Performance of Local Algorithms in Random Structures. Power and limitations

David Gamarnik

MIT

Sub-Linear Algorithms Bootcamp

June, 2018

Outline

Part I. Challenges

Part II. Solution Space Geometry

Part III. Local Algorithms

Part I. Challenges

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

- Consider $\mathbb{G}(n, p)$.

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

- Consider $\mathbb{G}(n, p)$.
- The largest clique (fully connected subgraph) is $2(1 + o(1)) \log_{\frac{1}{p}} n$.

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

- Consider $\mathbb{G}(n, p)$.
- The largest clique (fully connected subgraph) is $2(1 + o(1)) \log_{\frac{1}{p}} n$.
- A trivial greedy algorithm finds a clique of size $(1 + o(1)) \log_{\frac{1}{p}} n$.

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

- Consider $\mathbb{G}(n, p)$.
- The largest clique (fully connected subgraph) is $2(1 + o(1)) \log_{\frac{1}{p}} n$.
- A trivial greedy algorithm finds a clique of size $(1 + o(1)) \log_{\frac{1}{p}} n$.
- Karp [1976] Find a better algorithm.

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

- Consider $\mathbb{G}(n, p)$.
- The largest clique (fully connected subgraph) is $2(1 + o(1)) \log_{\frac{1}{p}} n$.
- A trivial greedy algorithm finds a clique of size $(1 + o(1)) \log_{\frac{1}{p}} n$.
- Karp [1976] Find a better algorithm.
- Still open. This is embarrassing...

More Examples: Ind Sets in Sparse Random Graphs

More Examples: Ind Sets in Sparse Random Graphs

- Erdös-Rényi $\mathbb{G}(n, \frac{d}{n})$ or random d-regular graph $\mathbb{G}_d(n)$.

More Examples: Ind Sets in Sparse Random Graphs

- Erdös-Rényi $\mathbb{G}(n, \frac{d}{n})$ or random d-regular graph $\mathbb{G}_d(n)$.
- The largest independent set (a largest subset of nodes with no edges in between) is

$$\frac{2(1 + o_d(1)) \log d}{d} n,$$

Frieze, Luczak [1992]

More Examples: Ind Sets in Sparse Random Graphs

- Erdös-Rényi $\mathbb{G}(n, \frac{d}{n})$ or random d-regular graph $\mathbb{G}_d(n)$.
- The largest independent set (a largest subset of nodes with no edges in between) is

$$\frac{2(1 + o_d(1)) \log d}{d} n,$$

Frieze, Luczak [1992]

- Greedy algorithm finds an independent set of size

$$\frac{(1 + o_d(1)) \log d}{d} n.$$

More Examples: Ind Sets in Sparse Random Graphs

- Erdös-Rényi $\mathbb{G}(n, \frac{d}{n})$ or random d-regular graph $\mathbb{G}_d(n)$.
- The largest independent set (a largest subset of nodes with no edges in between) is

$$\frac{2(1 + o_d(1)) \log d}{d} n,$$

Frieze, Luczak [1992]

- Greedy algorithm finds an independent set of size

$$\frac{(1 + o_d(1)) \log d}{d} n.$$

- Nothing better is known.

More Examples: Coloring of Sparse Random Graphs

More Examples: Coloring of Sparse Random Graphs

- Erdös-Rényi $\mathbb{G}(n, \frac{d}{n})$ or random d-regular graph $\mathbb{G}_d(n)$.

More Examples: Coloring of Sparse Random Graphs

- Erdös-Rényi $\mathbb{G}(n, \frac{d}{n})$ or random d-regular graph $\mathbb{G}_d(n)$.
- Chromatic number is $\chi(\mathbb{G}) = \frac{(1+o_d(1))d}{2\log d}$.

More Examples: Coloring of Sparse Random Graphs

- Erdös-Rényi $\mathbb{G}(n, \frac{d}{n})$ or random d-regular graph $\mathbb{G}_d(n)$.
- Chromatic number is $\chi(\mathbb{G}) = \frac{(1+o_d(1))d}{2\log d}$.
- Greedy can color with twice as many colors $\frac{(1+o_d(1))d}{\log d}$.

More Examples: Coloring of Sparse Random Graphs

- Erdös-Rényi $\mathbb{G}(n, \frac{d}{n})$ or random d-regular graph $\mathbb{G}_d(n)$.
- Chromatic number is $\chi(\mathbb{G}) = \frac{(1+o_d(1))d}{2\log d}$.
- Greedy can color with twice as many colors $\frac{(1+o_d(1))d}{\log d}$.
- Nothing better is known.

