DIMACS Workshop on Sublinear Algorithms
Princeton, NJ
24–27 septembre 2000

Test de propriétés sur des objets combinatoires: une introduction

Jean-François Baget

LIRMM
161, rue Ada, 34392 Montpellier – FRANCE

baget@lirmm.fr

Introduction

Exposé inspiré de celui d'O. Goldreich (DIMACS workshop) et d'un tutorial de D. Ron [Ron00]

Test de propriété ("property testing"): déterminer si un objet vérifie une propriété donnée P ou si il est "loin" (très différent) de tout objet vérifiant cette propriété.

Notion introduite dans [RS96], motivée par la vérification de programmes. Relation avec *PAC-learning* [Val84]

Très souvent utilisé dans le cas de propriétés *algébriques* (monotonie, linearité, multi-linéarité, ...). Ici, vérification de propriétés *combinatoires* [RS96, GGR96].

Définitions (d'après [GGR96])

Soit Π un problème de décision (i.e. un ensemble d'instances D_Π, et $Y_\Pi \subseteq D_\Pi$ le sous ensemble pour lequel la réponse au problème de décision est OUI). Nous notons D_Π^n (resp. Y_Π^n) les instances de taille n.

Une instance $\mathcal{I} \in D_\Pi^n$ est *décrite* par une fonction $f_\mathcal{I} : [n] \to \{0,1\}^*$.

Une *requête* sur $f_\mathcal{I}$ peut être:

- soit la demande d'un *exemple* $(x, f_\mathcal{I}(x))$, x tiré uniformément avec remise sur $[n]$;
- soit la valeur de $f_\mathcal{I}(x)$, x déterminé par le programme.

Soient $\mathcal{I}, \mathcal{J} \in D_\Pi^n$ deux instances de taille n.

$\mathcal{I}$ est dite ϵ-*proche* de $\mathcal{J}$ si il suffit de changer $\epsilon \times n$ valeurs de $f_\mathcal{I}(x)$ pour que $f_\mathcal{I} = f_\mathcal{J}$.

Définitions (suite)

$\mathcal{I}$ est dite ϵ-*éloignée* de $\mathcal{J}$ si elle n'est pas ϵ-proche de $\mathcal{J}$ (i.e. aucun changement de $\epsilon \times n$ valeurs de $f_{\mathcal{I}}(x)$ n'est suffisant pour que $f_{\mathcal{I}} = f_{\mathcal{J}}$).

$\mathcal{I}$ est dite ϵ-*éloignée* de Y_{Π}^{n} si elle est ϵ-éloignée de *toute* instance $\mathcal{J} \in Y_{\Pi}^{n}$.

Un *testeur de propriété* pour Π est une machine (à oracle probabiliste) qui, pour un paramètre $\epsilon > 0$ et *un certain nombre* de requêtes sur une fonction $f_{\mathcal{I}}$, répond:

- OUI, avec une probabilité $\geq 2/3$, si $\mathcal{I} \in Y_{\Pi}^{n}$;
- NON, avec une probabilité $\geq 2/3$, si $\mathcal{I}$ est ϵ-éloignée de Y_{Π}^{n}.

Objectif: Écrire des testeurs qui fonctionnent avec un nombre de requêtes sublinéaire, ou indépendant de n.

Exemple introductif: ce livre est-il écrit en français ?

Problème: un livre de n mots est-il écrit en Français?

Pour un livre $\mathcal{I}$ de taille n, la fonction $f_{\mathcal{I}}$ décrivant ce livre est donnée par $f(q) = 1$ si le $q^{\text{ième}}$ mot du livre est en français, 0 sinon.

```
début
    pour i := 1 à N faire
        si ¬ Oracle (f_I) alors retourner NON;
    fin
    retourner OUI;
fin
```

Analyse: Si le livre ne comprend que des mots en français, le testeur répond bien OUI avec une probabilité $\geq 2/3$ (et même 1). Si $N >= -\frac{\log(3)}{\log(1-\epsilon)}$, alors le testeur répond bien NON avec une probabilité $\geq 2/3$ si plus de $\epsilon \times n$ mots ne sont pas en français dans le texte.

