Jogos Gráficos

Renzo Gonzalo Gómez Diaz

2 de julho de 2013

1 Introdução

Representar um jogo de n jogadores em forma normal precisa de uma quantidade exponencial (em n) de números para especificar as utilidades de cada jogador.
Um jogo gráfico é uma representação que tenta aproveitar situações onde a utilidade de cada jogador só depende de um subconjunto dos jogadores (geralmente muito menor do que n).

Definição 1. *Um **jogo gráfico** é um par $(G, \mathcal{M})$, onde G é um grafo não-dirigido com conjunto de vértices $\{1, 2 \ldots, n\}$, e $\mathcal{M}$ é o conjunto de n matrizes M_i, que descreve a utilidade do jogador i para $i = 1, \ldots, n$.*

Neste caso, a diferença de um jogo representado em forma normal, as dimensões das matrizes M_i dependem do número de vértices adjacentes no G ao vértice que representa ao jogador i.
Além disso, vamos supor que o conjunto de estratégias S_i de cada jogador é $\{0, 1\}$ e $|M_i(\vec{x})| \leq 1$, para todo $\vec{x} \in \{0, 1\}^n$. Neste caso particular, um perfil de estratégias mistas pode-se representar como um vetor $\vec{p} \in [0, 1]^n$, tal que p_i é a probabilidade do jogador i escolher a estratégia 0 e $(1 - p_i)$ de escolher a estratégia 1. A seguir, vamos analisar os aspectos algorítmicos de este modelo.

2 Equilíbrio de Nash em jogos gráficos

Queremos agora analisar a complexidade de encontrar um equilíbrio de Nash neste tipo de representação. Vamos restringir à análise ao caso onde o grafo G é uma árvore.
Logo, denote-se por V um vértice/jogador do grafo e por v a estratégia mista que usa V. Além disso, usaremos M_V para denotar a matriz de utilidade do jogador V.

Agora, vamos descrever o algoritmo **TreeNash** que encontra um equilíbrio de Nash neste caso particular.

2.1 Algoritmo TreeNash

Em primeiro lugar, o algoritmo começa escolhendo um vértice arbitrariamente como raiz. Logo, encontra uma ordenação dos vértices de G e realiza duas fases. A primeira, que chamaremos de **fase descendente**, calcula as melhores respostas de um vértice V dada as estratégias de seus vizinhos e propaga essa informação em direção à raiz. A segunda fase, que chamaremos de **fase ascendente**, devolve um equilíbrio de Nash dada as informações calculadas na fase anterior.

Cada vértice V de G tem associado uma tabela binária $T(w, v)$, $w, v \in [0, 1]$, onde w representa a estratégia mista do pai de V e v representa a estratégia mista de V. Por enquanto vamos supor que é possível representar esta tabela de maneira finita, logo verá-se uma maneira de lidar com a sua representação. A seguir, apresentamos o algoritmo TreeNash.

Algoritmo TreeNash$(G, \mathcal{M})$

Escolher um vértice Z arbitrariamente como raiz
Ordenar os vértice de acordo com uma busca em profundidade a partir de Z
Para todo $V \in V(G)$, $T(w, v) \leftarrow 0$, $\forall v, w \in [0, 1]$
FASE-DESCENDENTE$(G, \mathcal{M})$
FASE-ASCENDENTE$(G, \mathcal{M})$

Algoritmo FASE-DESCENDENTE$(G, \mathcal{M})$

para todo $V \in V(G)$ na ordem dada pela DFS **faça**
    Seja W o pai do V (ou NIL se V é a raiz).
    se V é folha **então**
        para todo $w, v \in [0, 1]$, tal que $V = v$ é melhor resposta para $W = w$ **faça**
            $T(w, v) \leftarrow 1$
    senão
        Sejam $\vec{U} = (U_1, \ldots, U_k)$ os filhos de V; seja $T(v, u_i)$ a tabela passada por U_i para V.
        para todo $w, v \in [0, 1]$ **faça**
            para todo estratégia mista $\vec{u} = (u_1, \ldots, u_k)$ de $\vec{U}$ **faça**
                se $v = V$ é melhor resposta para $W = w, \vec{U} = \vec{u}$ **e** $T(v, u_i) = 1$ para $i = 1, \ldots, k$ **então**
                    $T(w, v) \leftarrow 1$
                    Adicione $\vec{u}$ à lista de testemunhas de $T(w, v)$
    Passar tabela $T(w, v)$ de V para W

Note que para cada entrada $T(w, v) = 1$ existirá pelo menos um vector de estratégias mistas $\vec{u} = (u_1, \ldots, u_k)$ para os filhos do vértice V tal que $T(v, u_i) = 1$. Chamaremos o vetor $\vec{u}$ de testemunha de $T(w, v)$. Além disso, denotaremos por G^V a subárvore de G que tem como raiz a V, e por $\mathcal{M}^V_{W=w}$ ao subconjunto de matrizes de utilidade de G^V tal que a matriz M_V tem uma dimensão a menos (do que em $\mathcal{M}$) fixando a estratégia $W = w$, onde W é o pai de V.

