CS294 Classical and Quantum Optimization	**Spring 2010**

Lecture 8: February 17, 2010

Lecturer: Profs. Satish Rao and Umesh Vazirani *Scribe: Kevin Dick*

8.1 Matrix Multiplicative Updates to Solve SDPs

We continue our discussion on using the matrix multiplicative update rule to solve SDPs. The matrix multiplicative update game features two players, A and B. In each round, A plays a density matrix ρ, and B plays a measurement matrix M such that $0 \preceq M \preceq I$. The loss to A at round t is given by $\rho_t \cdot M_t$.

A	**B**
ρ_1	M_1
ρ_2	M_2
$\vdots$	$\vdots$
ρ_T	M_T

The adaptive cost of A's play is $\sum_{j=1}^T \rho_j \cdot M_j$, as compared to the optimum of $\lambda_{\min}(\sum_{j=1}^T M_j) = \min_{|x\rangle} \langle x| \sum_{j=1}^T M_j |x\rangle$.

Theorem 8.1 *A can play such that the adaptive cost satisfies* $\sum_{j=1}^T \rho_j \cdot M_j \leq (1+\epsilon/2)\lambda_{min}(\sum_{j=1}^T M_j) + \ln n/\epsilon$.

Proof: A's strategy to achieve this bound is to define, at each round t, a weight matrix $W^{(t)}$ and a corresponding density matrix $\rho^{(t)}$. The first round has $W^{(1)} = I$ and $\rho^{(1)} = W^{(1)}/\mathrm{Tr}(W^{(1)}) = I/n$, with

$$W^{(t+1)} = (1-\epsilon)^{\sum_{j=1}^t M^{(j)}}, \quad \rho^{(t+1)} = \frac{W^{(t+1)}}{\mathrm{Tr}(W^{t+1})}.$$

Consider now the SDP $\sum y_i A_i - B \succeq 0$, with each A_i and B n by n matrices. The goal is to find y_i such that for all $|x\rangle$, $\langle x| \sum y_i A_i - B|x\rangle \geq 0$, or equivalently such that for all $X \succeq 0, X \cdot (\sum y_i A_i - B) \geq 0$. Let $\mathcal{L}_y = \sum y_i A_i - B$, and suppose $||L_y||_2 \leq \alpha$. Define $M_y = \frac{\mathcal{L}_y + \alpha I}{2\alpha}$, so that $0 \preceq M_y \preceq I$.

An analogy with the LP example we've seen earlier is finding x such that $f_1(x) - b_1 \leq 0, f_2(x) - b_2 \leq 0, \ldots, f_n(x) - b_n \leq 0$. Pick $w_1, w_2, \ldots, w_n$. Then an oracle takes $w_1, w_2, \ldots, w_n$ and gives an x such that $\sum_i w_i f_i(x) - b_i \leq 0$.

For our SDP problem, we have an oracle which, given X, produces y such that $\mathcal{L}_y \cdot X \geq \delta(I \cdot X) = \delta \mathrm{Tr}(X)$ (recall that $X \cdot (\sum y_i A_i - B) \geq 0$ is equivalent to $\sum y_i X \cdot A_i - B \cdot X \geq 0$). Update X by the matrix multiplicative update rule. Our loss $\sum_{t=1}^T X^{(t)} \cdot M^{(t)} = \sum_{t=1}^T X^{(t)} \cdot (\frac{\mathcal{L}_y + \alpha I}{2\alpha})$ is at least $(\frac{\delta}{2\alpha} + \frac{1}{2})T$, which implies

$$(1+\frac{\epsilon}{2})\lambda_{\min}(\sum_{t=1}^T (\frac{\mathcal{L}^{(t)}}{2\alpha} + \frac{I}{2})) + \frac{\ln n}{\epsilon} \geq (\frac{\delta}{2\alpha} + \frac{1}{2})T.$$

