CSC2420 - Algorithm Design, Analysis and Theory **Fall 2022**

Lecture 9: Vector Program and Random Walk

Lecturer: Allan Borodin *Scribe: Kevin Gao*

9.1 SDP and Vector Program for Max-2-SAT

We consider the formulation of max-2-SAT as a **semidefinite programming** (SDP) problem.

We introduce the variables y_i where

$$\begin{array}{ll} y_i = \pm 1 & \text{for } i = 0, \ldots, m \\ y_i = y_0 & \text{iff } x_i \text{ is true} \end{array}$$

To define the objective function, we want each clause C to have a value 1 if the clause is satisfied and 0 otherwise:

$$\begin{aligned} v(x_i) &= \frac{1 + y_i y_0}{2} \\ v(\neg x_i) &= \frac{1 - y_i y_0}{2} \\ v(x_i \vee x_j) &= 1 - v(\neg x_i) v(\neg x_j) = \frac{1 + y_0 y_i}{4} + \frac{1 + y_0 y_j}{4} + \frac{1 - y_i y_j}{4} \\ v(\neg x_i \vee x_j) &= \frac{3 - y_i y_0 + y_j y_0 + y_i y_j}{4} \\ v(\neg x_i \vee \neg x_j) &= \frac{3 - y_i y_0 - y_j y_0 - y_i y_j}{4} \end{aligned}$$

We can then define the objective as

$$\begin{array}{lll} \max & \sum_k w_k v(C_k) & \\ \text{s.t.} & y_i^2 = 1 & \forall i \in \{0, \ldots, m\} \end{array}$$

By collecting like terms, the objective can be expressed equivalently as

$$\max \sum_{0 \le i < j \le n} a_{ij}(1 + y_i y_j) + \sum_{0 \le i < j \le n} b_{ij}(1 - y_i y_j)$$

for some appropriate a_{ij} and b_{ij}.

We now relax the quadratic program to a vector program where each y_i is now a unit length vector $\mathbf{v}_i$ in $\mathbb{R}^{n+1}$ and scalar multiplication is replaced by vector dot product. This vector program can be approximated in polynomial time.

$$\max \sum_{0 \le i,j \le n} a_{ij}(1 + \mathbf{v}_i \cdot \mathbf{v}_j) + b_{ij}(1 - \mathbf{v}_i \cdot \mathbf{v}_j) \qquad \mathbf{v}_i \in \mathbb{R}^{n+1}, \; \|\mathbf{v}_i\| = 1$$

The randomized rounding proceeds by choosing a random hyperplane in $\mathbb{R}^{n+1}$ and then setting $y_i = 1$ if and only if $\mathbf{v}_i^*$ is on the same side of the hyperplane as $\mathbf{v}_0^*$. That is, let $\mathbf{r}$ is a uniform random vector in $\mathbb{R}^{n+1}$, and then set $y_1 = 1$ if and only if $\mathbf{r} \cdot \mathbf{v}_i^* \geq 0$.

Theorem 9.1. *The vector program relaxation is a 0.8785-approximation for Max-2-SAT.*

Proof: The expected weight produced by the randomly rounded vector program is

$$\mathbb{E}[W] = \sum_{0 \le i,j \le n} a_{ij}(1 + \mathbb{E}[y_i \cdot y_j]) + b_{ij}(1 - \mathbb{E}[y_i \cdot y_j])$$

where

$$\mathbb{E}[y_i \cdot y_j] = \Pr[y_i = y_j] \cdot 1 + \Pr[y_i \neq y_j] \cdot (-1)$$

We claim that

$$\Pr[y_i \neq y_j] = \frac{\theta_{ij}}{\pi}$$

Let $\mathbf{r}$ be a random vector and $\mathbf{s}$ be the projection of $\mathbf{r}$ onto the plane containing $\mathbf{v}_i$ and $\mathbf{v}_j$. Then,

$$\begin{aligned}\mathbf{v}_i \cdot \mathbf{r} &= \mathbf{v}_i \cdot (\mathbf{s} + \mathbf{r} - \mathbf{s}) \\ &= \mathbf{v}_i \cdot \mathbf{s} + \mathbf{v}_i \cdot (\mathbf{r} - \mathbf{s}) \\ &= \mathbf{v}_i \cdot \mathbf{s}\end{aligned}$$

Note that the second term vanishes because $\mathbf{r} - \mathbf{s}$ is orthogonal to $\mathbf{v}_i$. By the same argument, $\mathbf{v}_j \cdot \mathbf{r} = \mathbf{v}_i \cdot \mathbf{s}$. Now, y_i and y_j are set to different values iff $\mathbf{v}_i$ and $\mathbf{v}_j$ are separated by the hyperplane defined by $\mathbf{r}$. Hence, it suffices to consider the probability $\Pr[\mathbf{v}_i \text{ and } \mathbf{v}_j \text{ are separated}]$.

