O Lema Local de Lovász Algorítmico

Bruno Pasqualotto Cavalar

Orientador: **Yoshiharu Kohayakawa**

IME-USP

Introdução

O Lema Local de Lovász (LLL) é uma ferramenta poderosa para provar a existência de objetos combinatórios que satisfazem um dado conjunto de restrições. Mais precisamente, dado uma coleção finita $\mathcal{A}$ de eventos "ruins", o LLL dá condições suficientes para que a probabilidade $\mathbb{P}\left[\bigcap_{A\in\mathcal{A}}\overline{A}\right]$ seja positiva. A prova original de Lovász é não construtiva, e garante apenas uma probabilidade exponencialmente pequena. Além disso, o espaço de probabilidade que consideramos é tipicamente exponencial, de modo que uma busca exaustiva não é eficiente. Ainda assim, em um celebrado artigo [6], Moser e Tardos mostraram como construir eficientemente tais objetos cuja existência é garantida pelo LLL, supondo apenas que: i) o número de eventos não é exponencialmente grande; ii) verificar (e encontrar) a ocorrência de um evento ruim é computacionalmente fácil. Posteriormente, Haeupler et al. [2] encontraram modos de lidar com esses problemas, exigindo condições ligeiramente mais fortes que o do LLL.

Esse trabalho buscou estudar aplicações clássicas do LLL em problemas de combinatória e aplicar o algoritmo de Moser-Tardos nesses problemas.

O Lema Local de Lovász

Seja $\mathcal{A}$ uma coleção finita de eventos não triviais num mesmo espaço de probabilidade. Se os eventos de $\mathcal{A}$ forem mutuamente independentes, então vale que

$$\mathbb{P}\left[\bigcap_{A\in\mathcal{A}}\overline{A}\right] = \prod_{A\in\mathcal{A}}\mathbb{P}[\overline{A}] > 0.$$

É natural supor que algo similar acontece quando os eventos são "limitadamente dependentes". A seguinte definição ajuda a formalizar esse conceito.

Definição 1. Um digrafo $D = (\mathcal{A}, E)$ é dito um **digrafo de dependência** para $\mathcal{A}$ se cada evento $A \in \mathcal{A}$ é mutuamente independente dos eventos em $\mathcal{A} \setminus (\Gamma(A) \cup \{A\})$, onde

$$\Gamma(A) := \{B \in \mathcal{A} : (A, B) \in E\}.$$

O Lema Local de Lovász é uma ferramenta poderosa que nos permite evitar todos os eventos de $\mathcal{A}$, contanto que as probabilidades dos eventos não sejam muito grandes e o grafo de dependência não tenha muitas arestas.

Lema 2 (Lema Local de Lovász [1])**.** *Seja $\mathcal{A}$ uma coleção finita de eventos e $D = (\mathcal{A}, E)$ um digrafo de dependência para $\mathcal{A}$. Se existe uma função $x : \mathcal{A} \to (0, 1)$ tal que*

$$\mathbb{P}[A] \le x(A) \prod_{B\in\Gamma(A)}(1 - x(B)) \quad \textit{para todo } A \in \mathcal{A}, \tag{1}$$

então $\mathbb{P}\left[\bigcap_{A\in\mathcal{A}}\overline{A}\right] = \prod_{A\in\mathcal{A}}(1 - x(A)) > 0$. ∎

1 Algumas aplicações

Uma coloração própria de um grafo G é dita β-frugal se cada cor aparece no máximo β vezes na vizinhança de cada vértice. Defina $F(G)$ como o menor número para o qual existe uma coloração β-frugal de G com $F(G)$ cores. Alon provou que existe uma constante c tal que para todo Δ existe um grafo G com grau máximo Δ que satisfaz $F(G) > c\Delta^{1+1/\beta}$. O seguinte teorema mostra que esse resultado é assintoticamente ótimo.

Teorema 3 (Hind, Molloy e Reed [3])**.** *Se G tem grau máximo $\Delta \ge \beta^\beta$, então G tem uma coloração β-frugal usando no máximo $16\Delta^{1+1/\beta}$ cores.*

Demonstração. Para $\beta = 1$, o resultado é equivalente a encontrar uma coloração própria de G^2, isto é, o grafo obtido de G adicionando arestas entre vértices com distância igual a 2. Como esse grafo tem grau máximo Δ^2, ele tem uma coloração com $\Delta^2 + 1 \le 16\Delta^2 = 16\Delta^{1+1/\beta}$ cores.