Algoritmo FASE-ASCENDENTE($G, \mathcal{M}$)

para todo $V \in V(G)$ desde a raiz (em ordem reverso ao dado pela DFS) **faça**
    Seja $\vec{U} = (U_1, \dots, U_k)$ os filhos de V (ou vazio se V for uma folha)
    se V é raiz **então**
        Escolha arbitrariamente $v \in [0, 1]$, tal que $T(v) = 1$.
    senão
        Seja W o pai de V e (w, v) os valores passados por W a V.
    Etiquete V com o valor v
    Escolha qualquer testemunha $\vec{u}$ tal que $T(w, v) = 1$ (ou $T(v) = 1$)
    Para $i = 1, \dots, k$ passe (v, u_i) de V para U_i

Proposição 1. *Seja $V \in V(G)$ e W o pai de V. Logo, $T(w, v) = 1$ se existe um equilíbrio de Nash para $(G^V, \mathcal{M}^V_{W=w})$ tal que $V = v$.*

Demonstração. Por indução na altura de G^V.
Se V é uma folha de G, então como $T(w, v) = 1$, pela condição da linha 4 da fase descendente, tem-se que $V = v$ é melhor resposta para $W = w$. Logo, G^V está em equilíbrio se $V = v$ e $W = w$.

Se V não é uma folha, sejam $\{U_1, \dots, U_k\}$ os filhos de V em G e $\vec{u} = (u_1, \dots, u_k)$ uma testemunha de $T(w, v) = 1$. Então, $T(v, u_i) = 1$ para $i = 1, \dots, k$, logo pela hipótese de indução, se V joga a estratégia mista v, existe um equilíbrio de Nash em $(G^{U_i}, \mathcal{M}^{U_i}_{V=v})$ tal que $U_i = u_i$ para $i = 1, \dots, k$. Além disso, como $V = v$ é melhor resposta para $W = w$ e $U_i = u_i$, então $(G^V, \mathcal{M}^V_{W=w})$ está em equilíbrio se $V = v$. □

Corolário 1. *Seja $(G, \mathcal{M})$ um jogo gráfico tal que G é uma árvore. Então, o algoritmo TreeNash encontra um equilíbrio de Nash de $(G, \mathcal{M})$.*

Demonstração. Segue da proposição anterior. □

2.2 Algoritmo ApproximateTreeNash

Nesta seção descreveremos uma maneira de implementar as tabelas $T(w, v)$ do algoritmo TreeNash, que permite calcular um equilíbrio de Nash aproximado para o jogo $(G, \mathcal{M})$.

Definição 2. *Um perfil de estratégias mistas $\vec{p}$ é um **ϵ-equilíbrio de Nash** se para todo jogador i e para todo $p'_i \in [0, 1]$ tem-se que $M_i(p'_i, \vec{p}_{-i}) \leq M_i(\vec{p}) + \epsilon$.*

Agora, em vez de jogar uma estratégia mista em $[0, 1]$, cada jogador estará restringido a usar uma estratégia no conjunto $\{0, \tau, 2\tau, \dots, 1\}$, onde $\tau > 0$. Logo, defina-se o algoritmo **ApproximateTreeNash** como sendo igual ao algoritmo **TreeNash** com as seguintes exceções:

a) O algoritmo tem um parâmetro adicional $\epsilon > 0$.

b) Para cada vértice V com pai W, a tabela $T(w, v)$ só terá entradas correspondentes aos múltiplos de τ.

c) Os cálculos de melhor resposta no algoritmo **TreeNash** viram a cálculos de ϵ-melhor resposta.

Então, para analisar o consumo de tempo do algoritmo **ApproximateTreeNash**, note que cada tabela tem $\frac{1}{\tau^2}$ entradas e o tempo de execução está dominado pelo laço das linhas 8-12 na FASE-DESCENDENTE do TreeNash. Como ele percorre todas as entradas das tabelas dos k filhos do vértice V, ele tem ordem $O(\frac{1}{\tau^{2k}})$. Agora, vamos analisar como a escolha de τ influi na qualidade do equilíbrio encontrado pelo algoritmo.

Lema 1. *Seja $\vec{p}, \vec{q} \in [0,1]^k$ tal que $|p_i - q_i| \leq \tau$ para $i = 1, 2, \ldots, k$. Se $\tau \leq 2/(k \lg^2(k/2))$, logo $|\prod_{i=1}^k p_i - \prod_{i=1}^k q_i| \leq (2k\lg(k))\tau$.*

Demonstração. Por indução em k.
Se $k = 2$,

$$\begin{aligned}
|p_1p_2 - q_1q_2| &= |p_1p_2 + p_2q_1 - p_2q_1 - q_1q_2|,\\
&= |p_2(p_1 - q_1) + q_1(p_2 - q_2)|,\\
&\leq |p_2||p_1 - q_1| + |q_1||p_2 - q_2|,\\
&\leq 2\tau,\\
&\leq (2k\lg k)\tau.
\end{aligned}$$

