Difficulté d'approximation

P. Pansu, Université Paris-Sud

19 novembre 2011

Au menu aujourd'hui :

- Un théorème de la théorie du choix social.
- Une application à la difficulté d'approximation, en informatique théorique.

Une *fonction booléenne* est une fonction $\{-1,1\}^n \to \{-1,1\}$. On peut y penser comme à un procédé pour agréger des votes, i.e. produire une décision à partir des votes de n électeurs.

Exemple

Le i-ème dictateur *est* $Dict_i(x) = x_i$. *La* majorité *est* $Maj(x) = signe(\sum x_i)$.

Une *fonction booléenne* est une fonction $\{-1,1\}^n \to \{-1,1\}$. On peut y penser comme à un procédé pour agréger des votes, i.e. produire une décision à partir des votes de n électeurs.

Exemple

Le i-ème dictateur *est* $Dict_i(x) = x_i$. *La* majorité *est* $Maj(x) = signe(\sum x_i)$.

Un procédé d'agrégation devrait avoir les propriétés suivantes.

1. Aucun électeur ne joue de rôle prépondérant.
2. Des erreurs dans le dépouillement des votes ont peu de chance de faire basculer le résultat.

Une *fonction booléenne* est une fonction $\{-1,1\}^n \to \{-1,1\}$. On peut y penser comme à un procédé pour agréger des votes, i.e. produire une décision à partir des votes de n électeurs.

Exemple

Le i-ème dictateur *est* $Dict_i(x) = x_i$. *La* majorité *est* $Maj(x) = signe(\sum x_i)$.

Un procédé d'agrégation devrait avoir les propriétés suivantes.

1. Aucun électeur ne joue de rôle prépondérant.
2. Des erreurs dans le dépouillement des votes ont peu de chance de faire basculer le résultat.

Définition

*L'*influence $Inf_i(f)$ *du i-ème électeur sur* f *est la probabilité que, lorsque le i-ème électeur change d'avis, la valeur de* f *change.*

$$Inf_i(f) = \mathbb{P}(f(xe_i) \neq f(x)),$$

où $e_i \in \{-1,1\}^n$ *est le vecteur dont les coordonnées valent* 1 *sauf la i-ème.*

Définition

La sensibilité au c-bruit *de f est la probabilité que, lorsque chaque vote est modifié indépendamment avec probabilité c, la valeur de f change.*

$$Sens_c(f) = \mathbb{P}_{x,z}(f(xz) \neq f(x)),$$

où les coordonnées $z_i \in \{-1, 1\}$ sont i.i.d., indépendantes de x, et $\mathbb{P}(z_i = -1) = c$.

Exemple.
Pour le dictateur $Dict_i$,

$$Inf_i(Dict_i) = 1, \quad Inf_j(Dict_i) = 0 \text{ si } j \neq i \ ; \quad Sens_c(Dict_i) = c.$$

Pour la majorité, il résulte du Théorème Central Limite que

$$Inf_i(Maj) \sim \frac{2}{\sqrt{\pi n}}; \quad \lim_{n\to\infty} Sens_c(Maj) = \frac{1}{\pi} \arccos(1 - 2c).$$

De tous les procédés d'agrégation, la majorité est celui qui satisfait le mieux aux deux critères d'influence et de sensibilité ci-dessus. C'est la substance du Théorème (`Majority is stablest`).

Théorème (Mossel, O'Donnell, Oleskiewicz 2005)