More Examples: Random K-SAT

More Examples: Random K-SAT

- Generate an instance $\Phi(n, dn)$ of the K-SAT problem u.a.r. on n boolean $0, 1$ variables and dn clauses.

$$(x_{1,1} \vee \cdots \vee \bar{x}_{1,K}) \wedge (\bar{x}_{2,1} \vee \cdots \bar{\vee} x_{2,K}) \wedge \cdots \wedge (\bar{x}_{dn,1} \vee \cdots \vee x_{dn,K})$$

More Examples: Random K-SAT

- Generate an instance $\Phi(n, dn)$ of the K-SAT problem u.a.r. on n boolean $0, 1$ variables and dn clauses.

$$(x_{1,1} \vee \cdots \vee \bar{x}_{1,K}) \wedge (\bar{x}_{2,1} \vee \cdots \bar{\vee} x_{2,K}) \wedge \cdots \wedge (\bar{x}_{dn,1} \vee \cdots \vee x_{dn,K})$$

- Satisfiable iff

$$d \leq (1 + o_K(1)2^K \log 2 \equiv d^*(K)$$

Achlioptas & Moore [2002], Achlioptas & Peres [2003], Coja-Oghlan [2014], Ding, Sly & Sun [2015]

More Examples: Random K-SAT

- Generate an instance $\Phi(n, dn)$ of the K-SAT problem u.a.r. on n boolean $0, 1$ variables and dn clauses.

$$(x_{1,1} \vee \cdots \vee \bar{x}_{1,K}) \wedge (\bar{x}_{2,1} \vee \cdots \bar{\vee} x_{2,K}) \wedge \cdots \wedge (\bar{x}_{dn,1} \vee \cdots \vee x_{dn,K})$$

- Satisfiable iff

$$d \leq (1 + o_K(1)2^K \log 2 \equiv d^*(K)$$

Achlioptas & Moore [2002], Achlioptas & Peres [2003], Coja-Oghlan [2014], Ding, Sly & Sun [2015]

- Algorithmically can find a satisfying assignment only when

$$d \leq \frac{\log K}{K} d^*(K).$$

Coja-Oghlan [2011]

More Examples: Random K-SAT

- Generate an instance $\Phi(n, dn)$ of the K-SAT problem u.a.r. on n boolean $0, 1$ variables and dn clauses.

$$(x_{1,1} \vee \cdots \vee \bar{x}_{1,K}) \wedge (\bar{x}_{2,1} \vee \cdots \bar{\vee} x_{2,K}) \wedge \cdots \wedge (\bar{x}_{dn,1} \vee \cdots \vee x_{dn,K})$$

- Satisfiable iff

$$d \leq (1 + o_K(1)2^K \log 2 \equiv d^*(K)$$

Achlioptas & Moore [2002], Achlioptas & Peres [2003], Coja-Oghlan [2014], Ding, Sly & Sun [2015]

- Algorithmically can find a satisfying assignment only when

$$d \leq \frac{\log K}{K} d^*(K).$$

Coja-Oghlan [2011]

- Nothing better is known.

Examples in Statistics. Largest submatrix problem

Examples in Statistics. Largest submatrix problem

$$C_n = \begin{bmatrix} C_{11} & \dots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \dots & C_{nn} \end{bmatrix}, \qquad C_{i,j} \stackrel{d}{=} \text{i.i.d. } N(0,1)$$

Examples in Statistics. Largest submatrix problem

$$C_n = \begin{bmatrix} C_{11} & \dots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \dots & C_{nn} \end{bmatrix}, \qquad C_{i,j} \overset{d}{=} \text{i.i.d. } N(0,1)$$

Given k, find

$$\max_{I,J \subset [n], |I|=|J|=k} \text{Ave}(C_{I,J}).$$

Examples in Statistics. Largest submatrix problem

$$C_n = \begin{bmatrix} C_{11} & \dots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \dots & C_{nn} \end{bmatrix}, \qquad C_{i,j} \stackrel{d}{=} \text{i.i.d. } N(0,1)$$