Sem perda de generalidade suponha que k é potência de 2 e $k > 2$. Logo,

$$\begin{aligned}
\prod_{i=1}^k q_i &= (\prod_{i=1}^{k/2} q_i)(\prod_{i=k/2+1}^k q_i),\\
&\leq (\prod_{i=1}^{k/2} p_i + k(\lg(k/2))\tau)(\prod_{i=k/2+1}^k p_i + k(\lg(k/2))\tau),\\
&\leq \prod_{i=1}^k p_i + (k(\lg(k/2))\tau)^2 + (\prod_{i=1}^{k/2} p_i + \prod_{i=k/2+1}^k p_i +)k(\lg(k/2))\tau),\\
&\leq \prod_{i=1}^k p_i + (k(\lg(k/2))\tau)^2 + 2k(\lg(k/2))\tau,\\
&= \prod_{i=1}^k p_i + (k(\lg(k/2))\tau)^2 + 2k(\lg(k))\tau - 2k\tau,\\
&\leq \prod_{i=1}^k p_i + 2k(\lg(k))\tau.
\end{aligned}$$

Já que $(k(\lg(k/2))\tau)^2 - 2k\tau \leq 0$ se $\tau \leq 2/(k\lg^2(k/2))$. □

Lema 2. *Sejam $\vec{p}$,$\vec{q}$ perfis de estratégias mistas para $(G, \mathcal{M})$, tal que $|p_i - q_i| \leq \tau$ para $i = 1, \ldots, k$. Se $\tau \leq 2/(k\lg^2(k/2))$, então $|M_i(\vec{p}) - M_i(\vec{q})| \leq 2^{k+1}(k\lg(k))\tau$.*

Demonstração.

$$\begin{aligned}
|M_i(\vec{p}) - M_i(\vec{q})| &\leq \sum_{\vec{x}\in\{0,1\}^k} |\prod_{j=1}^k \sigma_j(x_j) - \prod_{j=1}^k \sigma_j^*(x_j)||M_i(\vec{x})|,\\
&\leq \sum_{\vec{x}\in\{0,1\}^k} (2k\lg(k))\tau,\\
&= 2^{k+1}(k\lg(k))\tau.
\end{aligned}$$

Onde $\sigma_j(x_j) = (p_j)^{1-x_j}(1-p_j)^{x_j}$ e $\sigma_j^*(x_j) = (q_j)^{1-x_j}(1-q_j)^{x_j}$. □

Lema 3. *Seja $\vec{p}$ um equilíbrio de Nash para $(G, \mathcal{M})$, e seja $\vec{q}$ o perfil mais próximo a $\vec{p}$ na τ-grade (na norma L_1). Logo, se $\tau \leq 2/(k \lg^2(k/2))$, $\vec{q}$ é um ϵ-equilíbrio de Nash, onde $\epsilon = 2^{k+2}(k \lg(k))\tau$.*

Demonstração. Seja $q_i^* \in [0,1]$ uma melhor resposta do jogador i a $\vec{q}_{-i}$. Pelo Lema 2 tem-se que

$$\begin{array}{rrcl} & |M_i(q_i^*, \vec{q}_{-i}) - M_i(q_i^*, \vec{p}_{-i})| & \leq & 2^{k+1}(k \lg(k))\tau, \\ \Rightarrow & M_i(q_i^*, \vec{q}_{-i}) & \leq & M_i(q_i^*, \vec{p}_{-i}) + 2^{k+1}(k \lg(k))\tau. \end{array}$$

Além disso, como $\vec{p}$ é um equilíbrio, $M_i(\vec{p}) \geq M_i(q_i^*, \vec{p}_{-i})$. Logo,

$$M_i(q_i^*, \vec{q}_{-i}) \quad \leq \quad M_i(\vec{p}) + 2^{k+1}(k \lg(k))\tau \qquad \ldots (1).$$

Por outro lado, aplicando o Lema 2 aos perfis $\vec{p}$ e $\vec{q}$,

$$M_i(\vec{q}) \quad \geq \quad M_i(\vec{p}) - 2^{k+1}(k \lg(k))\tau \qquad \ldots (2).$$

Por (1) e (2), segue que $M_i(q_i^*, \vec{q}_{-i}) \leq M_i(\vec{q}) + 2^{k+2}(k \lg(k))\tau$. Então, como q_i^* é melhor resposta para $\vec{q}_{-i}$, segue que $M_i(q_i', \vec{q}_{-i}) \leq M_i(\vec{q}) + 2^{k+2}(k \lg(k))\tau$ para todo $q' \in [0,1]$. Portanto, $\vec{q}$ é um ϵ-equilíbrio de Nash. □

Corolário 2. *Para todo $\epsilon > 0$, seja $\tau \leq min(\epsilon/2^{k+2}(k \lg(k)), 2/k \lg^2(k/2))$. Então, o algoritmo ApproximateTreeNash calcula um ϵ-equilíbrio de Nash para o jogo $(G, \mathcal{M})$.*

Demonstração. Segue do lema anterior. □

Referências

[1] Michael Kearns, Michael Littman, and Satinder Singh. Graphical models for game theory. In *Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 253–260, San Francisco, CA, 2001. Morgan Kaufmann.

[2] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.