CS 354: Unfulfilled Algorithmic Fantasies May 20

Lecture 1: Quasi-Polynomial Approximation for Bilinear Optimization

Lecturer: Aviad Rubinstein *Scribe: Hubert Teo*

1 Introduction

In this lecture we discuss a bilinear optimization framework that includes several problems and can be additively approximated in quasi-polynomial time. We introduce the Exponential Time Hypothesis (ETH), which implies that those problems are not NP-hard. We use the same hypothesis to show that some of them cannot be approximated in polynomial time. We do this using an important technique called *birthday repetition*.

2 Bilinear Optimization

$$\begin{aligned} \underset{x,y}{\text{maximize}} \quad & x^\top A y \\ \text{subject to} \quad & \text{some constraints} \end{aligned}$$

Bilinear optimization problems are those which can be defined in the above form. Some examples of bilinear optimization problems include:

1. **Sparse PCA.**

   Given the covariance matrix $A = \Sigma$ of some data, find x to maximize $x^T A x$, subject to $\|x\|_2 = 1$, such that $\|x\|_0 \leq m$ so that x is sparse.

2. **Densest k-subgraph.**

   Given the adjacency matrix A of some graph, find a subset of k vertices $x \in \left\{0, \frac{1}{k}\right\}^n, \|x\|_1 = 1$ that maximizes the number of edges $x^T A x$.

3. **2-player Nash equilibrium.**

   Find a locally maximal strategy x, y of the game by fixing y and maximizing $x^\top A y$, as well as fixing x and maximizing $x^\top B y$. We need $\|x\|_1 = \|y\|_1 = 1$, and $0 \preceq x, y \preceq 1$.

4. **Max-Cut**

   Find a maximum cut x of a graph and its complement $y = 1 - x$, with $x_i \in \{0, 1\}$.

5. **Community detection**

   Find a subset x of vertices with many edges within the subset, but few outside the subset. This is equivalent to maximizing $x^\top A x$ and minimizing $x^\top A(1 - x)$.

Theorem 2.1 (Additive approximation in quasi-polynomial time, informal [LMM03, AGSS12, Bar15])**.** *There is a quasi-polynomial* $N^{\mathcal{O}\left(\frac{\log N}{\varepsilon^2}\right)}$ *time algorithm that additively*[1] *ε-approximates bilinear optimization.*

It turns out that these problems are almost-polynomially approximable, by Theorem 2.1. The proof follows from an approximate Carathéodory theorem from [Bar15].

Theorem 2.2 (Approximate Carathéodory)**.** *Take any $x^\top A \in \operatorname{conv}(A_1, \ldots, A_n)$ with bounded columns $\max_{A_i}\|A_i\|_\infty \leq 1$. A vector x is a multiset of size k if each $x_i = \frac{m}{k}$ for $m \in \mathbb{Z}^*$, and $\|x\|_1 = 1$. For every $\varepsilon > 0$, $\exists \hat{x}$ which is a multiset of size $\mathcal{O}\left(\frac{\log N}{\varepsilon^2}\right)$, such that $\left\|x^\top A - \hat{x}^\top A\right\|_\infty < \varepsilon$.*

Proof of Theorem 2.1 (sketch). Enumerate all size-$k = \mathcal{O}\left(\frac{\log N}{\varepsilon^2}\right)$ multisets, of which there are $N^{\mathcal{O}(\log N)}$. For each $\hat{x}^\top A$, maximize $\hat{x}^\top Ay$ for y. (If y is normalized appropriately, this produces a ε-approximation to the solution). □

3 Approximation-hardness and quasi-polynomial time

So, we know that those 5 bilinear optimization problems can be additively approximated in quasi-polynomial time. **But can we approximate them in polynomial time?** It turns out that MAX-CUT [AKK95] and Sparse-PCA [ALSV13, APKD15] can actually be approximated in polynomial time. The key idea is similar to the proof of Theorem 2.1: we can exhaustively enumerate a smaller set of assignments, transform them in some way to a feasible solution, and the optimum solution should appear as one of those solutions.

