Лекция 12. Вероятностные компьютеры

12 декабря 2015 г.

Теория алгоритмов 2015

Определение

Вероятностная машина Тьюринга (ВМТ) - недетерминированная машина Тьюринга, в которой переходы по веткам определяются вероятностым образом.

Сразу хочется ввести классы сложности, связанные с такими задачами. Например, обозначить задачи, которые можно посчитать на ВМТ за полином. Назовем этот класс PP и определим следующим образом:

Определение

$L \subset PP : \exists M, w \in L \Leftrightarrow P(M(w) = 1) > 0.5$ *и машина* M *- полиномиальная.*

То есть, мы берем слово, загружаем его в ВМТ, эта машина говорит нам "да"или "нет". Для тех слов которые лежат в этом языке ВМТ должна говорить "да"с вероятностью больше чем 0.5.

На самом деле это плохое определение, так как при таком определении $NP \subset PP$. Чтобы это доказать, возьмем NP-полную задачу SAT $F(x_1, ..., x_n)$ и воспользуемся следующим вероятностным алгоритмом M_{sat}:

Алгоритм

Шаг1: $x_i = \begin{cases} 1, p = 1/2 \\ 0, p = 1/2 \end{cases}$ *(генератор случайных чисел)*

Шаг 2: Считаем $F(x_i)$

Шаг 3: if $(F(x_i) = 0)$

{

return $\begin{cases} 1, p = 1/2 \\ 0, p = 1/2 \end{cases}$

}

else return 1;

Теорема

$NP \subset PP$

Доказательство

Оценим вероятность того что $w \not\subset SAT$. *Так как мы будем всегда попадать в if* $(F(x_i) = 0)$, *то* $P(M(w) = 1) = 0.5$
Допустим, что теперь $w \subset SAT$. *Пусть есть* k *кортежей, которые удовлетворяют условию, тогда рассмотрим следующие вероятности:*
$P(M(v) = 1 \mid \text{повезло}) = 1$
$P(\text{повезло}) = \frac{k}{2^n}$
$P(M(v) = 1 \mid \text{не повезло}) = 1$
$P(\text{не повезло}) = 1$
Используя формулу полной вероятности получаем:
$P(M(w) = 1) = \frac{k}{2^n} + (1 - \frac{k}{2^n})\frac{1}{2} > 0.5$
Это и доказывает, что SAT *лежит в* PP

Замечание

Если алгоритм M_{sat} выдает на выходе 1, это еще ничего не значит. Что бы это что-то значило, нужно провести статистически важное число испытаний. Другими словами, провести экспоненциальное число испытаний.

Теорема

$PP \subset PSPACE$

Доказательство

Нам надо показать: $L \subset PP \to L \subset PSPACE$

Возьмем какой нибудь язык

$L \subset PP \to \exists M, w \in L \Leftrightarrow P(M(w) = 1) > 0.5$ *и машина* M *- полиномиальная*

Хотим построить другую машину M' *- обычную ДМТ, которая вычисляет* $M'(w, M) = P(M(w) = 1)$. *Это можно сделать полным перебором с возвратом в классе* $PSPACE$.

Продолжение доказательства

M' - полиномиальна. Это означает, что количество ветвлений в любом случае будет не больше чем полином. Перебор в глубину будет не более чем полиномиальный $\rightarrow$ полиномиальной памяти хватит чтобы сохранить одно состояние перебора. M' работает за полиномиальную память $\rightarrow L \subset PSPACE$

Замечание

В иерархии классов PP лежит где-то между NP и $PSPACE$

Введем правильное определение

Определение

$L \subset BPP : \exists M, w \in L \Leftrightarrow P(M(w) = 1) > \frac{2}{3}$ и машина M - полиномиальная.

Замечание

На самом деле вместо $\frac{2}{3}$ можно взять любое число больше $\frac{1}{2}$. Потому что при k испытаниях вероятность ошибки изменится экспоненциально, а сложность запуска полиномиально.

Рассмотрим этот момент подробнее

Допустим у нас есть $P(M(w) = 1) \geq \frac{1}{2} + k$. Надо перейти к $P(M(w) = 1) \geq \frac{2}{3}$. Рассмотрим на примере 2-ух монет.

Есть 2 монеты:

Первая:

0.5 + p - веротятность выпадения орла

0.5 - p - веротятность выпадения решки

Соответствует случаю, когда $w \subset L$

Вторая:

0.5 - p - веротятность выпадения орла

0.5 + p - веротятность выпадения решки

Соответствует случаю, когда $w \not\subset L$

Как понять с какой монетой мы имеем дело?

Замечание

Кидая достаточное количество раз монетку мы можем сделать веротяность ошибки сколь угодно маленькую. Вероятность ошибиться падает экспоненциально

Замечание

Возникает вопрос: $BPP = PP$?(например мы можем запустить алгоритм Монте-Карло, подождать полиномиально долгое число бросков и получить ответ). Поэтому возможно $BPP = PP$, но до сих пор не известно. Тем не менее многие задачи из BPP со временем оказывались в P (важный пример - проверка простоты).

Существует хороший пример задачи, про которую известно, что она лежит в BPP и на сегодняшний день неизвестно лежит ли в P. Это задача о проверке полинома на тождественное равенство нулю.

Задача

$P(x_1, ..., x_n)$ - полином от n пемененных, задан в виде произведения полиномов или как определитель полиномиальной матрицы.
Иначе говоря, полином задан таким образом, что мы не можем посчитать его коэффиценты(так как это будет полиномиальная задача).
Задача: проверить равен ли полином P нулю тождественно.

Замечание

Никто пока не придумал детерминированного полиномиального алгоритма решения данной задачи.

Лемма Шварца-Зиппеля

Пусть есть ненулевой полином $F(x_1, ..., x_n)$. d - его степень. S - конечное случайное подмножество R^n. R - случайно выбранный элемент S. Тогда $P(F(R) = 0) \leq \frac{d}{|S|}$ (маловероятно).

Доказательство

Будем доказывать индукцией по n.

База индукции. $n = 1$

$F(x_1)$ - *полином от одной переменной. У него максимум* d *корней. Сколько бы точек мы не взяли, максимум точек будет* $d \to P(F(R) = 0) = \frac{d}{|S|}$ *и в частности* $P(F(R) = 0) \leq \frac{d}{|S|}$

Продолжение доказательства

Шаг индукции

F *можно представить в виде:*

$F(x_1, ..., x_n) = \sum_{i=1}^{d} x_1^i \cdot F_i(x_2, ..., x_n)$ *(берем* x_1 *и всюду выносим за скобки)*

$F \neq 0 \rightarrow$*(из условия)* $\exists i : F_i \neq 0$

Выберем i *- максимальное из таких чисел:* $i = max_j F_j \neq 0$.

Fi - ненулевой полином и x_1^i *тоже не нулевой.*

Выбираем случайное $R \subset S$ *и оценим вероятность того, что* $F(R) = 0$.

Продолжение доказательства

Это может быть только в 2 случаях: когда $F_i = 0$ *или когда* $x_1^i = 0$

1) $P(F_i(R) = 0) \leq \frac{d-i}{|s|}$ *(так как можно применить предположение индукции)*

2) $P(x_1^i(R) = 0) \leq \frac{i}{|s|}$

Так как выполняется либо (1) либо (2) получаем:

$P(F(R) = 0) \leq \frac{d}{|s|}$

Про BPP также известны следующие утверждения:

Замечание

$BPP \subset PP$

$BPP \subset \sum^2 P \cap \Pi^2 P$

$P \cap BPP$