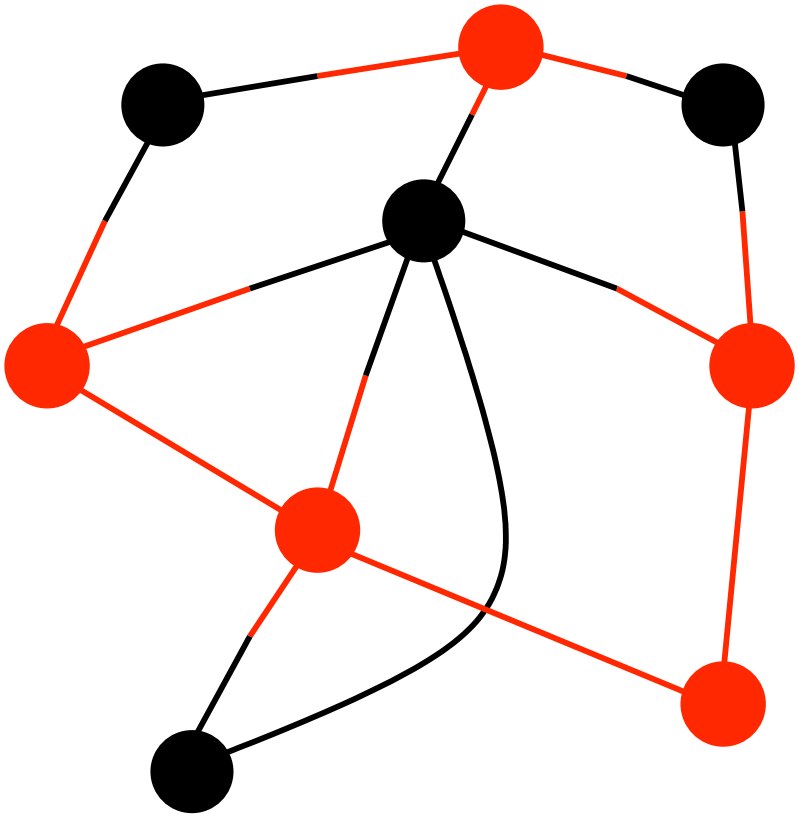
Soit $c \in [0, \frac{1}{2}]$. De toutes les fonctions booléennes $\{-1,1\}^n \to \{-1,1\}$ de moyenne nulle, dont les influences sont petites, Maj est celle dont la sensibilité au c-bruit est asymptotiquement la plus faible, lorsque n tend vers l'infini. Si $c \in [\frac{1}{2}, 1]$, Maj a la sensibilité au c-bruit la plus forte (sans condition de moyenne nulle).

En fait, l'énoncé est non asymptotique : pour tout $\epsilon > 0$, il existe $\tau(\epsilon)$ tel que si toutes les influences $Inf_i(f) < \tau$, alors $Sens_c(f) \geq \frac{1}{\pi} \arccos(1-2c) - \epsilon$ (resp. $\leq$ si $c \in [\frac{1}{2}, 1]$).

Preuve

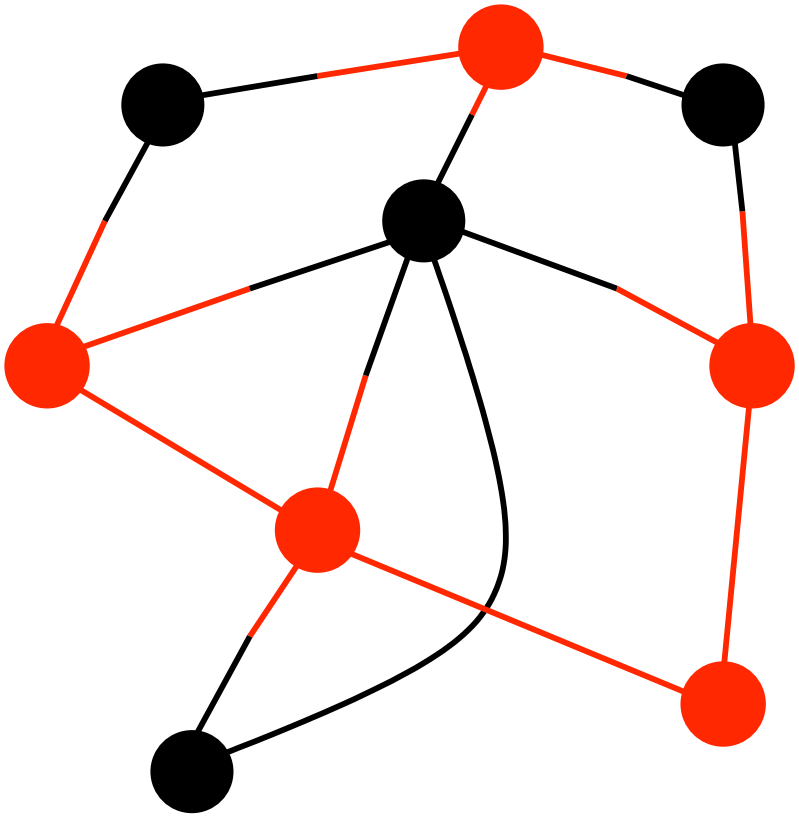
1. Principe d'invariance : on remplace le cube $\{-1,1\}^n$ muni de la mesure de probabilité uniforme, par l'espace gaussien, i.e. l'espace euclidien $\mathbb{R}^n$ muni de la mesure gaussienne γ_n, de densité $(2\pi)^{-n/2} \exp(-|x|^2/2)$.
2. Dans l'espace gaussien, un argument de symétrisation dû à Ehrhard et Borell montre que parmi les fonctions de moyenne nulle, à valeurs dans $[-1,1]$, les fonctions "signe de forme linéaire" maximisent $Sens_c$, $c \leq \frac{1}{2}$.