Suponha então que $\beta \ge 2$. Defina $C = 16\Delta^{1+1/\beta}$. Para cada vértice de G atribuímos aleatoriamente uma cor entre $\{1, \ldots, C\}$ com probabilidade uniforme. Para cada aresta uv de G definimos o evento A_{uv} de que os vértices u e v recebam a mesma cor. Chamamos tais eventos de eventos do Tipo A. Além disso, para cada conjunto $\{u_1, \ldots, u_{\beta+1}\}$ de $\beta + 1$ vértices todos na vizinhança de um mesmo vértice, definimos o evento $B_{u_1,\ldots,u_{\beta+1}}$ de que cada u_i receba a mesma cor. Chamamos tais eventos de eventos do Tipo B. Claramente, se evitamos todos os eventos do Tipo A e do Tipo B, então a nossa coloração aleatória é β-frugal.

Note que a probabilidade de cada evento do Tipo A é $1/C$, e a probabilidade de cada evento do Tipo B é $1/C^{\beta-1}$. Além disso, cada um dos eventos (Tipo A ou Tipo B) é independente de no máximo $(\beta+1)\Delta$ eventos do Tipo A e $(\beta+1)\Delta\binom{\Delta}{\beta}$ eventos do Tipo B. Feitas estas observações, não é difícil verificar que a atribuição $x(A) = 2/C$ para eventos A do Tipo A e $x(B) = 2/C^{\beta-1}$ para eventos B do Tipo B satisfaz (1). ∎

Outras aplicações

Delineamos aqui outras aplicações do LLL. Omitimos provas por questões de espaço.

1. **Coloração acíclica de arestas**. Uma coloração própria das arestas de um grafo é dita acíclica se o grafo induzido pela união de quaisquer duas classes de cores é uma floresta. Molloy e Reed [5] provaram usando o LLL que todo grafo G com grau máximo Δ tem uma coloração acíclica com no máximo 16Δ cores.
2. **Satisfatibilidade de fórmulas k-CNF**. O problema da satisfatibilidade de fórmulas k-CNF consiste em, dada uma fórmula booleana ϕ na forma normal conjuntiva, onde cada cláusula tem exatamente k literais distintos, decidir se existe uma interpretação satisfatível de ϕ. Dizemos que uma cláusula intersecta outra cláusula se elas compartilham alguma variável. Erdős e Lovász [1] provaram, usando o LLL, que toda fórmula k-CNF em que cada cláusula intersecta no máximo 2^{k-2} outras cláusulas é satisfatível. Note que isso não impõe restrições no número total de cláusulas.
3. **Roteamento de pacotes**. Considere um grafo G e um conjunto de pacotes i, a cada um dos quais é associado um par de vértices (s_i, t_i) e um caminho P_i que conecta s_i a t_i. Cada pacote i precisa percorrer o caminho P_i para chegar ao seu destino final. Assumimos um modelo síncrono, em que cada aresta pode carregar apenas um pacote por instante de tempo. Nosso objetivo é encontrar um escalonamento que minimize o tempo de todos os pacotes chegarem ao seu destino final. Como os caminhos já estão especificados, nossa única decisão no escalonamento diz respeito ao sistema de filas, isto é: quando vários pacotes querem passar por uma mesma aresta, precisamos escolher quem passa primeiro.

   Dois parâmetros são importantes no problema: a congestão c, definida como o número máximo de caminhos que passam por uma aresta qualquer, e a dilação d, definida como o comprimento maximo de um caminho qualquer. Não é difícil se convencer de que o tempo de um escalonamento ótimo é $\Omega(c + d)$. No entanto, Leighton et al. [4] demonstraram com o LLL a existência um escalonamento $O(c + d)$, usando filas de tamanho constante em cada aresta.

O algoritmo de Moser-Tardos

Para conseguir uma versão algorítmica do LLL, Moser e Tardos consideraram um cenário levemente modificado do Lema Local de Lovász, mas que ainda é válido na maior parte das aplicações conhecidas.

Seja $\mathcal{P}$ um conjunto finito de variáveis aleatórias mutuamente independentes num mesmo espaço de probabilidade. Suporemos que todo evento de $\mathcal{A}$ é determinado por um subconjunto dessas variáveis. Diremos que uma atribuição de valores para as variáveis de $\mathcal{P}$ *viola* o evento $A \in \mathcal{A}$ se essa atribuição faz com que A aconteça. Para cada evento $A \in \mathcal{A}$, denote por $\mathrm{vbl}(A)$ o conjunto minimal das variáveis de $\mathcal{P}$ que determina A. Defina também

$$\Gamma(A) := \{B \in \mathcal{A} : \mathrm{vbl}(B) \cap \mathrm{vbl}(A) \ne \emptyset\}.$$

Seja D o digrafo com conjunto de vértices $\mathcal{A}$ e tal que a vizinhança de um evento A é $\Gamma(A)$. Como A é mutuamente independente de todos os eventos em $\mathcal{A} \setminus (\Gamma(A) \cup \{A\})$, temos que D é um digrafo de dependência para $\mathcal{A}$. O celebrado algoritmo de Moser-Tardos é como segue.