Given k, find

$$\max_{I,J \subset [n], |I|=|J|=k} \text{Ave}(C_{I,J}).$$

Bhamidi, Dey & Nobel [2017]. Largest value is

$$\max \text{Ave}(C_{I,J}) = 2(1+o(1))\sqrt{\frac{\log n}{k}}.$$

Examples in Statistics. Largest submatrix problem

$$C_n = \begin{bmatrix} C_{11} & \dots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \dots & C_{nn} \end{bmatrix}, \qquad C_{i,j} \stackrel{d}{=} \text{i.i.d. } N(0,1)$$

Given k, find

$$\max_{I,J \subset [n], |I|=|J|=k} \text{Ave}(C_{I,J}).$$

Bhamidi, Dey & Nobel [2017]. Largest value is

$$\max \text{Ave}(C_{I,J}) = 2(1+o(1))\sqrt{\frac{\log n}{k}}.$$

Algorithmically (G & Li [2017])

$$\text{Ave}_{ALG}(C_{I,J}) = \frac{4(1+o(1))}{3\sqrt{2}}\sqrt{\frac{\log n}{k}}.$$

Examples in Statistics. Sparse Regression

Examples in Statistics. Sparse Regression

$X \in \mathbb{R}^{n \times p}$, $W \in \mathbb{R}^n$ i.i.d. $N(0, \sigma^2)$.

Examples in Statistics. Sparse Regression

$X \in \mathbb{R}^{n \times p}$, $W \in \mathbb{R}^n$ i.i.d. $N(0, \sigma^2)$.

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{np} \end{bmatrix} \begin{bmatrix} \beta_1^* \\ \vdots \\ \beta_p^* \end{bmatrix} + \begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix}.$$

Examples in Statistics. Sparse Regression

$X \in \mathbb{R}^{n \times p}$, $W \in \mathbb{R}^n$ i.i.d. $N(0, \sigma^2)$.

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{np} \end{bmatrix} \begin{bmatrix} \beta_1^* \\ \vdots \\ \beta_p^* \end{bmatrix} + \begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix}.$$

Goal: Recover β^* from observed X and Y when $\|\beta^*\|_0 \le k$ by solving

$$\min_{\|\beta\|_0 \le k} \|Y - X\beta\|_2.$$

Examples in Statistics. Sparse Regression

$X \in \mathbb{R}^{n \times p}$, $W \in \mathbb{R}^n$ i.i.d. $N(0, \sigma^2)$.

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{np} \end{bmatrix} \begin{bmatrix} \beta_1^* \\ \vdots \\ \beta_p^* \end{bmatrix} + \begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix}.$$

Goal: Recover β^* from observed X and Y when $\|\beta^*\|_0 \leq k$ by solving

$$\min_{\|\beta\|_0 \leq k} \|Y - X\beta\|_2.$$

Brute force works iff $n \geq n^*_{\text{Info}} \triangleq \Omega\left(\frac{k \log p}{\log k}\right)$. Wang, Wainwright & Ramchandran [2010], Rad [2011], G & Zadik [2017]

Examples in Statistics. Sparse Regression

$X \in \mathbb{R}^{n\times p}$, $W \in \mathbb{R}^n$ i.i.d. $N(0,\sigma^2)$.

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{np} \end{bmatrix} \begin{bmatrix} \beta_1^* \\ \vdots \\ \beta_p^* \end{bmatrix} + \begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix}.$$

Goal: Recover β^* from observed X and Y when $\|\beta^*\|_0 \le k$ by solving

$$\min_{\|\beta\|_0 \le k} \|Y - X\beta\|_2.$$

Brute force works iff $n \ge n^*_{\text{Info}} \triangleq \Omega\left(\frac{k \log p}{\log k}\right)$. Wang, Wainwright & Ramchandran [2010], Rad [2011], G & Zadik [2017]
Efficiently (convex optimization) only when $n \ge n^*_{\text{Convex}} = \Omega\left(k \log p\right)$.
Donoho, Tanner, Wainwright, Hastie, Tibshirani, Candes, Tao [1996–]

Part II. Geometry of the solution space

Overlap Gap Property

Overlap Gap Property

Optimization with random input $\mathcal{X}$

$$\min_{\theta \in \Theta} \mathcal{L}(\theta, \mathcal{X}).$$

Overlap Gap Property

Optimization with random input $\mathcal{X}$

$$\min_{\theta \in \Theta} \mathcal{L}(\theta, \mathcal{X}).$$

OGP holds if there exists $R > 0$, such that the set

$$\{\theta : \mathcal{L}(\theta, \mathcal{X}) \leq \min_{\theta \in \Theta} \mathcal{L}(\theta, \mathcal{X}) + R\}$$

is disconnected.

