Algoritmos Probabilísticos

Departamento de Ciência da Computação – IME/USP

Segundo Semestre de 2006

Cadeias de Markov e passeios aleatórios

O seguinte algoritmo é um algoritmo RP para o 2-SAT:

Algoritmo-2sat (ϕ, s)
$u \leftarrow$ atribuição arbitrária
para $j \leftarrow 1$ **até** $s(2n^2)$ **faça**
    se u satisfaz ϕ
        então devolva u
    seja C uma cláusula de ϕ não satisfeita por u
    seja x a variável correspondente a um dos literais de C
    $u(x) \leftarrow \neg u(x)$
se u satisfaz ϕ
    então devolva u
devolva "ϕ não é satisfatível"

Se n é o número de variáveis em ϕ, então o número de cláusulas em ϕ é no máximo $O(n^2)$ e cada iteração pode ser executada em tempo polinomial. Só precisamos determinar o valor de s.

Se a fórmula não é satisfatível, então o algoritmo produz a resposta correta. Se a fórmula é satisfatível, então ele pode produzir a resposta errada. Vamos limitar a probabilidade disso ocorrer.

Suponha que ϕ é satisfatível. Fixe uma atribuição v que satisfaça ϕ. Seja X_j o número de variáveis na atribuição u da iteração j que tem o mesmo valor que na atribuição v. Quando $X_j = n$, o algoritmo termina. Vamos observar como as variáveis $X_0, X_1, X_2, \ldots$ evoluem com o tempo.

Se $X_j = 0$ então $X_{j+1} = 1$, ou seja, $\mathrm{P}[X_{j+1} = 1 | X_j = 0] = 1$. Como pelo menos um dos literais de C é verdadeiro em v, se $1 \leq X_j \leq n-1$, então temos chance de pelo menos $1/2$ de alterar o valor de uma variável de maneira a casar com o valor de v para tal variável. Ou seja, $\mathrm{P}[X_{j+1} = i+1 | X_j = i] \geq 1/2$ e $\mathrm{P}[X_{j+1} = i-1 | X_j = i] \leq 1/2$. Tal processo estocástico não é necessariamente uma cadeia de Markov, mas vamos analisar a cadeia de Markov $Y_0, Y_1, Y_2, \ldots$, derivada desse processo e dada por

$$\begin{aligned}
Y_0 &= X_0 \\
\mathrm{P}[Y_{j+1} = 1 | X_j = 0] &= 1 \\
\mathrm{P}[Y_{j+1} = i+1 | X_j = i] &= 1/2 \\
\mathrm{P}[Y_{j+1} = i-1 | X_j = i] &= 1/2 \qquad \text{para todo } i \text{ e } j \geq 1.
\end{aligned}$$

Tal cadeia de Markov é uma versão pessimista de $X_0, X_1, X_2, \ldots$ no seguinte sentido. O tempo esperado para $X_0, X_1, X_2, \ldots$ atingir n é menor ou igual ao tempo esperado para $Y_0, Y_1, Y_2, \ldots$ atingir n se ambas começarem no mesmo valor. Tal cadeia de Markov modela um passeio aleatório em um caminho com vértices rotulados $0, 1, 2, \ldots, n$.

Seja h_i o tempo esperado para um passeio aleatório que comece em i chegar a n. Claro que $h_n = 0$ e $h_0 = h_1 + 1$. Seja Z_i o número de passos que um passeio aleatório leva para ir de i

até n. Com $1/2$ de probabilidade, $Z_i = 1 + Z_{i-1}$, e com $1/2$ de probabilidade, $Z_i = 1 + Z_{i+1}$. Ou seja,

$$h_i \;=\; \mathrm{E}[Z_i] \;=\; \mathrm{E}[\frac{1}{2}(1 + Z_{i-1}) + \frac{1}{2}(1 + Z_{i+1})] \;=\; 1 + \frac{1}{2}\mathrm{E}[Z_{i-1}] + \frac{1}{2}\mathrm{E}[Z_{i+1}] \;=\; 1 + \frac{h_{i-1}}{2} + \frac{h_{i+1}}{2}.$$

Exercício 1: Mostre por indução em j que $h_j = h_{j+1} + 2j + 1$, para $0 \leq j \leq n-1$. Deduza que $h_0 = n^2$.

Como $h_0 = n^2$, temos que o tempo esperado para $Y_0, Y_1, Y_2, \ldots$ atingir n é no máximo n^2. Depois de $2n^2$ passos portanto, pela desigualdade de Markov, a chance do passeio aleatório não ter atingido n é no máximo $1/2$. Podemos pensar nas $s(2n^2)$ iterações do algoritmo como s passeios aleatórios de comprimento $2n^2$. A chance de nenhum deles atingir o vértice n é no máximo $(1/2)^s$, ou seja, um deles atinge o n (o que corresponde ao algoritmo responder corretamente) com probabilidade pelo menos $1 - (1/2)^s$. Isso completa a prova de que este é um algoritmo RP para o 2-SAT.

O que acontece se adaptarmos o algoritmo acima para o 3-SAT?