Ce soir, je dois recevoir 9 invités. Je dois les répartir sur deux tables. Je les connais bien, je sais quelle solide inimitié certains éprouvent pour d'autres. Par exemple, je dois absolument éviter de placer L... et M... à la même table. Et de même pour N... et P... Mais il y a trop de couples ennemis. Je vais tout de même chercher un plan de table qui maximise le nombre de couples séparés. Il y a 13 couples.



Les inimitiés constituent un graphe : ici, 9 sommets, 13 arêtes. Le plan de table consiste à colorier une partie des sommets.

Ce soir, je dois recevoir 9 invités. Je dois les répartir sur deux tables. Je les connais bien, je sais quelle solide inimitié certains éprouvent pour d'autres. Par exemple, je dois absolument éviter de placer L... et M... à la même table. Et de même pour N... et P... Mais il y a trop de couples ennemis. Je vais tout de même chercher un plan de table qui maximise le nombre de couples séparés. Il y a 13 couples.



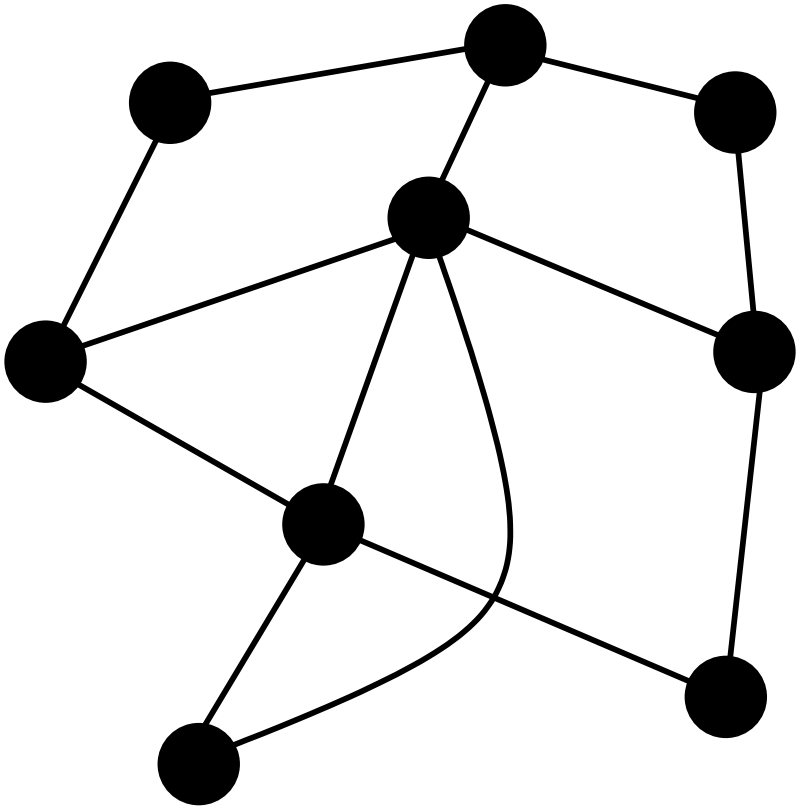
Les inimitiés constituent un graphe : ici, 9 sommets, 13 arêtes. Le plan de table consiste à colorier une partie des sommets. Ici, 9 couples ennemis séparés.

Ce plan de table a séparé 9 couples. Est ce qu'on peut faire mieux ?

Ce plan de table a séparé 9 couples. Est ce qu'on peut faire mieux ?