Without loss of generality, suppose that $\mathbf{v}_i$ has angular coordinate 0 and $\mathbf{v}_j$ has angular coordinate θ_{ij}. Further, let ϕ denote the angular coordinate of $\mathbf{s}$.

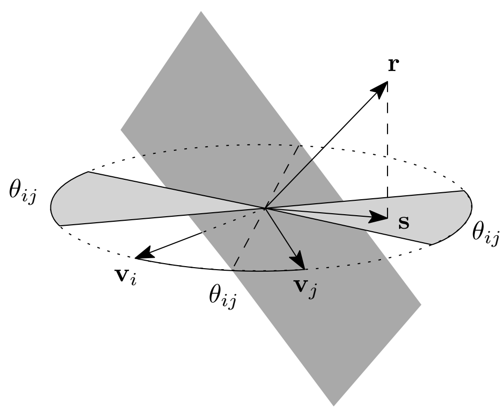

Figure 9.1: A random hyperplane defined by $\mathbf{r}$ separating $\mathbf{v}_j$ and $\mathbf{v}_i$.

$\mathbf{v}_i$ and $\mathbf{v}_j$ are separated by the hyperplane whenever $\mathbf{s}$ is within the lightly shaded region. There are two such regions, each with angular diameter θ_{ij}. Since $\mathbf{r}$ is uniformly chosen from $\mathbb{R}^{n+1}$, its projection $\mathbf{s}$ is uniformly distributed over $[0, 2\pi)$. There are two slices of diameter θ_{ij} that $\mathbf{s}$ can be in so that the hyperplane separate $\mathbf{v}_i$ and $\mathbf{v}_j$. Hence,

$$\Pr[\mathbf{v}_i \text{ and } \mathbf{v}_j \text{ are separated}] = \frac{2 \cdot \theta_{ij}}{2\pi} = \frac{\theta_{ij}}{\pi}.$$

It follows that $\Pr[y_i = y_j] = \frac{\theta_{ij}}{\pi} = 1 - \frac{\theta_{ij}}{\pi}$.

We define

$$\alpha = \frac{2}{\pi} \min_{0 \le \theta \le \pi} \frac{\theta}{1 - \cos\theta} \approx 0.87856$$

Observe that from calculus, we have for $0 \leq \theta \leq \pi$,

$$\frac{\theta}{\pi} \geq \alpha \left(\frac{1 - \cos\theta}{2} \right)$$

and similarly, for $0 \leq \theta \leq \pi$, $1 - \frac{\theta}{\pi} \geq \alpha \left(\frac{1+\cos\theta}{2} \right)$. Then,

$$\begin{aligned}
\mathbb{E}[W] &= \sum_{0 \leq i,j \leq n} a_{ij}(1 + \mathbb{E}[y_i \cdot y_j]) + b_{ij}(1 - \mathbb{E}[y_i \cdot y_j]) \\
&= \sum_{0 \leq i,j \leq n} a_{ij} \left(2 - \frac{2\theta_{ij}}{\pi} \right) + b_{ij} \cdot \frac{2\theta_{ij}}{\pi} \\
&= \sum_{0 \leq i,j \leq n} 2a_{ij} \left(1 - \frac{\theta_{ij}}{\pi} \right) + 2b_{ij} \cdot \frac{\theta_{ij}}{\pi} \\
&\geq \sum_{0 \leq i,j \leq n} 2a_{ij} \left(\frac{1 + \cos\theta_{ij}}{2} \right) + 2b_{ij} \cdot \left(\frac{1 - \cos\theta_{ij}}{2} \right)
\end{aligned}$$