Remarques à propos de cet exemple

- Nombre d'appels à l'*oracle* ("query complexity"), et complexité en temps ("running time complexity"): $O(-\frac{1}{\log(1-\epsilon)})$. Ceci est *indépendant de la taille de la donnée*
- Aucune hypothèse sur la répartition de la donnée.
- "One-sided error": la machine accepte toujours une donnée correcte.
- Dans le cas d'un rejet, peut donner un *témoin* ("witness") prouvant ce rejet (souvent le cas).
- Algorithme extrêmement simple à implémenter.
- Si on veut une probabilité de rejet $\geq \delta$, dans le cas où plus de $\epsilon \times n$ mots ne sont pas en français dans le texte, il suffit de prendre $N = \frac{\log(1-\delta)}{\log(1-\epsilon)}$ (pour $\epsilon = 0.01$ et $\delta = 0.99$, $N = 459$).

Plan de l'exposé

- Motivation, champs d'application
- Test de propriétés sur des graphes
  - Plusieurs représentations pour f
  - Quelques résultats
  - Exemple: BIPARTI
- Que peut-on tester ?
  - Généralisation: PARTITION GÉNÉRALISÉE
  - Propriétés du premier ordre

Motivation

Paradigme pouvant être utilisé dans plusieurs cas de figure:

- Lancer le "testeur de propriété" avant l'algorithme de décision. Si il rejette et fournit un "témoin", l'algorithme de décision n'est plus nécessaire.
- Ceci peut être très efficace si on a la *garantie* que les instances sont soit "bonnes", soit "très mauvaises".
- Dans certains cas, savoir qu'une propriété est "presque vérifiée" est suffisant.
- La donnée peut être trop grande pour pouvoir être lue en entier ...
- L'approximation peut être une alternative dans le cas de problèmes NP-difficiles.

Test de propriétés de graphes

Pourquoi les graphes ?

- intérêt au LIRMM pour les "gros" graphes (graphe du web)
  "Ce sujet devient à la mode et semble très intéressant."
  M. Habib – mail ifa/arc, 27 Oct 2000
- property testing: souvent des algorithmes dont la complexité ne dépend pas de la taille du graphe
- une procédure déterministe pour décider une propriété monotone non triviale sur les graphes doit examiner au moins $\Omega(n^2)$ entrées dans sa matrice d'adjacence (conjecture Aanderaa-Rosenberg [Ros73], prouvée [RV76])
- $\Omega(n^{4/3})$ dans le cas des algorithmes randomisés [Yao87, Kin91, Haj91]

Représentation par Matrice d'Adjacence [GGR96]

- Le graphe $\mathcal{I}$ (de taille k) est représenté par sa matrice d'adjacence $M_{\mathcal{I}}$
- La taille du problème est $n = k^2$
- $f_{\mathcal{I}}(x) = 1$ si et seulement si $M_{\mathcal{I}}[x/n][x\%n] = 1$
- $\mathcal{I}$ est ϵ-proche de $\mathcal{J}$ ssi il suffit de changer $\epsilon/2 \times k^2$ entrées dans la matrice $M_{\mathcal{I}}$ pour la rendre identique à $M_{\mathcal{J}}$

La représentation par matrice d'adjacence est appropriée dans le cas de graphes *denses*, et d'ailleurs les résulats présentés ici reposent sur cette hypothèse.