We want to show that $\frac{1}{T}\sum^T \mathcal{L}_y = \frac{1}{T}\sum y_i A_i - B \succeq 0$ for sufficiently large T. We have that

$$\begin{aligned}
(1+\frac{\epsilon}{2})\lambda_{\min}(\sum_{t=1}^{T}\frac{\mathcal{L}^{(t)}}{2\alpha}) + \frac{T}{2}(1+\frac{\epsilon}{2}) + \frac{\ln n}{\epsilon} &\geq \frac{\delta T}{2\alpha} + \frac{T}{2} \\
\implies (1+\frac{\epsilon}{2})\lambda_{\min}(\sum_{t=1}^{T}\frac{\mathcal{L}^{(t)}}{2\alpha}) &\geq T(\frac{\delta}{2\alpha} - \frac{\epsilon}{4}) - \frac{\ln n}{\epsilon} \\
\implies \lambda_{\min}(\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}^{(t)}) &\geq \frac{2\alpha}{1+\epsilon/2}((\frac{\delta}{2\alpha} - \frac{\epsilon}{4}) - \frac{\ln n}{\epsilon T}).
\end{aligned}$$

We need to choose T to be large enough that the right-hand side is at least zero. $T = \frac{8\alpha^2 \ln n}{\delta^2}$ with $\epsilon \leq \frac{\delta}{2\alpha}$ is sufficient.

■

8.2 An application

We'll now see an application of the matrix multiplicative update rule. Consider the SDP of choosing X to maximize $C \cdot X$ subject to $A \cdot X \leq b$ and $X \succeq 0$. It's dual is choosing y to minimize by subject to $\sum A_j y_j \succeq C$ and $y_j \geq 0$. Set $A_1 = I$, $b_1 = R$, and $\mathrm{Tr}(X) = R$. Let ρ be such that $||\sum A_j y_j - C||_2 \leq \rho$, and set $\alpha = C \cdot X$. Then $\overline{y} = \frac{1}{T}\sum y^{(t)} + \frac{\delta\alpha}{R} e_1$ is a feasible objective, with $T = \theta(\frac{\rho^2 R^2 \ln(n)}{\delta^2 \alpha^2})$. Our oracle takes as input the matrix X, and returns a y such that $by \leq \alpha$ and $(\sum A_i y_i - C) \cdot X \geq 0$.

We'll use this framework to solve the max-cut problem. Given a graph, we wish to partition the vertices to maximize $\sum_e ||v_i - v_j||^2$, where the summation is taken over edges crossing the partition. Clearly $||v_i||^2 \leq 1$ for all i. If X is the n by n matrix of dot products, we can write this as maximizing $C \cdot X$, with $X_{ii} \leq 1$ for all i. Here $C = dI - A$, with A the adjacency matrix of our graph.

The dual of this program is to minimize $\sum y_i$ with $\mathrm{diag}(y_i) \succeq C$ and $y_i \geq 0$ for all i. Intuitively, one way to solve this is the following procedure:

(1) Push apart the vectors by moving them away from their neighbors.
(2) Push back onto the unit sphere.
(3) Repeat.

We work with graphs of degree d, and $nd \leq \alpha \leq 3nd$ with $0 \preceq C \preceq 2dI$. Our update rule sets $M^{(t)} = (\sum_i A_i y_i - C + \rho I)/2\rho$, $W^{(t)} = e^{-\epsilon \sum_t M^{(t)}}$, and $X = \frac{RW^{(t)}}{\mathrm{Tr}(W^{(t)})}$. Observe that $\mathrm{Tr}(X) = n$ and $y_i = O(\frac{\alpha}{n}) = O(d)$.

Returning to the SDP, we are given X and desire y_i such that $\sum y_i \leq \alpha$. We get $X \cdot \mathrm{diag}(y_i) - C \cdot X \geq 0 \implies \sum y_i X_{ii} - C \cdot X \geq 0$, and $||\mathrm{diag}(y) - C|| \leq O(d) = \rho$. Note the expansion $\exp(-\epsilon \sum M^{(t)}) \approx \sum_k \frac{1}{k!} \sum \prod_{\sigma \in S_k^{1 \ldots t}} \epsilon M^{\sigma(i)}$.

To build an oracle, we have two cases. The first is $C \cdot X < \alpha$. Then $y_i = \frac{\alpha}{n}$, and $M^{(t)} = (\mathrm{diag}(\frac{\alpha}{n}) - C + \rho I)/2\rho$. The second is $C \cdot X \geq \alpha$. Then $C \cdot X = \lambda\alpha$, with $\lambda \geq 1$.