Finally, since $\mathbf{v}_i \cdot \mathbf{v}_j = \|\mathbf{v}_i\| \cdot \|\mathbf{v}_j\| \cdot \cos\theta_{ij}$ and $\|\mathbf{v}_i\| = \|\mathbf{v}_j\| = 1$,

$$\begin{aligned}
\mathbb{E}[W] &\geq \sum_{0 \leq i,j \leq n} 2a_{ij} \left(\frac{1 + \cos\theta_{ij}}{2} \right) + 2b_{ij} \cdot \left(\frac{1 - \cos\theta_{ij}}{2} \right) \\
&= \sum_{0 \leq i,j \leq n} 2a_{ij} \left(\frac{1 + \mathbf{v}_i \cdot \mathbf{v}_j}{2} \right) + 2b_{ij} \cdot \left(\frac{1 - \mathbf{v}_i \cdot \mathbf{v}_j}{2} \right) \\
&= \sum_{0 \leq i,j \leq n} a_{ij} \left(1 + \mathbf{v}_i \cdot \mathbf{v}_j \right) + b_{ij} \cdot \left(1 - \mathbf{v}_i \cdot \mathbf{v}_j \right) \\
&= \alpha \cdot OPT_{\text{VP}} \\
&\geq \alpha \cdot OPT_{\text{SDP}}
\end{aligned}$$

■

9.2 SAT

9.2.1 Polynomial-time Algorithm for 2-SAT

Consider a 2CNF formula F in n variables. We consider each clause $x \vee y$ as an implication $\neg x \implies y$ and $\neg y \implies x$. We then construct a directed graph on $2n$ nodes corresponding to the $2n$ literals and edges corresponding to the implications above. For example, $x \implies y$ induces an edge (x, y). We claim that F is satisfiable iff there does not exist a variable x such that there is a directed path from x to $\neg x$ and a path $\neg x$ to x.

Clearly, k-SAT (for CNF formula with n variables, m clauses, and at most k literals/clauses) cna be trivially decided in time $\widetilde{O}(2^n m)$. $\widetilde{O}$ is the soft-O notation, in which we drops lower order terms (including multiplicative log factors).

9.2.2 3-SAT

Consider a 3-CNF formula with n variables and m clauses. While there are any clauses with 3 literals (say, x, y, z), branch on each of the 7 possible settings for (x, y, z) that can make the clause true (8 possible truth assignments, excluding the one that falsifies the clause).

On any branch, the current truth value setting can satisfy some clauses which can then be eliminated. In other clauses involving some variable in the current truth assignment, one or two variables can be eliminated. We repeat this until there are no clauses remaining with 3 literals. If any branch satisfies all clauses or if we are left with only consistent unit clauses, the given formula is satisfiable. If we are left with two contradictory unit clauses, the given formula is unsatisfiable.

The depth of the tree is at most $n/3$ since we are eliminating 3 distinct variables at each level. Hence, the time complexity is

$$O(7^{n/3} poly(m)) = 2^{\log_2 7n/3} poly(m) \approx 2^{2.81n/3} poly(m)$$

This branch-and-bound algorithm is a slight improvement on the naive algorithm which tries all 2^n possible truth assignments.

9.3 Random Walk for 2-SAT

The randomized algorithm for 2-SAT is based on arandom walk on the line graph with nodes $\{0, 1, \ldots, n\}$. Being on node i can be interpreted as havinga truth assignment τ that is Hamming distance i from some fixed satisfying assignment τ^* if F is satisfiable.

Start with an arbitrary truth assignment τ and if $F(\tau)$ is true then we are done; else find an arbitrary unsatisfied clause C and randomly choose one of the two variables x_i occurring in C and then change τ to τ' by setting $\tau'(x_i) = 1 - \tau(x_i)$.

Note that when we randomly select one of the two literals in C and flip it, we are getting closer to τ^* with probability at least 1/2. The distance to node 0 in a random walk on the line where on each random step, the distance to node 0 is reduced by 1 with at least 1/2 probability and otherwise increased by 1.

Theorem 9.2. *The expected time to hit node 0 is at most* $2n^2$.

9.3.1 Markov Chain

A **finite Markov chain** M is a discrete-time random proces defined over a set of **states** S and a matrix $\mathbf{P} = \{P_{ij}\}$ of **transition probabilities**.