Question

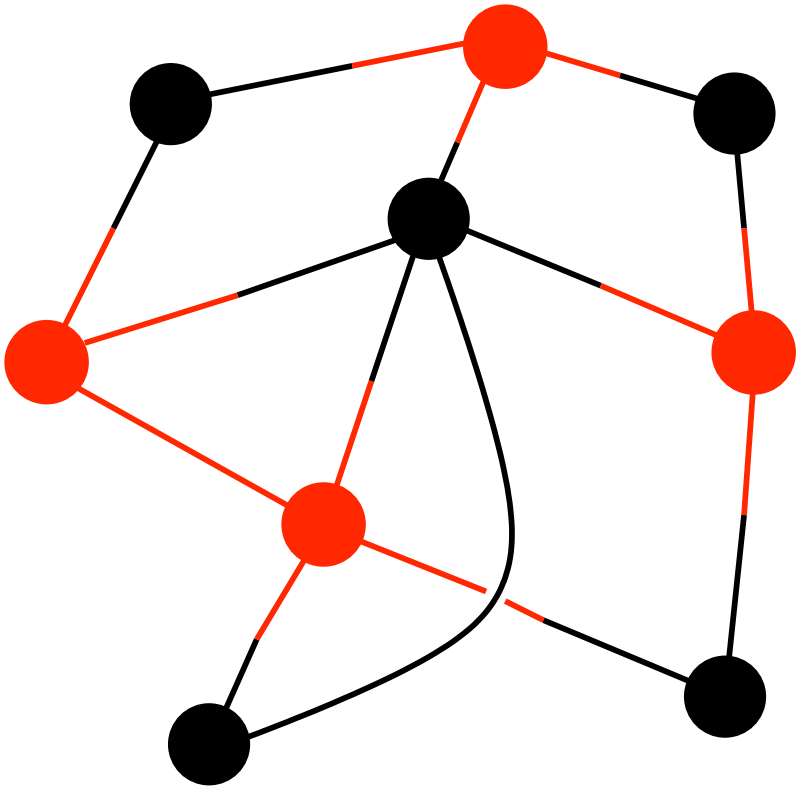
Existe-t'il un coloriage donnant plus de 9 arêtes bicolores ?



Ce plan de table a séparé 9 couples. Est ce qu'on peut faire mieux ?

Question

Existe-t'il un coloriage donnant plus de 9 arêtes bicolores ?



Oui, 11 arêtes. Mais jamais davantage.

Définition

Soit G un graphe pondéré (i.e. une distribution de probabilité sur les arêtes du graphe complet). On appelle coupe maximale(G) le maximum, sur tous les coloriages de G, de la probabilité qu'une arête soit coupée.

Le graphe étudié jusqu'à présent a une coupe maximale de $\frac{11}{13}$.

Définition

Soit G un graphe pondéré (i.e. une distribution de probabilité sur les arêtes du graphe complet). On appelle coupe maximale(G) le maximum, sur tous les coloriages de G, de la probabilité qu'une arête soit coupée.

Le graphe étudié jusqu'à présent a une coupe maximale de $\frac{11}{13}$.

Problème

MAX CUT : Ecrire un algorithme qui, étant donnés un graphe G à n sommets, et un nombre entier k, décide si la coupe maximale de G est $\geq k$ ou non.

Définition

Soit G un graphe pondéré (i.e. une distribution de probabilité sur les arêtes du graphe complet). On appelle coupe maximale(G) le maximum, sur tous les coloriages de G, de la probabilité qu'une arête soit coupée.

Le graphe étudié jusqu'à présent a une coupe maximale de $\frac{11}{13}$.

Problème

MAX CUT : Ecrire un algorithme qui, étant donnés un graphe G à n sommets, et un nombre entier k, décide si la coupe maximale de G est $\geq k$ ou non.

Définition

Un problème est dit NP-complet *s'il est NP, et si tout problème NP s'y ramène en temps polynomial.*

Théorème (Cook, Levine 1971, Karp 1972)

MAX CUT est NP-complet.

Puisqu'on ne peut problement pas résoudre exactement le problème MAX CUT, on tente de le résoudre de façon approchée.

Puisqu'on ne peut problement pas résoudre exactement le problème MAX CUT, on tente de le résoudre de façon approchée.

Définition

Soit $\alpha < 1$. Une résolution α-approchée *de MAX CUT, c'est un algorithme qui, étant donné un graphe pondéré G à n sommets, trouve en un temps polynomial en n, un coloriage dont le nombre d'arêtes bicolores est $\geq \alpha \times$ coupe maximale(G). Cet algorithme a le droit d'effectuer des tirages au hasard. On demande dans ce cas que le score annoncé soit obtenu avec probabilité $\geq \frac{1}{2}$.*

Puisqu'on ne peut problement pas résoudre exactement le problème MAX CUT, on tente de le résoudre de façon approchée.