Algorithm 3.1 (Approximate MAX-CUT in polynomial time)**.** *Sample a subset S of $k = \mathcal{O}\left(\frac{\log N}{\varepsilon^2}\right)$ vertices at random, and enumerate all $2^k = N^{\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)}$ assignments to those vertices. For each of the other vertices, put it on the opposite side of the majority of its neighbors. Return the best cut.*

Proof of Alg. 3.1 (sketch). [AKK95] With high probability, every unsampled vertex has at least some of its neighbors sampled. In a maximal cut, every vertex has the majority of its neighbors on the opposite side, otherwise it wouldn't be maximal. In any cut, either the cut or its complement has at least half of the vertices. Hence, we expect most of the sampled neighbors to be on the opposite side of the unsampled vertex. The resulting error is additive. □

Algorithm 3.2 (Approximate Sparse-PCA in polynomial time)**.** *[ALSV13], [APKD15] $A = \Sigma \succeq 0$, so $\exists B$, $A = BB^\top$. It is possible to construct a lower-rank representation $\hat{B} \in \mathbb{R}^{n\times d}$, with $d = \mathcal{O}\left(\frac{\log N}{\varepsilon^2}\right)$ such that $\left\|BB^\top - \hat{B}\hat{B}^\top\right\|_\infty \leq \varepsilon$ with high probability. Now, instead of enumerating x, we can enumerate $x^\top \hat{B}$ with an ε-net of $\mathbb{R}^d$.*

But for the other problems, Densest-k-Subgraph, 2-NE and COMMUNITY we don't know polynomial time approximation algorithms. Maybe none exist... and if so, can we explain why?

[1] This additive approximation is w.r.t. an appropriate normalization.

4 Exponential Time Hypothesis

Definition 4.1 (Exponential Time Hypothesis)**.** 3-SAT takes $2^{\Omega(n)}$ time.

Assuming the ETH, Densest-k-Subgraph cannot be NP-Hard to approximate. But our main result for this lecture and the next is that assuming the same ETH, Densest-k-Subgraph also doesn't have polynomial time approximation algorithms.

Theorem 4.2 (Almost-polynomial ETH-hardness of approximating Densest-k-Subgraph [Man17])**.** *There is no polynomial time algorithm distinguishing graphs with the following properties:*

1. ***Completeness.*** *There is a k-clique in the graph.*
2. ***Soundness.*** *Every k-subgraph has $o(k^2)$ edges.*

Remark (Planted Clique Conjecture (PC) *v.s.* ETH)*.* What is the relationship between PC and ETH? Recall that the Planted Clique conjecture is that there is no polynomial time algorithm that distinguishes random graphs with graphs that have a planted k-clique. Hence,

1. PC is a special (average) case of the $\frac{1}{2}$-*v.s.*-1 Densest-k-Subgraph problem.
2. [AAM+11] showed that PC also implies hardness for the easier $o(1)$-*v.s.*-1 Densest-k-Subgraph, similarly to Theorem 4.2.
3. PC and ETH are incomparable, i.e. neither assumption implies the other one:
   (a) On one hand, PC is a statement about average-case hardness. In this sense PC is a stronger assumption, so ETH seems unlikely to imply PC.
   (b) On the other hand, ETH asks for exponential hardness. In this sense ETH is stronger, so it is also unlikely that PC implies ETH.

Remark (k-CLIQUE)*.* How hard is it to approximate k-CLIQUE? That depends on what we mean by "approximation". We can look at relaxing the "k" in k-CLIQUE, namely look for a smaller clique in a graph that contains a k-clique; this is NP-hard to approximate within $N^{1-\varepsilon}$ [Hås99, Zuc07]. Alternatively, we can consider relaxing the "clique" in k-CLIQUE, namely look for a dense k-subgraph in a graph that contains a k-clique; as we show in this and the next lecture, the right complexity here is quasi-polynomial.

5 Hardness of exact k-CLIQUE

As a warmup to the next lecture (where we will sketch the reduction provided by [Man17]) let's prove the hardness of approximating k-CLIQUE assuming ETH. Firstly, to solve k-CLIQUE, we can brute-force all k-subsets of vertices and check if they are cliques. This takes $N^{\emptyset(k)}$. In addition, assuming the ETH, this is optimal:

Theorem 5.1. *Assuming* ETH*, the k-CLIQUE problem requires $N^{\Omega(k)}$ time.*

We reduce from 3-SAT to 3-COLOR to k-CLIQUE. There is a classical reduction from 3-SAT to 3-COLOR which produces 2 nodes per variable. Hence, it suffices to reduce 3-COLOR to k-CLIQUE.

5.1 Reducing 3-COLOR to k-CLIQUE

Given a graph $G = (V, E)$, partition the vertices $V = \sqcup_{i=1}^{k} V_i$ into k sets of size $|V_i| = l_i = \frac{n}{k}$. Then, for each assignment $\sigma_i : V_i \rightarrow [3]$, create a new vertex $u_{i,\sigma_i} \in V_{clique}$. Hence, for each subset V_i and each of the $3^{|V_i|} = 3^{n/k}$ assignments σ_i we have one vertex u_{i,σ_i}, so there are $N = |V_{clique}| = k3^{n/k} = 2^{\Theta(n/k)}$ vertices.