Représentation par Liste d'Incidence [GR97]

- Représentation d'un graphe $\mathcal{I}$ de taille k et de *degré borné* par d par une matrice de taille $k \times d$. $M_{\mathcal{I}}[v][i]$ contient l'identifiant du $i^{\text{ième}}$ voisin du sommet v, 0 si il n'y en a pas.
- La taille du problème est $n = k \times d$
- $f_{\mathcal{I}}(x) = M_{\mathcal{I}}[x/d][x\%d]$
- $\mathcal{I}$ est ϵ-proche de $\mathcal{J}$ ssi il suffit de changer $\epsilon/2 \times d \times k$ entrées dans la matrice $M_{\mathcal{I}}$ pour la rendre identique à $M_{\mathcal{J}}$

Ici, il n'y a plus d'hypothèse sur la densité du graphe. Mais le *degré borné* peut sembler une grosse limitation. Dans ce cas, voir [PR99] pour des listes d'incidence de longueur variable.

- Distance exprimée en nombre d'arêtes, représentation non fonctionelle
- Les tests sont au moins aussi difficiles que dans le cas "degré borné"
- Pour certains problèmes, ils sont strictement plus difficiles

Quelques Résultats ...

Problème	Matrice adjacence	Liste bornée	Liste non bornée
BIPARTI	$\mathcal{O}(\epsilon^{-3})$ [GGR96] $\mathcal{O}(\epsilon^{-2})$ [AK99]	LB: $\Omega(\sqrt{N})$ [GR97] $\mathcal{O}(\sqrt{N}\text{pol}(\epsilon^{-1}\log(N))$ [GR99]	
k-COLORABLE	$\mathcal{O}(exp(k^4\epsilon^{-6}))$ [GGR96] $\mathcal{O}(exp(k\epsilon^{-2}))$ [AK99]		
ρ-CLIQUE	$\mathcal{O}(exp(\rho\epsilon^{-2}))$ [GGR96]		
ρ-BISECTION	$\mathcal{O}(exp(\epsilon^{-3}))$ [GGR96]		
CONNEXE		$\mathcal{O}(\epsilon^{-1})$ [GR97]	$\mathcal{O}(\epsilon^{-2})$ [Ron00]
k-CONNEXE		$\mathcal{O}(k^3\epsilon^{-3+2/k})$ [GR97]	$\mathcal{O}(k^3\epsilon^{-4+2/k})$ [Ron00]
EULERIEN		$\mathcal{O}(\epsilon^{-1})$ [GR97]	$\mathcal{O}(\epsilon^{-2})$ [Ron00]
SANS CYCLE		$\mathcal{O}(\epsilon^{-3}(1+d\epsilon))$ [GR97] *orienté*, LB: $\Omega(\sqrt[3]{N})$ [BR00]	LB: $\Omega(\sqrt{N})$ [PR99]
DIAMETRE $\mathcal{D}$			ϵ-far de $[\mathcal{D}+4; 4\mathcal{D}+2]$ $\mathcal{O}(\epsilon^{-3})$ [PR99]

BIPARTI [GGR96]

Définition: Un graphe $G = (V, E)$ est dit *biparti* si il existe une partition $V = V_1 \oplus V_2$ telle qu'il n'existe pas d'arête entre des sommets appartenant à la même partition.

Notations: Soit $V = V_1 \oplus V_2$ une partition. $xy \in E$ est *mauvaise* si x et y sont dans la même partition. La partition (V_1, V_2) est ϵ-*bonne* si E contient au plus ϵN^2 mauvaises arêtes (elle est ϵ-*mauvaise* sinon).

Algorithme: Sélectionner $\Theta(\frac{\log(1/\epsilon)}{\epsilon^2})$ sommets de G, uniformément, avec remise. Soit G' le sous-graphe de G induit par ces sommets. Si G' est biparti, répondre OUI, sinon, répondre NON.

Théorème: *Cet algorithme est bien un testeur de propriété pour* BIPARTI, *avec "one-sided error", et il fournit un témoin dans le cas d'un rejet.*

BIPARTI: **Analyse [GGR96]**

- Prouver que si G est biparti, alors l'algorithme répond OUI: *évident*
- Prouver que si G est ϵ-éloigné de n'importe quel graphe biparti de taille n, alors l'algorithme répond NON avec une probabilité $\geq 2/3$.

