Subexponential Algorithms for Unique Games and Related Problems

Sanjeev Arora Boaz Barak David Steurer

May 5, 2011

1 Small-Set Expanstion

Theorem 1 *(Subexponential algorithm for small-set expansion). For every* $\beta \in (0,1)$*,* $\varepsilon > 0$*, and* $\delta > 0$*, there is an* $\exp(n^{O(\varepsilon^{1-\beta})})$*-time algorithm that on input a regular graph* G *with* n *vertices that has a set* S *of at most* δn *vertices satisfying* $\Phi(S) \leq \varepsilon$*, finds a set* S' *of at most* δn *vertices satisfying* $\Phi(S) \leq O(\varepsilon^{\beta/3})$*.*

Proof. Follows from Theorem 2 and 3. ■

Theorem 2 *(Eigenspace enumeration). There is an* $\exp(rank_{1-\eta}(G))$*-time algorithm that given* $\varepsilon > 0$ *and a graph* G *containing a set* S *with* $\Phi(S) \leq \varepsilon$*, outputs a sequence of sets, one of which has symmetric difference at most* $8(\varepsilon/\eta)|S|$ *with the set* S*.*

Theorem 3 *(Rank bound for small-set expanders). Let* G *be a regular graph on* n *vertices such that* $rank_{1-\eta}(G) > n^{100\eta/\gamma}$*. Then there exists a vertex set* S *of size at most* $n^{1-\eta/\gamma}$ *that satisfies* $\Phi(S) \leq \sqrt{\gamma}$*. Moreover,* S *is a level set of a column of* $(\frac{1}{2}I + \frac{1}{2}G)^j$ *for some* $j \leq O(\log n)$*.*

Proof. Follows from Theorem 4. ■

Theorem 4 *(Schatten norm bound). Let* G *be a lazy regular graph on* n *vertices. Suppose every vertex set* S *with* $\mu(S) \leq \delta$ *satisfies* $\Phi(S) \geq \varepsilon$*. Then, for all even* $k > 2$*, the* k*-Schatten norm of* G *satisfies*

$$S_k(G)^k \leq \max\left\{ n\left(1 - \frac{\varepsilon^2}{32}\right)^k, \frac{4}{\delta} \right\}.$$

Moreover, for any graph that does not satisfy the above bound, we can compute in polynomial time a vertex subset S *with* $\mu(S) \leq \delta$ *and* $\Phi(S) \leq \varepsilon$*, where* S *is a level set of a column of* G^j *for some* $j \leq k$*.*

2 Unique Games

Definition 5 *A unique game of* n *variables and alphabet* k *is an* n *vertex graph* G *whose edges are labeled with permutations on the set* $[k]$*, where the edge* (i,j) *labeled with* π *iff the edge* (j,i) *is labeled with* π^{-1}*. An assignment to the game* *is a string* $y = (y_1, \dots, y_n) \in [k]^n$*, and the value of* y *is the fraction of edges* (i, j) *for which* $y_j = \pi(y_i)$*, where* π *is the label of* (i, j)*. The value of the game* G *is the maximum value of* y *over all* $y \in [k]^n$*.*

Theorem 6 *(Subexponential algorithm for unique games). There is an* $exp(kn^{O(\varepsilon)})$ *poly(n)-time algorithm that on input a unique game* G *on* n *vertices and alphabet size* k *that has an assignment satisfying* $1 - \varepsilon^6$ *of its constraints outputs an assignment satisfying* $1 - O(\varepsilon log(1/\varepsilon))$ *of the constraints.*

Theorem 7 *(Low threshold rank decomposition theorem). There is a polynomial time algorithm that on input a graph* G *and* $\varepsilon > 0$*, outputs a partition* $\chi = (A_1, \dots, A_q)$ *of* $V(G)$ *such that* $\Phi(\chi) \leq O(\varepsilon \log(1/\varepsilon))$ *and for every* $i \in [q]$*,*

$$rank_{1-\varepsilon^5}(G[A_i]) \leq n^{100\varepsilon}.$$

Lemma 8 *There is a polynomial-time algorithm that given an* n *vertex graph* G *and* $\varepsilon > 0$*, outputs a partition* $\chi = (A_1, \dots, A_r, B)$ *of* $V(G)$*, such that* $\Phi(\chi) \leq O(\varepsilon^2), |A_i| \leq n^{1-\varepsilon}$ *for all* $i \in [r]$*, and*

$$rank_{1-\varepsilon^5}(G[B]) \leq n^{100\varepsilon}.$$

Lemma 9 *There is an algorithm that given an* n *vertex graph* G*, and* $\varepsilon > 0$*, outputs a partition* $\chi = (A_1, \dots, A_q, B_1, \dots, B_r)$ *of* $[n]$*, such that* $\Phi(\chi) \leq O(\varepsilon \log(1/\varepsilon))$*,* $|A_i| \leq n^{100\varepsilon}$ *for all* $i \in [q]$*, and for all* $j \in [r]$

$$rank_{1-\varepsilon^5}(G[B_j]) \leq n^{100\varepsilon}.$$

Algorithm 1: A subexponential algorithm form Unique Games

Input: Unique Game G on n variables of alphabet k that has value at least $1 - \varepsilon^6$.

Make G lazy.

Run the partition algorithm of Theorem 7 to obtain a partition $\chi = (A_1, \dots, A_q)$, of the graph G with $O(\varepsilon \log(1/\varepsilon))$ such that for every i, $rank_{1-\varepsilon^5}(\hat{G}[A_i]) \leq kn^{100\varepsilon}$.

for *every* $t = 1, \dots q$ **do**

- Run $exp(rank_{1-\varepsilon^5}(G[A_t])$-time enumeration algorithm of Theorem 2 to obtain a sequence of sets S_t.
- For every set $S \in S_t$, we compute an assignment f_S to the vertices in A_t as follows: For every $i \in A_t$, if $C_i \cap S = \emptyset$, then f_S assigns an arbitrary label to the vertex i, if $|C_i \cap S| > 0$, then f_S assigns one of the labels in $C \cap S$ to the vertex i. We pick f_t with maximum value.