Create an edge $(u_{i,\sigma_i}, u_{j,\tau_j}) \in E_{clique}$ only between distinct subsets $i \neq j$, and only if the subsets σ_i, τ_j satisfy all the coloring constraints on V_i, V_j. (That is, all edges within V_i and V_j have endpoints with different colors under the assignment provided by σ_i, τ_j.)

Intuitively, there are no edges within one subset V_i's nodes. Hence, a k-clique in G_{clique} selects an assignment for each of the k subsets V_i which satisfies all the coloring constraints in G. More formally, For completeness, suppose there is some coloring $\sigma : V \rightarrow [3]$. Then let $\sigma_i = \sigma(v_i)$. $u_{1,\sigma_1}, \dots, u_{n,\sigma_n}$ form a clique because they satisfy all constraints. For soundness, each k-clique must have one vertex for each i, which produces a coloring that satisfies all the constraints.

This implies a lower-bound for k-CLIQUE under ETH. Since $2^{\Theta(n)} = \left(2^{\Theta(n/k)}\right)^k = N^{\Theta(k)}$, if it takes at least $2^{\Theta(n)}$ time to solve 3-COLOR, it must take at least $N^{\Theta(k)}$ time to solve k-CLIQUE.

6 Birthday Repetition

Suppose we wanted to use the above reduction to show hardness of Densest-k-Subgraph, aka approximate k-CLIQUE. Consider a random or even arbitrary partial coloring σ_i, σ_j. Even if σ_i, σ_j don't correspond to an (approximately) satisfying coloring, in the clique graph we would still have an edge $(u_{i,\sigma_i}, u_{j,\sigma_j})$ if V_i, V_j don't share any edges. For some fixed i, j, what is the probability that the edge $(u_{i,\sigma_i}, u_{j,\sigma_j})$ is in the graph E_{clique}? Note that

$$\Pr\left[(u_{i,\sigma_i}, u_{j,\sigma_j}) \in E_{clique}\right] = \Pr\left[\sigma_i, \sigma_j \text{ satisfies constraints on } V_i, V_j\right] \approx \left(\frac{2}{3}\right)^{|E \cap V_i \times V_j|}$$

How large should $|V_i|$ be for there to be at least one edge between each V_i, V_j? The **birthday paradox** states that we need $|V_i| \approx \sqrt{n}$. This means that to prove hardness of Densest-k-Subgraph, we should set $|V_i| \approx \sqrt{n}$, which implies $N \approx 2^{\sqrt{n}}$. Hence, such a reduction could show that, assuming ETH, Densest-k-Subgraph takes $2^{\Omega(n)} \approx N^{\Omega(\log N)}$ time.

Constructing meta-vertices that correspond to a large set V_i of 3-COLOR-nodes is closely related to an important technique in hardness-of-approximation which is called *parallel repetition*. In typical NP-hardness proofs $|V_i|$ is constant so the reduction can run in polynomial time. As described above, the use of $|V_i| \approx \sqrt{n}$ is inspired by the birthday paradox, and hence the name *birthday repetition*.

References

[AAM+11] Noga Alon, Sanjeev Arora, Rajsekar Manokaran, Dana Moshkovitz, and Omri Weinstein. Inapproximabilty of densest k-subgraph from average case hardness, 2011.

[AGSS12] Sanjeev Arora, Rong Ge, Sushant Sachdeva, and Grant Schoenebeck. Finding overlapping communities in social networks: toward a rigorous approach. In *Proceedings of the*

13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012, pages 37–54, 2012.

[AKK95] Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 284–293, New York, NY, USA, 1995. ACM.

[ALSV13] Noga Alon, Troy Lee, Adi Shraibman, and Santosh Vempala. The approximate rank of a matrix and its algorithmic applications: Approximate rank. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 675–684, New York, NY, USA, 2013. ACM.

[APKD15] Megasthenis Asteris, Dimitris Papailiopoulos, Anastasios Kyrillidis, and Alexandros G Dimakis. Sparse pca via bipartite matchings. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 766–774. Curran Associates, Inc., 2015.

[Bar15] Siddharth Barman. Approximating Nash equilibria and dense bipartite subgraphs via an approximate version of Caratheodory's theorem. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 361–369, New York, NY, USA, 2015. ACM.

[Hås99] Johan Håstad. Clique is hard to approximate within $n^{\text{1-epsilon}}$. *Acta Mathematica*, 182(1):105–142, 1999.

[LMM03] Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, EC '03, pages 36–41, New York, NY, USA, 2003. ACM.

[Man17] Pasin Manurangsi. Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 954–961, New York, NY, USA, 2017. ACM.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007.