EDAN55, supplementary notes for inapproximability

2012-10-19, rev. d995d2c

This note extends the presentation in Kleinberg and Tardos, chapter 11 with some inapproximability results, making plausible that the tools from computational complexity can be used to argue about approximation quality.

11.9 Inapproximability

No FPTAS for Max 3-Sat

In [KT, 11.8] we saw that there is an algorithm for the Knapsack problem that given any $\epsilon > 0$ computed a feasible solution to an instance of size n in time $O(n^3\epsilon^{-1})$ that is at most a factor $(1+\epsilon)$ below the maximum possible.

Such an algorithm is called a *fully polynomial time approximation scheme.*

Such an algorithm cannot exist for Max 3-Sat, the problem considered in [KT 13.4].

Theorem 1. *There is no FPTAS for for Max 3-Sat unless P = NP.*

Proof. Assume there was such an FPTAS. Let ϕ be an instance to the decision problem 3-Sat with m clauses. (The total size of ϕ is $O(m)$.) Set $\epsilon = \frac{1}{2m}$ and run the FPTAS.

If ϕ is satisifiable, all $m =:$ OPT clauses can be satisfied, so the FPTAS returns a solution of size at least $(1-\epsilon)m = m - \frac{1}{2}$. Since the solution size is an integer, the solution size is equal to m, so the FPTAS has solved the decision problem.

The running time of the FPTAS is polynomial in the input size m and inverse polynomial in the approximation guarantee $\frac{1}{2m}$, so the algorithm runs in polynomial time. □

It is known (but far beyond the scope of these notes) that for any $\epsilon > 0$ there is no polynomial-time algorithm for 3-Sat that satisfies more than $\frac{7}{8} + \epsilon$ of the clauses, unless P = NP.[1]. This is a tight bound for Johnson's algorithm from [KT 13.4], so this particular problem is fully understood.

[1] Johan Håstad, *Some optimal inapproximability results*, Journal of the ACM (ACM) 48: 798–859, 2011.

TSP

We first present a simple approximation algorithm for the Traveling Salesman Problem in *metric* graphs, i.e., the distance function satisfies the triangle inequality $d(u,w) \leq d(u,v) + d(v,w)$ for all vertices u, v, w.

Theorem 2. *If the distances in an instance to TSP satisfy the triangle inequality then there is a polynomial-time 2-approximation algorithm.*

Proof. 1. Find a minimum spanning tree T of the given graph G.

2. Perform a depth-first search of T from vertex 1.

3. Visit the vertices of the given graph in the depth first search order, and finally return to 1.

To see that this works, let OPT denote the length of an optimal TSP tour in G. Note that the total weight $|T|$ is at most OPT, because removing any edge from a tour makes the tour a spanning tree, and T is the minimum spanning tree in G. Consider now the nonsimple tour T' given by a traversal of T. This tour has length $2|T|$, because it travels along every edge of T exactly twice. Note that the depth first search order is a subsequence of T', so it can be viewed as the result of a successive application of operations that replace sequences u, v, w by the sequence u, w. If the distances in G satisfy the triangle inequality, none of these operations can increase the length of the tour. In particular, the resulting tour has length at most 2OPT. □

A simple modification of this algorithm reduces the approximation factor from 2 to $\frac{3}{2}$; this is a classical result.[2] With even more restrictions on the distance measure, much faster algorithms are possible: If the require that the distance function is Euclidian then TSP can be approximated within a factor $(1+\epsilon)$ in polynomial time for any given $\epsilon > 0$.[3]

[2] N. Christofides, *Worst-case analysis of a new heuristic for the travelling salesman problem*, Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, 1976.

[3] Sanjeev Arora. *Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and other Geometric Problems.* Journal of the ACM, Vol.45, Issue 5, pp.753–782, 1998. J.S.B. Mitchell, *Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems*, SIAM Journal on Computing 28 (4): 1298–1309, 1999.

Is the assumption about the triangle inequality crucial? It turns out that the answer is yes.

Theorem 3. *There can be no polynomial-time 2-approximation algorithm for TSP unless P = NP.*

Proof. We reduce from Hamiltonian Cycle, which is known to be NP-hard. Given an instance $G = (V, E)$ to Hamiltonian Cycle, build an instance $K = (V, E')$ to TSP as follows. The edge set is the complete set $E' = \{\, \{u, v\} \colon u \in V, v \in V \,\}$, and the distances are given by

$$c(e) = \begin{cases} 1\,, & e \in E\,; \\ |V| + 2\,, & e \notin E\,. \end{cases}$$

If G is a yes-instance, the shortest TSP tour has length $|V|$, so the hypothetical 2-approximation algorithm would return a solution of value at most $2|V|$.

If G is a no-instance, the shortest TSP tour must include at least one edge from $E' - E$, so the optimal tour has length at least $(|V| + 2) + |V| - 1 = 2|V| + 1$.

In particular, the approximation algorithm distinguishes yes- and no-instances to the Hamiltonian Cycle problem in polynomial time. □

The same proof rules out much less impressive approximation factors as well, such as $\rho = 10$ or $\rho = 10^{10}$. To show that TSP admits no polynomial-time ρ-approximation algorithm, the cost function in the proof has to be set to

$$c(e) = \begin{cases} 1\,, & e \in E\,; \\ (\rho - 1)|V| + 2\,, & e \notin E\,. \end{cases}$$

In fact, the proof works even for nonconstant approximation factors, and even superpolynomial ones. To prove that TSP cannot be approximated within $\exp(\Omega(|V|^{1/3}))$, set

$$c(e) = \begin{cases} 1\,, & e \in E\,; \\ 2^{|V|}|V| + 1\,, & e \notin E\,. \end{cases}$$

This is as far as we can push this proof, because the space needed to store the values on the edges start to dominate the instance size.

Approximation-preserving reductions

The examples so far established approximation hardness by reducing directly from a decision problem. However, there is a rich theory about reducibility *among* approximation problems as well.

For a simple example, consider the well-known reduction from 3-Sat to Independent Set in [KT, (8.8)]. There, it was given as a reduction between two *decision* problems. But he same construction can be used to reason about interdependencies with respect to approximation quality instead. These are called approximation-preserving reductions. A completely formal treatment of these notions is beyond the scope of these notes.

Theorem 4. *If Independent Set can be approximated within a factor* $(1+\epsilon)$ *in polynomial time then so can Max 3-Sat.*

Proof. Given an instance ϕ to Max 3-Sat with m clauses, follow the reduction in [KT, (8.8)] to build an instance G of Independent Set. The graph G contains $3m$ vertices. We note from the reduction that if the maximum number of satisfiable clauses ϕ is OPT then the largest independent set in G has size OPT as well. Conversely, every indpendent set of size k in G corresponds to an assignment in ϕ that satisfies at least k clauses.

Thus, a $(1+\epsilon)$-approximation algorithm for Independent Set would achieve the same approximation guarantee for Max 3-Sat. $\square$

In particular, we can use this reduction and the result of Håstad mentioned earlier to establish that no polynomial-time algorithm can approximate the size of a maximum independent set in a graph better than $\frac{7}{8} + \epsilon$. However, unlike the case for Max 3-Sat, this is far from optimal. It is known (but far beyond the scope of these notes) that Independent Set cannot be approximated in polynomial time within a factor $n^{1-\epsilon}$ for every $\epsilon > 0$, unless $\mathrm{P} = \mathrm{NP}$.[4]

[4] David Zuckerman, *Linear degree extractors and the inapproximability of max clique and chromatic number*, Proc. 38th ACM Symp. Theory of Computing, pp. 681–690, 2006.