Overlap Gap Property

Optimization with random input $\mathcal{X}$

$$\min_{\theta \in \Theta} \mathcal{L}(\theta, \mathcal{X}).$$

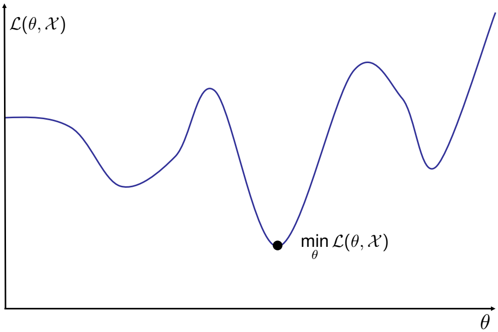
OGP holds if there exists $R > 0$, such that the set

$$\{\theta : \mathcal{L}(\theta, \mathcal{X}) \leq \min_{\theta \in \Theta} \mathcal{L}(\theta, \mathcal{X}) + R\}$$

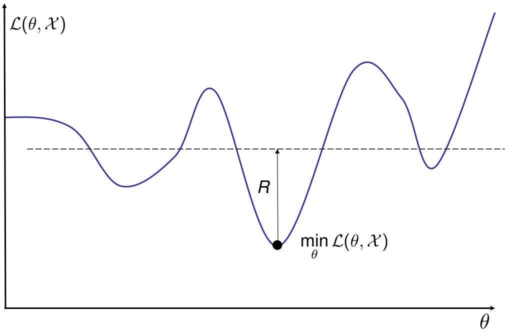
is disconnected.

The set of R-optimal solutions is partitioned into separated disconnected components.

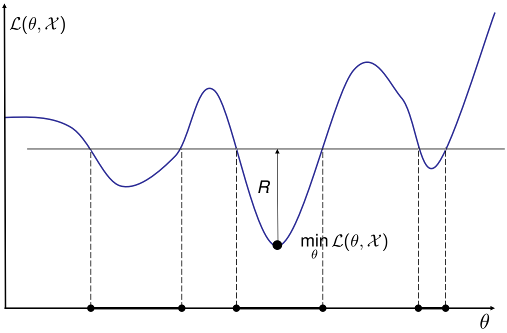
Overlap Gap Property



Overlap Gap Property



Overlap Gap Property



OGP for Independent Sets in Random Graphs

OGP for Independent Sets in Random Graphs

For either $\mathbb{G}(n, p)$ or $\mathbb{G}(n, \frac{d}{n})$, let $\mathcal{I}(\beta), \beta \in (0, 1)$ be the set of ind sets with multiplicative approximation ratio β. Then the set of overlaps

$$|I \cap J|, \qquad I, J \in \mathcal{I}(\beta)$$

is

OGP for Independent Sets in Random Graphs

For either $\mathbb{G}(n, p)$ or $\mathbb{G}(n, \frac{d}{n})$, let $\mathcal{I}(\beta), \beta \in (0, 1)$ be the set of ind sets with multiplicative approximation ratio β. Then the set of overlaps

$$|I \cap J|, \qquad I, J \in \mathcal{I}(\beta)$$

is

- disconnected, for *all* pairs when $\beta > \frac{1}{2} + \frac{1}{2\sqrt{2}}$, and for *"most"* pairs when $\beta > \frac{1}{2}$, Coja–Oghlan & Efthymiou [2013], G & Sudan [2014]

OGP for Independent Sets in Random Graphs

For either $\mathbb{G}(n, p)$ or $\mathbb{G}(n, \frac{d}{n})$, let $\mathcal{I}(\beta), \beta \in (0, 1)$ be the set of ind sets with multiplicative approximation ratio β. Then the set of overlaps

$$|I \cap J|, \qquad I, J \in \mathcal{I}(\beta)$$

is

- disconnected, for *all* pairs when $\beta > \frac{1}{2} + \frac{1}{2\sqrt{2}}$, and for *"most"* pairs when $\beta > \frac{1}{2}$, Coja–Oghlan & Efthymiou [2013], G & Sudan [2014]
- connected for $\beta < \frac{1}{2}$, Coja–Oghlan & Efthymiou [2013].