Denote by X_t the state of the Markov chain at time t. It is a memoryless process since the future behavior of a Markov chain *depends only on its current state*:

$$\Pr[X_{t+1} = j \mid X_t = i] = P_{ij}$$

and hence

$$\Pr[X_{t+1} = j] = \sum_i \Pr[X_{t+1} = j \mid X_t = i] \Pr[X_t = i]$$

Given an initial state i, denote by r_{ij}^t the probability that the first time the process reaches state j occurs at time t.

$$r_{ij}^t = \Pr[X_t = j \wedge X_s \neq j,\, 1 \leq s \leq t-1 \mid X_0 = i]$$

Let f_{ij} be the probability that state j is reachable from initial state j.

$$f_{ij} = \sum_{t>0} r_{ij}^t$$

Let h_{ij} denote the expected number of steps to reach state j starting from state i (**hitting time**).

$$h_{ij} = \sum_{t>0} t \cdot r_{ij}^t$$

Finally, the **commute time** c_{ij} is the expected number of steps to reach state j sstarting from state i, and then return from i to j. That is,

$$c_{ij} = h_{ij} + h_{ji}.$$

The underlying directed graph of a Markov chain is a graph where each vertex in the graph corresponds to a state of the Markov chain and there is a directed edge from vertex i to vertex j iff $P_{ij} > 0$. A Markov chain is **irreducible** if its underlying graph *consists of a single strongly connected component.* We have the following theorem.

Theorem 9.3 (Fundamental Theorem of Markov Chains)**.** *For any finite, irreducible, and aperiodic Markov chain, there exists a unique stationary distribution π and for all states i, $h_{ii} < \infty$ and $h_{ii} = 1/\pi_i$.*

9.3.2 Random Walk on Graphs

We can now use our knowledge of Markov chains to analyze the expected time of the random walk algorithm for 2-SAT.

Let $G = (V, E)$ be a connected, non-bipartite, undirected graph with $|V| = n$ and $|E| = m$. A uniform random walk induces a Markov chain M_G where

- the states of M_G are the vertices
- for $u, v \in V$, $P_{uv} = 1/\deg(u)$ if $(u, v) \in E$ and $P_{uv} = 0$ if $(u, v) \notin E$

And

$$\left(\frac{\deg(v_1)}{2m}, \ldots, \frac{\deg(v_n)}{2m}\right)$$

is a stationary distribution of M_G.

Define $\mathbf{q}^t = [q_1^t \; q_2^t \; \ldots \; q_n^t]$ as the state probability vector, which is a row vector whose i-th component is the probability that the Markov chain is in state i at time t. A distribution π is a stationary distribution for a Markov chain with transition matrix $\mathbf{P}$ if $\pi = \pi\mathbf{P}$.

Let $C_u(G)$ be the expected time to visit every vertex, starting from u. Also, $C(G) = \max_u C_u(G)$ to be the cover time for G.

Theorem 9.4 (Aleliunas et al, 1979)**.** *Let G be a connected undirected graph. Then, for each edge (u, v), $C_{u,v} \leq 2m$. And $C(G) \leq 2m(n-1)$.*

It follows from this theorem that the simple random walk algorithm for 2-SAT has expected time at most $2n^2$.

9.3.3 Schoning's k-SAT Random Walk

choose a random assignment τ
repeat $3n$ times
    if τ satisfies F
        stop and accept
    else
        C = arbitrary unsatisfied clause
        randomly pick and flip one of the literals in C

Theorem 9.5. *If F is satisfiable, then the above succeeds with probability $p \geq [\frac{1}{2}\frac{k}{k-1}]^n$. It follows that if we repeat the above for t trials, then the probability that we fail to find a satisfying assignment is at most $(1-p)^t < e^{-pt}$. Setting $t = c/p$ for some constant c, we obtain error probability $(1/e)^c$.*

9.4 ETH and SETH

Conjecture 9.6 (Exponential Time Hypothesis)**.** *There exists some δ such that 3-SAT is not decidable in time $O(2^{(1+\delta)^n})$.*

Conjecture 9.7 (Strong Exponential Time Hypothesis)**.** *For all $\gamma < 1$, SAT is not decidable in time $O(2^{\gamma^n})$.*