\neg l_3 \vee y_i), (l_2 \vee y_i)$$

  Здесь веса всех клозов кроме последнего равны $1/2$, а вес последнего равен 1.

Спасибо за внимание!

Доклад основан на обзорной статье Райана Вильямса “Applying Practice to Theory”, которую можно найти по адресу http://arxiv.org/abs/0811.1305