CSC 373: Algorithm Design and Analysis
Lecture 24

Allan Borodin

March 18, 2013

Announcements

- All tutorials today in BA 2155
- I have posted additional questions for problem set 3.
- Proposal for change in grading scheme (if unanimous consent):
  1. For each student, I will count the final exam as 45% and term test 2 as 10% if your final exam score is better than your term test 2 score.
  2. A student pointed out a technical error in my slides on the local search algorithm for Exact Max-k-Sat and I appreciate being told about any and all errors. Depending on the seriousness of the error, I will award extra credit for technical corrections. Pointing out (say) notational errors will not gain extra credit but will be appreciated.

Today's outline

- Continue discussion of randomized algorithms
- Random sampling
- Polynomial identities and the symbolic determinant problem
- The Max-Sat problem and randomized rounding (if time permits)

Random assignments and the probabilistic method

- The naive randomized algorithm for Exact Max-k-Sat is an example of random sampling and the probabilistic method.
- That is, we are asserting the existence of something (i.e. an assignment satisfying some fraction of clauses) by a probabilistic argument.
- This is a standard approach where the expectation or non-zero probability of a random variable shows that something exists.
- In general, this is a non-constructive argument as we do not constructively give a specific solution satisfying the existential claim.
- However, in the case of the Exact Max-k-Sat problem, the method of conditional expectations does give us a constructive method.

- As another example, consider the following

The edge weighted 4-colouring optimization problem

- Given an edge weighted graph $G = (V, E, w)$ with edge weights $w(e) > 0$ on each edge $e \in E$.
- **Goal:** is to find a 4-colouring σ (of the nodes) so as to maximize the weighted sum of edges $e = (u, v)$ such that $\sigma(u) \neq \sigma(v)$; that is,

$$\max_{\sigma: V \to \{1,2,3,4\}} \sigma(G)$$

where $\sigma(G) = \sum_{e:e=(u,v)\in E, \sigma(u)\neq\sigma(v)} w_e$.

- **Claim:** There is a randomized algorithm for computing a 4-colouring σ such that for all inputs G, the expected value $E[\sigma(G)] \geq \frac{3}{4} W(G)$ where $W(G) = \sum_{e\in E} w(e)$.
- As in the Exact Max-k-Sat problem, the same naive setting of node colours guarantees the desired expectation and hence the existence of some colouring acheiving the expectation.

Polynomial identities – more random sampling

- We want to exploit the fact that "low degree" non zero polynomials have "few" zeros.

- In probabilistic terms when evaluated on a random point, a low degree non zero polynomial will likely not evaluate to zero.

Schwartz-Zipple Lemma

Let f be a non zero m-variate polynomial (say over a ring R) of degree $d \geq 0$. Let each r_i be randomly chosen from a subset S of R. Then

$$\mathbb{P}[f(r_1, \ldots, r_m) = 0] \leq \frac{d}{|S|}.$$

- We will consider two applications relating to polynomial identities, namely testing a matrix multiplication algorithm, and determining if a symbolic determinant is identically zero.

First application: testing if $C = A \cdot B$

- We might have a fast but not proven matrix multiplication algorithm.
- We want to use it but would like to be confident that when using it for a given input (A, B), it is unlikely to have made a mistake.
  (Debugging vs testing vs proving correctness)
- Suppose these are $n \times n$ matrices with elements in a ring R (e.g. $\mathbb{Z}$).
- We want to be able to test that the result $C = A \cdot B$ and do so much faster than say using a standard well proven (say $O(n^3)$) algorithm.
- Let S be an arbitrary subset of R and choose a random vector $x \in S^n$. Now check if $C \cdot x = A \cdot (B \cdot x)$, which takes time $3n^2$ using the standard matrix vector product algorithm.

Claim

$$\text{If } C \neq A \cdot B, \text{ then } \mathbb{P}[C \cdot x = A \cdot (B \cdot x)] \leq \frac{1}{|S|}.$$

A "puzzle" relating to interpolation

- Given an input n, we want to check if

$$det\left(\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \dots & x_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{bmatrix}\right) - \prod_{1 \le i < j \le n} (x_j - x_i) \equiv 0?$$

- From a theorem by Vandermonde, the answer is always yes.

- As a consequence, it follows that given the values of a polynomial at n distinct points, there is always a unique degree $n-1$ polynomial that satifies those values.

- But assume that we don't know this theorem (and a proof), how do we test if this identity is true?

Symbolic Determinant

- Recall the definition of a matrix determinant

$$det(A) = \sum_{\text{permutations } \pi} (-1)^{sgn(\pi)} \prod_{i} a_{i,\pi(i)}$$

- The definition makes sense when the matrix elements are in any ring R. In particular, R can be ring of polynomials in variables x_i and say integer or rational coefficients.

- Let A be an $n \times n$ matrix and say each matrix entry a_{ij} is a linear (resp. degree d) polynomial, then $det(A)$ is a degree n (resp. degree dn) polynomial in the variables x_i.

- The symbolic determinant problem is to determine whether or not $det(A)$ is the zero polynomial.

Motivation for symbolic determinant

- Consider the $n \times n$ adjacency matrix for a bipartite graph G.
- Suppose we wish to determine if G has a perfect matching.
- As we have seen, this problem can be solved in polytime by a transformation to max flow. But the max flow algorithm seems to be inherently sequential.
- We can solve the perfect matching problem by a transformation to the symbolic determinant problem. Define

$$A_G = \begin{cases} 0 & \text{if } (i,j) \notin E \\ x_{i,j} & \text{if } (i,j) \in E \end{cases}$$

- It is easy to observe that G has a perfect matching iff the $det(A_G)$ is not the zero polynomial.

The complexity of symbolic determinant

- As a polynomial, $det(A)$ could have $n!$ terms and hence just writing out $det(A)$ is not feasible for large n.
- But since $det(A)$ is a degree n polynomial in the x_{ij}, we can invoke the Schwartz-Zipple lemma using say a set S of scalars with $|S| \geq 2n$.
- Then assuming $det(A)$ is not the zero polynomial,

$$\mathbb{P}_{s \text{ uniform random in } S^{n^2}} \left[det(A(s)) = 0 \right] \leq \frac{1}{2}$$

- Note that $det(A(s))$ can be computed as fast as matrix product and can be efficiently computed in parallel.
- The symbolic determinant problem is one main example of a decision problem that can be computed efficiently with randomization but (currently) not known to be in P.

Randomized rounding – The weighted Max-Sat problem

The weighted Max-Sat problem

- Given a CNF formula $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ over a set of variables $x_1, \ldots, x_n$ with clause C_i having weight W_i.
- In contrast to Max-k-Sat and Exact Max-k-Sat, each clause can have any number of literals.
- **Goal:** is to find a truth assignment that maximizes that the total weight of the satisfied clauses.