Définition

Soit $\alpha < 1$. Une résolution α-approchée *de MAX CUT, c'est un algorithme qui, étant donné un graphe pondéré G à n sommets, trouve en un temps polynomial en n, un coloriage dont le nombre d'arêtes bicolores est $\geq \alpha \times$ coupe maximale(G). Cet algorithme a le droit d'effectuer des tirages au hasard. On demande dans ce cas que le score annoncé soit obtenu avec probabilité $\geq \frac{1}{2}$.*

Exemple

Coloriage aléatoire. On tire indépendamment au hasard la couleur de chaque sommet.

Puisqu'on ne peut problement pas résoudre exactement le problème MAX CUT, on tente de le résoudre de façon approchée.

Définition

Soit $\alpha < 1$. Une résolution α-approchée *de MAX CUT, c'est un algorithme qui, étant donné un graphe pondéré G à n sommets, trouve en un temps polynomial en n, un coloriage dont le nombre d'arêtes bicolores est $\geq \alpha \times$ coupe maximale(G). Cet algorithme a le droit d'effectuer des tirages au hasard. On demande dans ce cas que le score annoncé soit obtenu avec probabilité $\geq \frac{1}{2}$.*

Exemple

Coloriage aléatoire. On tire indépendamment au hasard la couleur de chaque sommet.

Analyse : Chaque arête a une chance sur deux d'être bicolore. L'espérance du nombre d'arêtes bicolores vaut $\frac{1}{2} \geq \frac{1}{2} \times$coupe maximale$(G)$. Par symétrie, avec probabilité $\geq \frac{1}{2}$,

$$\text{nombre d'arêtes bicolores} \geq \frac{1}{2}\text{coupe maximale}(G).$$

Donc il s'agit d'une résolution $\frac{1}{2}$-approchée de MAX CUT.

Théorème (Goemans-Williamson 1995)

Le problème MAX CUT possède une résolution 0.878...*-approchée.*

Théorème (Goemans-Williamson 1995)

Le problème MAX CUT possède une résolution 0.878...*-approchée.*

Arithmétisation. Un coloriage est une fonction booléenne $x : \{\text{sommets}\} \to \{-1, 1\}$. Il faut maximiser

$$\mathrm{OBJ} = \mathbb{E}_{\text{aretes } ij}\left(\frac{1}{2}(1 - x_i x_j)\right).$$

Théorème (Goemans-Williamson 1995)

Le problème MAX CUT possède une résolution 0.878...*-approchée.*

Arithmétisation. Un coloriage est une fonction booléenne $x : \{\text{sommets}\} \to \{-1, 1\}$. Il faut maximiser

$$\mathrm{OBJ} = \mathbb{E}_{\text{aretes } ij}(\frac{1}{2}(1 - x_i x_j)).$$

Relaxation. On maximise sur un ensemble plus grand, les applications $v : \{\text{sommets}\} \to S$, la sphère unité d'un espace euclidien de grande dimension.

$$\mathrm{SDP} = \mathbb{E}_{\text{aretes } ij}(\frac{1}{2}(1 - v_i \cdot v_j)).$$

Théorème (Goemans-Williamson 1995)

Le problème MAX CUT possède une résolution 0.878...*-approchée.*

Arithmétisation. Un coloriage est une fonction booléenne $x : \{\text{sommets}\} \to \{-1, 1\}$. Il faut maximiser

$$\mathrm{OBJ} = \mathbb{E}_{\text{aretes } ij}(\frac{1}{2}(1 - x_i x_j)).$$

Relaxation. On maximise sur un ensemble plus grand, les applications $v : \{\text{sommets}\} \to S$, la sphère unité d'un espace euclidien de grande dimension.

$$\mathrm{SDP} = \mathbb{E}_{\text{aretes } ij}(\frac{1}{2}(1 - v_i \cdot v_j)).$$

Il s'agit d'un problème de *programmation semi-définie*. max SDP peut être calculé, avec une précision arbitraire, en temps polynomial.