Exercício 2: Descreva a cadeia de Markov $Y_0, Y_1, Y_2, \ldots$ correspondente a esta adaptação. Considere a variável Z_i como acima e o valor $h_i = \mathrm{E}[Z_i]$. Calcule, como acima, o valor dos h_i's e em particular de h_0 para este caso. O que você conclui sobre o número de iterações necessárias para que essa adaptação termine com uma probabilidade de acerto aceitável?

Um outro algoritmo para o 3-SAT

Infelizmente, como você deve ter visto no exercício 2, a adaptação direta do algoritmo do 2-SAT para o 3-SAT não dá um resultado especialmente interessante. Em particular, lembre-se que há um algoritmo óbvio para o 3-SAT que consome tempo $O(m2^n)$, onde n é o número de variáveis e m é o número de cláusulas da fórmula.

Abaixo mostramos uma outra adaptação que dá um resultado mais interessante. A idéia dessa adaptação é que, se escolhermos uma atribuição aleatoriamente, acertamos em média o valor de metade das variáveis em relação à atribuição v que satisfaz a fórmula (supondo que a fórmula é satisfatível).

Algoritmo-3sat (ϕ, s)
- **para** $j \leftarrow 1$ **até** s **faça**
  - $u \leftarrow$ atribuição escolhida aleatoriamente
  - **para** $j \leftarrow 1$ **até** $3n$ **faça**
    - **se** u satisfaz ϕ
      - **então devolva** u
    - seja C uma cláusula de ϕ não satisfeita por u
    - seja x a variável correspondente a um dos literais de C
    - $u(x) \leftarrow \neg u(x)$
  - **se** u satisfaz ϕ
    - **então devolva** u
- **devolva** "ϕ não é satisfatível"

A idéia desse algoritmo é que, se depois de $3n$ passos não chegamos a uma atribuição que satisfaça ϕ, então provavelmente estamos bem mais longe de v do que inicialmente. Ou seja, é melhor recomeçarmos o passeio aleatório de uma nova atribuição do que continuarmos da corrente, que muito provavelmente está diferente da atribuição v em bem mais do que a metade das variáveis.

Para analisar tal algoritmo, seja q a probabilidade de que, a partir da atribuição u escolhida no passo 2, o algoritmo, depois de $3n$ iterações, tenha encontrado uma atribuição que satisfaça ϕ. Seja q_j uma delimitação para a probabilidade de que esse algoritmo modificado atinja v em $3n$ iterações uma vez que a atribuição da linha 2 só não coincide com v em j variáveis.

Considere um passeio aleatório no caminho de 0 a n que mova um passo para a esquerda com probabilidade $2/3$ e um passo para a direita com probabilidade $1/3$. (Do vértice 0, ele sempre move para a direita, e do vértice n ele continua no vértice n.) Suponha que tal passeio começa no vértice j, com $1 \leq j < n$. A chance de tal caminho, depois de $j + 2k$ passos, ter se movido $j + k$ passos para a direita e k passos para a esquerda é

$$\binom{j+2k}{k}\left(\frac{2}{3}\right)^k\left(\frac{1}{3}\right)^{j+k}.$$

Então temos que

$$q_j \;\geq\; \max_{k=0,\ldots,j}\binom{j+2k}{k}\left(\frac{2}{3}\right)^k\left(\frac{1}{3}\right)^{j+k} \;\geq\; \binom{3j}{j}\left(\frac{2}{3}\right)^j\left(\frac{1}{3}\right)^{2j}.$$

Exercício 3: Use a aproximação de Stirling para deduzir que

$$\binom{3j}{j} \;\geq\; \frac{c}{\sqrt{j}}\left(\frac{27}{4}\right)^j,$$

onde c é uma constante.

Conclua da delimitação do exercício 3 que $q_j \geq \frac{c}{\sqrt{j}}\frac{1}{2^j}$. Além disso, claro que $q_0 = 1$. Dessas delimitações, podemos concluir uma delimitação para q:

$$\begin{aligned}
q \;&\geq\; \sum_{j=0}^{n} \mathrm{P}[\text{uma atribuição aleatória difere de } v \text{ em } j \text{ variáveis}] \cdot q_j \\
&\geq\; \frac{1}{2^n} + \sum_{j=0}^{n}\binom{n}{j}\left(\frac{1}{2}\right)^n \frac{c}{\sqrt{j}}\frac{1}{2^j} \\
&\geq\; \frac{c}{\sqrt{j}}\left(\frac{1}{2}\right)^n \sum_{j=0}^{n}\binom{n}{j}\left(\frac{1}{2}\right)^j (1)^{n-j} \\
&=\; \frac{c}{\sqrt{j}}\left(\frac{1}{2}\right)^n\left(\frac{3}{2}\right)^n \\
&=\; \frac{c}{\sqrt{j}}\left(\frac{3}{4}\right)^n.
\end{aligned}$$

Assumindo que ϕ é satisfatível, o número de atribuições aleatórias iniciais que o processo tem que tentar antes de chegar a uma atribuição que satisfaça ϕ é uma variável com distribuição geométrica de parâmetro q. O número esperado de tentativas é portanto $1/q$, e a cada vez o algoritmo executa $3n$ iterações. Assim o número esperado de iterações no total é $O(n^{3/2}(4/3)^n)$. Tal algoritmo Las Vegas implica em um algoritmo Monte Carlo que efetua $O(n^{3/2}(4/3)^n)$ iterações.