OGP for Independent Sets in Random Graphs

For either $\mathbb{G}(n,p)$ or $\mathbb{G}(n,\frac{d}{n})$, let $\mathcal{I}(\beta), \beta \in (0,1)$ be the set of ind sets with multiplicative approximation ratio β. Then the set of overlaps

$$|I \cap J|, \qquad I, J \in \mathcal{I}(\beta)$$

is

- disconnected, for *all* pairs when $\beta > \frac{1}{2} + \frac{1}{2\sqrt{2}}$, and for *"most"* pairs when $\beta > \frac{1}{2}$, Coja–Oghlan & Efthymiou [2013], G & Sudan [2014]
- connected for $\beta < \frac{1}{2}$, Coja–Oghlan & Efthymiou [2013].
- Extension to multi-overlaps $|I_1 \cap \cdots \cap I_m|$ by Rahman & Virag [2015].

OGP for Random K-SAT Problem

Let $\mathcal{SAT}_n$ be the set of satisfying assignments of a random formula $\Phi(n, dn)$.

Let $\mathcal{SAT}_n$ be the set of satisfying assignments of a random formula $\Phi(n, dn)$.

Then the set of multi-overlaps for a fixed m

$$(d(\sigma_\ell, \sigma_r), \sigma_\ell \in \mathcal{SAT}_n, 1 \leq \ell, r \leq m).$$

OGP for Random K-SAT Problem

Let $\mathcal{SAT}_n$ be the set of satisfying assignments of a random formula $\Phi(n, dn)$.

Then the set of multi-overlaps for a fixed m

$$(d(\sigma_\ell, \sigma_r), \sigma_\ell \in \mathcal{SAT}_n, 1 \leq \ell, r \leq m).$$

- is connected when $d < \frac{2^K \log 2}{K} = \frac{d^*}{K}$ for $m = 2$, general case open.

OGP for Random K-SAT Problem

Let $\mathcal{SAT}_n$ be the set of satisfying assignments of a random formula $\Phi(n, dn)$.

Then the set of multi-overlaps for a fixed m

$$(d(\sigma_\ell, \sigma_r), \sigma_\ell \in \mathcal{SAT}_n, 1 \le \ell, r \le m).$$

- is connected when $d < \frac{2^K \log 2}{K} = \frac{d^*}{K}$ for $m = 2$, general case open.
- disconnected when $d > \frac{2^K \log^2 K \log 2}{K} = \frac{\log^2 K}{K} d^*$, Achlioptas, Coja-Oghlan & Ricci-Tersenghi [2011], G & Sudan [2014] (for NAE-K-SAT).

OGP in Sparse Regression

OGP in Sparse Regression

Assume binary case $\beta_i^* = 0, 1$.

OGP in Sparse Regression

Assume binary case $\beta_i^* = 0, 1$.
Consider the optimization problem parametrized by $\zeta \in (0, 1)$

$$\Gamma^*(\zeta) \triangleq \min_{\beta} \|Y - X\beta\|_2$$

$$\text{Subject to: } \|\beta\|_0 = k,$$
$$\|\beta^* - \beta\|_0 = 2k\zeta.$$

Then $\Gamma(\zeta)$ is

OGP in Sparse Regression

Assume binary case $\beta_i^* = 0, 1$.
Consider the optimization problem parametrized by $\zeta \in (0, 1)$

$$\Gamma^*(\zeta) \triangleq \min_{\beta} \|Y - X\beta\|_2$$
$$\text{Subject to: } \|\beta\|_0 = k,$$
$$\|\beta^* - \beta\|_0 = 2k\zeta.$$

Then $\Gamma(\zeta)$ is

- increasing when $n > Cn^*_{\text{Convex}}$,

OGP in Sparse Regression

Assume binary case $\beta_i^* = 0, 1$.
Consider the optimization problem parametrized by $\zeta \in (0, 1)$

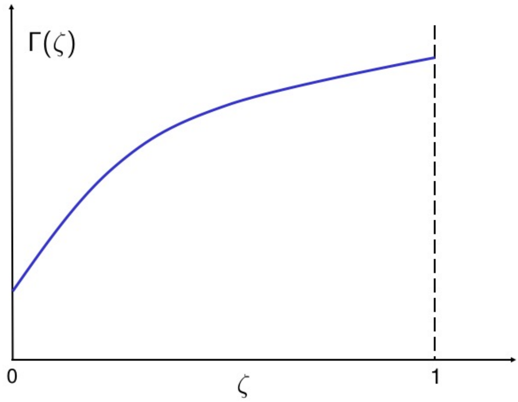
$$\Gamma^*(\zeta) \triangleq \min_{\beta} \|Y - X\beta\|_2$$
$$\text{Subject to: } \|\beta\|_0 = k,$$
$$\|\beta^* - \beta\|_0 = 2k\zeta.$$