Théorème (Goemans-Williamson 1995)

Le problème MAX CUT possède une résolution 0.878...*-approchée.*

Arithmétisation. Un coloriage est une fonction booléenne $x : \{\text{sommets}\} \to \{-1, 1\}$. Il faut maximiser

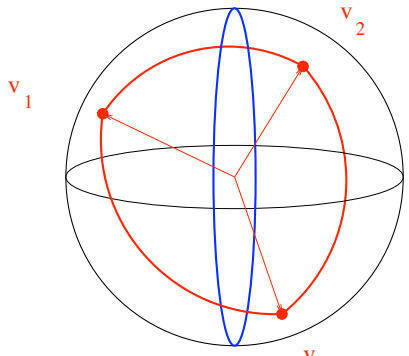
$$\mathrm{OBJ} = \mathbb{E}_{\text{aretes } ij}(\frac{1}{2}(1 - x_i x_j)).$$

Relaxation. On maximise sur un ensemble plus grand, les applications $v : \{\text{sommets}\} \to S$, la sphère unité d'un espace euclidien de grande dimension.

$$\mathrm{SDP} = \mathbb{E}_{\text{aretes } ij}(\frac{1}{2}(1 - v_i \cdot v_j)).$$

Il s'agit d'un problème de *programmation semi-définie*. max SDP peut être calculé, avec une précision arbitraire, en temps polynomial.

Procédure d'arrondi. A partir d'un plongement v du graphe dans la sphère unité S, on construit un coloriage en coupant le graphe en deux au moyen d'un plan tiré au hasard.



Analyse. La probabilité que l'arête $v_i v_j$ soit coupée (bicolore) vaut $\frac{1}{\pi} \arccos(v_i \cdot v_j)$. L'espérance du nombre d'arêtes bicolores est

$$E = \mathbb{E}_{ij}(\frac{1}{\pi} \arccos(v_i \cdot v_j)).$$

Comme pour tout $x \in [0, 1[$,

$$\frac{\frac{1}{\pi} \arccos(x)}{\frac{1}{2}(1-x)} \geq 0.878...,$$

$$E \geq 0.878... \max \mathrm{SDP} \geq 0.878... \times \max \mathrm{OBJ} = 0.878... \times \text{coupe maximale}.$$

Analyse. La probabilité que l'arête $v_i v_j$ soit coupée (bicolore) vaut $\frac{1}{\pi}\arccos(v_i \cdot v_j)$. L'espérance du nombre d'arêtes bicolores est

$$E = \mathbb{E}_{ij}(\frac{1}{\pi}\arccos(v_i \cdot v_j)).$$

Comme pour tout $x \in [0, 1[$,

$$\frac{\frac{1}{\pi}\arccos(x)}{\frac{1}{2}(1-x)} \geq 0.878...,$$

$$E \geq 0.878... \max \mathrm{SDP} \geq 0.878... \times \max \mathrm{OBJ} = 0.878... \times \text{coupe maximale}.$$

Remarque (U. Feige, G. Schechtman 2002)

Il existe des graphes pour lesquels la méthode donne une coupe arbitrairement proche de $0.878... \times$ *coupe maximale.*

Théorème (S. Khot, G. Kindler, E. Mossel, R. O'Donnell 2005)

Sous une hypothèse un peu plus forte que P≠NP, 0.878... est le seuil optimal d'approximabilité pour MAX CUT. Autrement dit, si $\alpha > 0.8785672057848516...$, il n'existe pas de résolution α-approchée de MAX CUT.

Théorème (S. Khot, G. Kindler, E. Mossel, R. O'Donnell 2005)

Sous une hypothèse un peu plus forte que P≠NP, 0.878... est le seuil optimal d'approximabilité pour MAX CUT. Autrement dit, si $\alpha > 0.8785672057848516...$, il n'existe pas de résolution α-approchée de MAX CUT.

La preuve est l'aboutissement d'idées qui remontent à Gödel (vérification de preuves, vérification probabiliste de preuves) et repose sur le théorème `Majority is Stablest`.

L'hypothèse un peu plus forte que P≠NP, c'est la Conjecture des Jeux Uniques (`Unique Games Conjecture`) de Subhash Khot.