Supposons G ϵ-éloigné de BIPARTI (*i.e.* n'importe quelle partition des sommets de G est ϵ-mauvaise).

Un sommet est dit *influent* si son degré dans G est $\geq \epsilon/4N$ (voir qu'il y a au plus $\epsilon N^2/4$ arêtes dans G, incidentes à des sommets non influents, qui sont mauvaises pour une partition quelconque).

Ne tirons pour l'instant qu'un graphe U de taille $\Theta(\frac{\log(1/\epsilon)}{\epsilon})$ (une fraction ϵ de ce dont on aura besoin au total).

Lemme 1: Avec une probabilité $\geq 5/6$, tous les sommets influents de G sauf $\epsilon N/4$ ont un voisin dans U. (*Notons C ces sommets, R les autres*)

BIPARTI: **Analyse (suite)**

Fixons une partition quelconque (U_1, U_2) de U. Ceci induit une partition (C_1, C_2) de C: C_2 contient tous les sommets de C qui ont un voisin dans U_1.

Quelle que soit la partition (R_1, R_2) de R telle que $U_i \cap R \subseteq R_i$, $(C_1 \cup R_1, C_2 \cup R_2)$ est une partition ϵ-mauvaise de G. Où sont ses ϵN^2 mauvaises arêtes ? Voir (ceci découle du lemme 1), qu'il ne peut y en avoir que la moitié incidentes à R (avec probabilité $\geq 5/6$). Donc il y en a au moins $\epsilon N^2/2$ qui ne sont incidentes qu'à des sommets de C (i.e. entre sommets de C_1 ou sommets de C_2): ces arêtes sont mauvaises pour la partition (C_1, C_2).

Lemme 2: Si on tire maintenant un graphe S de taille assez grande $\Theta(|U|/\epsilon)$ parmi les sommets de G, alors avec une probabilité $\geq 1/6(1 - 2^{-|U|})$, pour chaque partition (S_1, S_2) de S, il existe une arête dans G' qui est mauvaise pour la partition $(U_1 \cup S_1, U_2 \cup S_2)$.

BIPARTI: **Analyse (suite et fin)**

Preuve du lemme 2: Pour chaque paire de sommets dans S, la probabilité qu'elle soit mauvaise pour (C_1, C_2) est d'au moins $\epsilon/2$. Il n'y a donc aucune de ces mauvaises arêtes dans S qu'avec au plus une probabilité $2^{-|U|}/6$. Si on en trouve une, alors on ne peut pas partitionner S en (S_1, S_2) de façon à ce que $(U_1 \cup S_1, U_2 \cup S_2)$ n'admette aucune mauvaise arête.

En effet, soit xy une telle mauvaise arête (supposons que x et y soient dans C_2). Pour qu'elle ne soit pas mauvaise pour la partition $(U_1 \cup S_1, U_2 \cup S_2)$, il faudrait que x soit dans S_1 et y dans S_2, comme x et y ont tous deux un voisin dans U_1, c'est impossible.

Pour conclure: Il y a $2^{|U|}$ partitions de U, et pour chacune la probabilité du lemme 2 intervient. Donc avec une probabilité $\geq 5/6$, pour chaque partition (U_1, U_2) de U, pour chaque partition (S_1, S_2) de S, l'échantillon contient une arête mauvaise pour $(U_1 \cup S_1, U_2 \cup S_2)$. Donc $U \cup S$ n'est pas biparti.

PARTITION GÉNÉRALISÉE [GGR98]

Problème: Soit G un graphe, k un entier, et $\{\rho_i^L\}_{i=1}^k$, $\{\rho_i^U\}_{i=1}^k$, $\{\rho_{ij}^L\}_{i,j=1}^k$ et $\{\rho_{ij}^U\}_{i,j=1}^k$ 4 ensembles de nombres positifs ou nuls.