Then $\Gamma(\zeta)$ is

- increasing when $n > Cn^*_{\text{Convex}}$,
- non-monotone when $n < cn^*_{\text{Convex}}$, G & Zadik [2017].

OGP in Sparse Regression

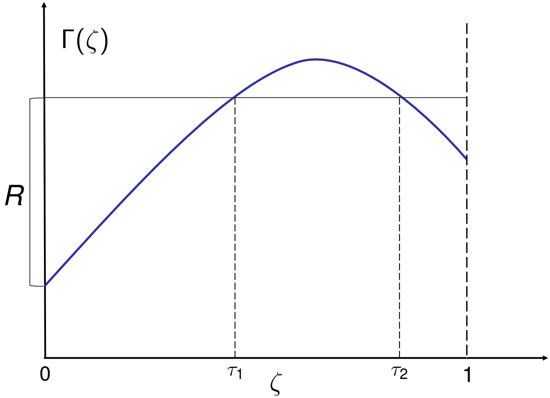
When $n > Cn^*_{\text{Convex}}$



OGP in Sparse Regression

OGP in Sparse Regression

when $n < cn^*_{\text{Convex}}$ – OGP.



Part III. Local algorithms

Three Types of Local Algorithms

Three Types of Local Algorithms

- Greedy. Example for ind sets: select a node into I. Delete neighbors. Repeat.

Three Types of Local Algorithms

- Greedy. Example for ind sets: select a node into I. Delete neighbors. Repeat.

Three Types of Local Algorithms

- Greedy. Example for ind sets: select a node into I. Delete neighbors. Repeat.
-
    - Local-in-Parallel – Factors of I.I.D (FIID). Make randomized decision based on the radius r neighborhood $B(u, r)$ for each node u.

Three Types of Local Algorithms

- Greedy. Example for ind sets: select a node into I. Delete neighbors. Repeat.
-
    - Local-in-Parallel – Factors of I.I.D (FIID). Make randomized decision based on the radius r neighborhood $B(u, r)$ for each node u.
    - Local-Sequential. Make decision on u_i based on "implied" decisions for $u_1, \ldots, u_{i-1}$.

Three Types of Local Algorithms

- Greedy. Example for ind sets: select a node into I. Delete neighbors. Repeat.
-
    - Local-in-Parallel – Factors of I.I.D (FIID). Make randomized decision based on the radius r neighborhood $B(u, r)$ for each node u.
    - Local-Sequential. Make decision on u_i based on "implied" decisions for $u_1, \ldots, u_{i-1}$.
- Markov Chain Monte Carlo (MCMC). Example for ind sets. Fix β – inverse temperature.

Three Types of Local Algorithms

- Greedy. Example for ind sets: select a node into I. Delete neighbors. Repeat.
-
    - Local-in-Parallel – Factors of I.I.D (FIID). Make randomized decision based on the radius r neighborhood $B(u, r)$ for each node u.
    - Local-Sequential. Make decision on u_i based on "implied" decisions for $u_1, \ldots, u_{i-1}$.
- Markov Chain Monte Carlo (MCMC). Example for ind sets. Fix β – inverse temperature.
  Pick u u.a.r.

Three Types of Local Algorithms

- Greedy. Example for ind sets: select a node into I. Delete neighbors. Repeat.
-
  - Local-in-Parallel – Factors of I.I.D (FIID). Make randomized decision based on the radius r neighborhood $B(u, r)$ for each node u.
  - Local-Sequential. Make decision on u_i based on "implied" decisions for $u_1, \dots, u_{i-1}$.
- Markov Chain Monte Carlo (MCMC). Example for ind sets. Fix β – inverse temperature.
  Pick u u.a.r.
  - If $u \in I$ delete with probability $\frac{1}{1+e^{\beta}}$.