Algoritmo 1: Algoritmo de Moser-Tardos

```
1 para todo P ∈ 𝒫 faça
2     v_P ← uma valoração aleatória de P;
3 enquanto ∃A ∈ 𝒜 : A é violado quando (P = v_P : ∀P ∈ 𝒫) faça
4     escolha arbitrariamente um evento violado A ∈ 𝒜;
5     para todo P ∈ vbl(A) faça
6         v_P ← uma nova valoração aleatória de P;
7 devolva (v_P)_{P∈𝒫}
```

Cada vez que um evento A é escolhido na linha 4 dizemos que ele foi *reamostrado*. Note que a eficiência do método depende de que i) o número de reamostragens não é muito grande; ii) valores aleatórios para cada variável $P \in \mathcal{P}$ podem ser eficientemente amostrados; iii) verificar (e encontrar) a ocorrência de um evento também pode ser feito eficientemente. A versão construtiva do LLL de Moser e Tardos trata do primeiro problema.

Teorema 4 (Moser e Tardos [6])**.** *Seja $\mathcal{P}$ um conjunto finito de variáveis aleatórias mutuamente independentes num mesmo espaço de probabilidade e $\mathcal{A}$ uma coleção finita de eventos determinados por essas variáveis. Se existe uma função $x : \mathcal{A} \to (0, 1)$ tal que*

$$\mathbb{P}[A] \le x(A) \prod_{B\in\Gamma(A)}(1 - x(B)) \quad \textit{para todo } A \in \mathcal{A},$$

então existe uma atribuição de valores às variáveis de $\mathcal{P}$ que não viola nenhum dos eventos de $\mathcal{A}$. Além disso, o número esperado de reamostragens do evento $A \in \mathcal{A}$ que o algoritmo aleatório acima faz é no máximo $\frac{x(A)}{1-x(A)}$. Portanto, o número total de amostragens esperado é $\sum_{A\in\mathcal{A}}\frac{x(A)}{1-x(A)}$.

O algoritmo na prática

Uma análise mais cuidadosa do algoritmo de Moser-Tardos é feita em [2]. Um dos parâmetros fundamentais da análise é

$$\delta := \min_{A\in\mathcal{A}} x(A) \prod_{B\in\Gamma(A)}(1 - x(B)).$$

Note que, nas hipóteses do LLL, vale que $\delta \ge \min_{A\in\mathcal{A}}\mathbb{P}(A)$. O primeiro resultado de [2] é o seguinte.

Teorema 5 (Haeupler, Saha e Srinivasan)**.** *Suponha que estamos nas mesmas condições do Teorema 4. Então o número total de amostragens esperado E a ser realizado pelo algoritmo satisfaz*

$$E \le n\log(1/\delta)\max_{A\in\mathcal{A}}\frac{1}{1 - x(A)}.$$

Aplicação em colorações frugais

Faremos aqui uma análise da complexidade do algoritmo de Moser-Tardos aplicado no problema de colorações frugais, que foi o tópico do Teorema 3. Seja G um grafo com n vértices tal que $\Delta(G) \ge \beta^\beta$. Primeiramente, note que é possível descobrir se uma coloração não é frugal em tempo $O(n\Delta)$. Além disso, na prova do Teorema 3 os eventos satisfazem que $\delta \ge \min_{A\in\mathcal{A}}\mathbb{P}(A) = \frac{1}{C^{\beta-1}}$. Portanto, $\log(1/\delta) \le \log(C^{\beta-1}) \le \beta\log\Delta$. Temos ainda que $\max_{A\in\mathcal{A}}\frac{1}{1-x(A)} = \frac{1}{1-2/C} \le 3$. Segue que o tempo esperado do algoritmo de Moser-Tardos nesse problema é $E = O(\beta n^2\Delta\log\Delta)$.

Referências

[1] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), Vol. II*, pages 609–627. Colloq. Math. Soc. János Bolyai, Vol. 10. North-Holland, Amsterdam, 1975.

[2] Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. New constructive aspects of the Lovász local lemma. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science FOCS 2010*, pages 397–406. IEEE Computer Soc., Los Alamitos, CA, 2010.

[3] Hugh Hind, Michael Molloy, and Bruce Reed. Colouring a graph frugally. *Combinatorica*, 17(4):469–482, 1997.

[4] Tom Leighton, Bruce Maggs, and Andréa W. Richa. Fast algorithms for finding O (congestion + dilation) packet routing schedules. *Combinatorica*, 19(3):375–401, 1999.

[5] Michael Molloy and Bruce Reed. Further algorithmic aspects of the local lemma. In *STOC '98 (Dallas, TX)*, pages 524–529. ACM, New York, 1999.

[6] Robin A. Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. *J. ACM*, 57(2):Art. 11, 15, 2010.

Agradecimentos

Durante a elaboração deste trabalho o autor recebeu auxílio financeiro da FAPESP, Processo 2015/26678-9