Définition

On s'intéresse à des jeux coopératifs à deux joueurs. Des couples de questions (q, q') *sont tirés suivant une distribution connue. Les joueurs doivent donner des réponses* $r = S(q)$ *et* $r' = S(q')$ *sans savoir quelle question a été posée à l'autre. Dans un* jeu projectif, *pour chaque question* q, *chaque question* q' *et chaque réponse possible* r *du premier joueur, il y a une unique réponse* $r' = \pi_{qq'}(r)$ *du second qui les fait gagner tous les deux. Les joueurs cherchent une stratégie commune* S *qui maximise la probabilité de gain,*

$$Valeur\ du\ jeu = \max_S \mathbb{P}_{(q,q')}(S(q') = \pi_{qq'}(S(q))).$$

Définition

On s'intéresse à des jeux coopératifs à deux joueurs. Des couples de questions (q, q') sont tirés suivant une distribution connue. Les joueurs doivent donner des réponses $r = S(q)$ et $r' = S(q')$ sans savoir quelle question a été posée à l'autre. Dans un jeu projectif, *pour chaque question q, chaque question q' et chaque réponse possible r du premier joueur, il y a une unique réponse $r' = \pi_{qq'}(r)$ du second qui les fait gagner tous les deux. Les joueurs cherchent une stratégie commune S qui maximise la probabilité de gain,*

$$Valeur\ du\ jeu = \max_S \mathbb{P}_{(q,q')}(S(q') = \pi_{qq'}(S(q))).$$

Théorème

Soit n le nombre de questions et k le nombre de réponses par question. Pour tout $\epsilon > 0$, il existe k tel qu'il est NP-difficile de décider, pour un jeu projectif à k questions, dans quel cas de figure on se trouve (sachant qu'on est dans l'un des deux).

1. *La valeur du jeu est* 1.
2. *La valeur du jeu est $< \epsilon$.*

C'est une conséquence folklorique du théorème PCP (`Probabilistically Checkable Proofs`) de Arora, Safra et al. (1992), et du théorème de répétition parallèle de Ran Raz (1995).

Définition

Un jeu unique *est un jeu projectif dans les deux sens, i.e. pour chaque question q, chaque question q' et chaque réponse possible r' du second joueur, il y a aussi une unique réponse $r = \pi_{qq'}^{-1}(r')$ du premier qui les fait gagner.*

Définition

Un jeu unique *est un jeu projectif dans les deux sens, i.e. pour chaque question q, chaque question q' et chaque réponse possible r' du second joueur, il y a aussi une unique réponse $r = \pi_{qq'}^{-1}(r')$ du premier qui les fait gagner.*

Il est facile de décider si la valeur du jeu vaut 1. En effet, dans ce cas, un choix quelconque de réponse à une question détermine uniquement les réponses à toutes les autres. Il suffit de vérifier la cohérence des réponses obtenues.

Définition

Un jeu unique *est un jeu projectif dans les deux sens, i.e. pour chaque question q, chaque question q' et chaque réponse possible r' du second joueur, il y a aussi une unique réponse $r = \pi_{qq'}^{-1}(r')$ du premier qui les fait gagner.*

Il est facile de décider si la valeur du jeu vaut 1. En effet, dans ce cas, un choix quelconque de réponse à une question détermine uniquement les réponses à toutes les autres. Il suffit de vérifier la cohérence des réponses obtenues. En revanche, il semble difficile de décider si la valeur du jeu est proche de 1 ou non.

Conjecture (S. Khot 2002)

Pour tout $\epsilon > 0$ et tout $\delta > 0$, il existe k tel qu'il est NP-difficile de décider dans lequel des deux cas de figure suivants un jeu unique à k questions se trouve (sachant qu'il est dans l'un des deux).

1. *La valeur du jeu est $> 1 - \delta$.*
2. *La valeur du jeu est $< \epsilon$.*

Les opinions sont partagées sur cette conjecture. En revanche, personne ne parie sur le fait que le problème des jeux uniques est dans P.

Une *réduction* des jeux uniques à MAX CUT consiste à associer à un jeu unique J un graphe pondéré G, de sorte que si la valeur du jeu J est $> 1-\delta$ (resp. $< \epsilon$), alors la coupe maximale de G est $> c-\delta'$ (resp. $< s+\epsilon'$).

Un algorithme calculant en temps polynomial coupe maximale(G) à un facteur multiplicatif $\frac{s+\epsilon'}{c-\delta'}$ près décide si coupe maximale(G) $> c-\delta'$ ou si coupe maximale(G) $< s+\epsilon'$, donc décide si la valeur de J est $> 1-\delta$ ou $< \epsilon$. Ce qui est interdit par la conjecture des jeux uniques.