Three Types of Local Algorithms

- Greedy. Example for ind sets: select a node into I. Delete neighbors. Repeat.
-
  - Local-in-Parallel – Factors of I.I.D (FIID). Make randomized decision based on the radius r neighborhood $B(u, r)$ for each node u.
  - Local-Sequential. Make decision on u_i based on "implied" decisions for $u_1, \ldots, u_{i-1}$.
- Markov Chain Monte Carlo (MCMC). Example for ind sets. Fix β – inverse temperature.
  Pick u u.a.r.
  - If $u \in I$ delete with probability $\frac{1}{1+e^{\beta}}$.
  - If $u \notin I$ and can be added without conflict, add it with prob $\frac{e^{\beta}}{1+e^{\beta}}$.

Three Types of Local Algorithms

- Greedy. Example for ind sets: select a node into I. Delete neighbors. Repeat.
-
  - Local-in-Parallel – Factors of I.I.D (FIID). Make randomized decision based on the radius r neighborhood $B(u, r)$ for each node u.
  - Local-Sequential. Make decision on u_i based on "implied" decisions for $u_1, \ldots, u_{i-1}$.
- Markov Chain Monte Carlo (MCMC). Example for ind sets. Fix β – inverse temperature.
  Pick u u.a.r.
  - If $u \in I$ delete with probability $\frac{1}{1+e^\beta}$.
  - If $u \notin I$ and can be added without conflict, add it with prob $\frac{e^\beta}{1+e^\beta}$.
- Gradient Descent (GD): keep improving while you can (MCMC with $\beta = \infty$)

Properties

Properties

- Greedy and Local-in-Parallel are special cases of Local-Sequential (Obvious).

Properties

- Greedy and Local-in-Parallel are special cases of Local-Sequential (Obvious).
- Greedy for ind set can be well approximated by Local-in-Parallel, G & Goldberg [2010]

Properties

- Greedy and Local-in-Parallel are special cases of Local-Sequential (Obvious).
- Greedy for ind set can be well approximated by Local-in-Parallel, G & Goldberg [2010]
  - Proof sketch: assign weights $R(u_1), \ldots, R(u_n)$ i.i.d. u.a.r. from $[0, 1]$.

Properties

- Greedy and Local-in-Parallel are special cases of Local-Sequential (Obvious).
- Greedy for ind set can be well approximated by Local-in-Parallel, G & Goldberg [2010]
    - Proof sketch: assign weights $R(u_1), \ldots, R(u_n)$ i.i.d. u.a.r. from $[0, 1]$.
    - For each node u for the decision *not to be* r-local there should exist a path $u, u_1, \ldots, u_r$ such that $R(u_1) < R(u_2) < \cdots < R(u_r)$.

Properties

- Greedy and Local-in-Parallel are special cases of Local-Sequential (Obvious).
- Greedy for ind set can be well approximated by Local-in-Parallel, G & Goldberg [2010]
  - Proof sketch: assign weights $R(u_1), \ldots, R(u_n)$ i.i.d. u.a.r. from $[0, 1]$.
  - For each node u for the decision *not to be* r-local there should exist a path $u, u_1, \ldots, u_r$ such that $R(u_1) < R(u_2) < \cdots < R(u_r)$.
  - The probability of this is $O\left(\frac{1}{r!}\right)$ – decays faster than the exponential growth rate d^r of the neighborhood of r.

Properties

- Greedy and Local-in-Parallel are special cases of Local-Sequential (Obvious).
- Greedy for ind set can be well approximated by Local-in-Parallel, G & Goldberg [2010]
    - Proof sketch: assign weights $R(u_1), \ldots, R(u_n)$ i.i.d. u.a.r. from $[0, 1]$.
    - For each node u for the decision *not to be* r-local there should exist a path $u, u_1, \ldots, u_r$ such that $R(u_1) < R(u_2) < \cdots < R(u_r)$.
    - The probability of this is $O\left(\frac{1}{r!}\right)$ – decays faster than the exponential growth rate d^r of the neighborhood of r.
    - As a result, for most nodes the decision is r-localized.