Existe-t-il une partition des sommets de G en k ensembles $(V_1, \ldots, V_k)$ tq:

- $\forall j,\ \rho_j^L|G| \leq |V_j| \leq \rho_j^U|G|$
- $\forall i,j\ \rho_{ij}^L|G|^2 \leq |E(V_i, V_j)| \leq \rho_{ij}^U|G|^2$

Applications: En choisissant les bonnes valeurs pour k et ρ, on peut obtenir les problèmes BIPARTI, k-COLORABLE, ρ-CLIQUE ou ρ-BISECTION.

Coût de la généralisation: (Modèle matrices d'adjacence) Algorithme en $\mathcal{O}(exp^{k+1}(k^2\epsilon^{-1}))$

Propriétés du premier ordre [AFKS99]

Propriété du premier ordre: Expression de la forme

$$T_1 x_1 \dots T_t x_t \ \mathcal{A}(x_1, \dots, x_t)$$

où les T_i sont des quantificateurs $\forall$ ou $\exists$, et $\mathcal{A}$ est une expression sans quantificateurs dont les relations sont l'égalité ou la présence d'une arête entre deux sommets, et construite avec les opérateurs booléens classiques.

Théorème 1 [AFKS99] Une propriété pouvant être exprimée par une formule $\exists^*\forall^*$ est testable en un temps indépendant de la taille du graphe (modèle Matrice d'adjacence).

Complexité: (# de requêtes) $\mathcal{O}(\text{tower}(\text{tower}(\text{poly}(\epsilon^{-1}, \#(\forall) + \#(\exists)))))$
Coût exorbitant de la généralité !!!

Théorème 2 [AFKS99] Il existe une propriété de la forme $\forall^+\exists^+$ qui n'est pas testable en un temps indépendant de la taille du graphe (un dérivé d'isomorphisme de sous-graphe, testable en $\Omega(\sqrt{N})$)

References

[AFKS99] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient Testing of Large Graphs. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 645–655, 1999.

[AK99] N. Alon and M. Krivelevich. Testing k-Colorability. Manuscript, 1999.

[BR00] M. Bender and D. Ron. Testing Acyclicity of Directed Graphs in Sublinear Time. In *Proceedings of ICALP*, pages 809–820, 2000.

[GGR96] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property Testing and Its Connection to Learning and Approximation. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October 1996*, pages 339–348, 1996.

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron. Property Testing and its Connection to Learning and Approximation. *JACM*, 45(4):653–750, 1998.

[GR97] Oded Goldreich and Dana Ron. Property Testing in Bounded Degree Graphs. In *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing*, pages 406–415, 1997.

[GR99] Oded Goldreich and Dana Ron. A Sublinear Bipartite Tester for Bounded Degree Graphs. *Combinatorica*, 19(3):335–373, 1999.

[Haj91] P. Hajnal. An $\Omega(n^{4/3})$ Lower Bound on the Randomized Complexity of Graph Properties. *Combinatorica*, 11(2):131–144, 1991.

[Kin91] V. King. An $\Omega(n^{5/4})$ Lower Bound on the Randomized Complexity of Graph Properties. *Combinatorica*, 11(1):23–32, 1991.

[PR99] M. Parnas and D. Ron. "testing the Diameter of Graphs. In

Proceedings of RANDOM, pages 85–96, 1999.

[Ron00] Dana Ron. Property Testing: a Tutorial. To Appear: Handbook on Randomization, 2000.

[Ros73] A. L. Rosenberg. On the Time Required to Recognize Properies of Graphs: a Problem. *SIGACT News*, 5:15–16, 1973.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust Characterization of Polynomials with Applications to Program Testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[RV76] R. L. Rivest and J. Vuillemin. On Recognizing Graphs Properties from Adjacency Matrices. *Theoretical Computer Science*, 3:371–384, 1976.

[Val84] L.G. Valiant. A Theory of the Learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Yao87] A. C. C. Yao. Lower Bounds to Randomized Algorithms for Graph Properties. In *Proceedings of the 28th Annual Symposium*

on Foundations of Computer Science, pages 393–400, 1987.