On conclut qu'il n'existe pas d'algorithme d'α-approximation de MAX CUT pour $\alpha > \frac{s}{c}$.

Une *réduction* des jeux uniques à MAX CUT consiste à associer à un jeu unique J un graphe pondéré G, de sorte que si la valeur du jeu J est $> 1-\delta$ (resp. $< \epsilon$), alors la coupe maximale de G est $> c-\delta'$ (resp. $< s+\epsilon'$).

Un algorithme calculant en temps polynomial coupe maximale(G) à un facteur multiplicatif $\frac{s+\epsilon'}{c-\delta'}$ près décide si coupe maximale(G) $> c-\delta'$ ou si coupe maximale(G) $< s+\epsilon'$, donc décide si la valeur de J est $> 1-\delta$ ou $< \epsilon$. Ce qui est interdit par la conjecture des jeux uniques.

On conclut qu'il n'existe pas d'algorithme d'α-approximation de MAX CUT pour $\alpha > \frac{s}{c}$.

Pour tout $c \in]\frac{1}{2}, 1]$, la construction donne $s = \frac{1}{\pi}\arccos(1-2c)$. Le facteur optimal d'approximation est donc

$$min_{c\in]\frac{1}{2},1]} \frac{\arccos(1-2c)}{\pi c} = 0.878567205784851 6....$$

L'idée de la réduction est de coder les stratégies par des coloriages.

Chaque réponse $r \in R$ est codée par une fonction booléenne sur $\{-1,1\}^R$, le dictateur $Dict_r$. Une stratégie S est codée par la phrase constituée des dictateurs $(Dict_{S(q)})_{q\in Q}$, puis on applique un brouillage : on change chaque bit avec probabilité $c > \frac{1}{2}$. On obtient une fonction booléenne sur $Q \times \{-1,1\}^R$.

L'idée de la réduction est de coder les stratégies par des coloriages.

Chaque réponse $r \in R$ est codée par une fonction booléenne sur $\{-1,1\}^R$, le dictateur $Dict_r$. Une stratégie S est codée par la phrase constituée des dictateurs $(Dict_{S(q)})_{q\in Q}$, puis on applique un brouillage : on change chaque bit avec probabilité $c > \frac{1}{2}$. On obtient une fonction booléenne sur $Q \times \{-1,1\}^R$.

On considère donc le graphe pondéré G dont l'ensemble des sommets est $Q \times \{-1,1\}^R$. Une distribution de probabilité sur les arêtes est obtenue (grosso modo) en tirant les paires de points (q,q') de Q au hasard (suivant la loi du jeu), en tirant un vecteur $x \in \{-1,1\}^R$ au hasard, en changeant chaque bit de $x' = \pi_{qq'}(x)$ avec probabilité c et en reliant (q,x) à (q',x').

L'idée de la réduction est de coder les stratégies par des coloriages.

Chaque réponse $r \in R$ est codée par une fonction booléenne sur $\{-1,1\}^R$, le dictateur $Dict_r$. Une stratégie S est codée par la phrase constituée des dictateurs $(Dict_{S(q)})_{q\in Q}$, puis on applique un brouillage : on change chaque bit avec probabilité $c > \frac{1}{2}$. On obtient une fonction booléenne sur $Q \times \{-1,1\}^R$.

On considère donc le graphe pondéré G dont l'ensemble des sommets est $Q \times \{-1,1\}^R$. Une distribution de probabilité sur les arêtes est obtenue (grosso modo) en tirant les paires de points (q,q') de Q au hasard (suivant la loi du jeu), en tirant un vecteur $x \in \{-1,1\}^R$ au hasard, en changeant chaque bit de $x' = \pi_{qq'}(x)$ avec probabilité c et en reliant (q,x) à (q',x').

Pour le coloriage de G codant une stratégie S, la probabilité qu'une arête soit bicolore est au moins $c\times$valeur(S).

Inversement, un coloriage de G donne une phrase $(f_q)_{q\in Q}$ qu'il s'agit de décoder. Si la probabilité qu'une arête soit bicolore est $> s+\epsilon'$, alors en moyenne, f_q a une sensibilité au c-bruit $> s+\epsilon'$. D'après le Théorème `Majority is Stablest`, f_q possède au moins une variable $r = S(q)$ de forte influence. La stratégie S obtenue a une valeur $> \epsilon$.