Properties

Properties

- MCMC has a simple to describe (Gibbs) steady state: for every ind set

$$\mathbb{P}(I) = \frac{e^{\beta|I|}}{\sum_J e^{\beta|I|}}.$$

Properties

- MCMC has a simple to describe (Gibbs) steady state: for every ind set

$$\mathbb{P}(I) = \frac{e^{\beta|I|}}{\sum_J e^{\beta|I|}}.$$

- MCMC *always* produces a near-optimal solution for large enough β, (but possibly after exponential time), for the following reason:

$$|I^*| \le \frac{1}{\beta}\log\left(\sum_J e^{\beta|I|}\right) \le |I^*| + n\frac{\log 2}{\beta}.$$

Properties

- MCMC has a simple to describe (Gibbs) steady state: for every ind set

$$\mathbb{P}(I) = \frac{e^{\beta|I|}}{\sum_J e^{\beta|I|}}.$$

- MCMC *always* produces a near-optimal solution for large enough β, (but possibly after exponential time), for the following reason:

$$|I^*| \le \frac{1}{\beta}\log\left(\sum_J e^{\beta|I|}\right) \le |I^*| + n\frac{\log 2}{\beta}.$$

- As a result, most of the Gibbs ”mass” is concentrated on ind sets I with $|I| \approx |I^*|$.

Summary

Punchline: For all of the discussed models, below the OGP one of the versions of the local algorithms finds a solution. Above the threshold no algorithms are known and possibly no poly-time algorithms exist.

Examples: Ind set in sparse random graphs

In either $\mathbb{G}(n, \frac{d}{n})$ or $\mathbb{G}_d(n)$, local parallel (FIID) algorithms can construct ind sets with size

$$\frac{(1 + o_d(1)) \log d}{d} n$$

Lauer & Wormald [2007], G & Goldberg [2010].

Examples: Ind set in sparse random graphs

In either $\mathbb{G}(n, \frac{d}{n})$ or $\mathbb{G}_d(n)$, local parallel (FIID) algorithms can construct ind sets with size

$$\frac{(1 + o_d(1)) \log d}{d} n$$

Lauer & Wormald [2007], G & Goldberg [2010].
and no Local Parallel algorithms exist which construct ind set with larger size w.h.p., G & Sudan, Rahman & Virag [2015]. The OGP is used as an obstruction to such algorithms.

Examples: Ind set in sparse random graphs

In either $\mathbb{G}(n, \frac{d}{n})$ or $\mathbb{G}_d(n)$, local parallel (FIID) algorithms can construct ind sets with size

$$\frac{(1+o_d(1))\log d}{d} n$$

Lauer & Wormald [2007], G & Goldberg [2010].
and no Local Parallel algorithms exist which construct ind set with larger size w.h.p., G & Sudan, Rahman & Virag [2015]. The OGP is used as an obstruction to such algorithms.

Also MCMC fail finding ind sets of the same size Coja–Oghlan & Efthymiou [2013] (MCMC mixes in exponential time).

Examples: Sparse Regression

For sparse regression problem $Y = X\beta^* + W$, Gradient Descent reconstructs β^* when

$$n > Cn_{\text{Convex}} = \Omega\left(k \log p\right),$$

Examples: Sparse Regression

For sparse regression problem $Y = X\beta^* + W$, Gradient Descent reconstructs β^* when

$$n > Cn_{\text{Convex}} = \Omega\left(k \log p\right),$$

and Gradient Descent provably fails for certain starting points when

$$n < cn_{\text{Convex}} = \Omega\left(k \log p\right).$$

Examples: Sparse Regression

For sparse regression problem $Y = X\beta^* + W$, Gradient Descent reconstructs β^* when

$$n > Cn_{\text{Convex}} = \Omega\left(k \log p\right),$$

and Gradient Descent provably fails for certain starting points when

$$n < cn_{\text{Convex}} = \Omega\left(k \log p\right).$$

OGP serves as an obstruction to the GD algorithm, G & Zadik [2017]

Conclusion

In conclusion, we have a fairly complete theory: optimization in random graphs is solvable by local algorithms below the OGP threshold. Above the threshold local algorithms provably fails, but no other poly-time algorithms are known, and possibly none exist.

Future Work

Future Work

- Planted clique problem $\mathbb{G}(n, \frac{1}{2}, m)$ exhibits non-monotonicity implying OGP ***iff*** the size of the planted clique $m = o(\sqrt{n})$, G & Zadik [201?]

Future Work

- Planted clique problem $\mathbb{G}(n, \frac{1}{2}, m)$ exhibits non-monotonicity implying OGP ***iff*** the size of the planted clique $m = o(\sqrt{n})$, G & Zadik [201?]
- Sparse Planted XOR-K-SAT formula $\Phi(n, dn)$ exhibits non-monotonicity implying OGP.

Thank you