Approximating Quadratic Programs with Positive Semidefinite Constraints

Elad Hazan* Satyen Kale†

Abstract

We describe a polynomial time approximation algorithm to the problem of maximizing a quadratic form subject to quadratic constraints specified by PSD matrices. A special case, that has applications for clustering [CW04], is optimizing quadratic forms over the unit cube.

Approximation algorithms with similar guarantees are known [Nes98, NRT99, Meg01, CW04], and there is evidence that this factor is optimal [ABH+]. The following analysis is particularly simple.

Consider the following quadratic program:

$$\begin{array}{rl} & \max\ x^T A x \\ i = 1, \ldots, m & x^T A_i x \leq 1 \end{array} \tag{1}$$

where, $x \in \mathbb{R}^n$, and $A_1, A_2, \ldots, A_m$ are positive semidefinite matrices in $\mathbb{R}^{n \times n}$. We consider the following semidefinite programming relaxation of the problem:

$$\begin{array}{rl} & \max\ A \bullet X \\ i = 1, \ldots, m & A_i \bullet X \leq 1 \\ & X \succeq 0 \end{array} \tag{2}$$

Note that the trivial solution $x = \mathbf{0}$ shows that the optimum of (1) is non-negative. We will assume that the optimum is positive henceforth. We give the following poly-time randomized algorithm to approximate (1) up to a factor of $O(\log(mn))$:

Procedure ApproximateQP

1. Solve the SDP relaxation (2) to get optimal solution X.
2. Compute a matrix V such that $X = V^T V$.
3. Choose a unit vector $u \in \mathbb{R}^n$ uniformly at random.
4. Compute $x = \left[\sqrt{\frac{n}{8\ln(mn)}}\right] Vu$.
5. **if** $\exists i$ s.t. $x^T A_i x > 1$ **or** $x^T A x < \frac{1}{16\ln(mn)} A \bullet X$ **then**
   abort
   else
   return x as a solution to (1).
   end

The second step in the algorithm, namely computing the decomposition $X = V^T V$, can be performed by computing the Cholesky decomposition of X or by finding the square root of X.

Clearly, if the algorithm does not abort, it returns an $O(\log(mn))$ approximate solution to (1). We will show that the algorithm succeeds with non-negligible probability:

*Computer Science Department, Princeton University. ehazan@cs.princeton.edu.
†Computer Science Department, Princeton University. satyen@cs.princeton.edu.

Theorem 1 *The algorithm succeeds with probability at least* $1/4n$*.*

So we can run the algorithm $O(n)$ times to get a constant probability of success. Before we prove Theorem 1, we need the following well-known lemma on the Gaussian nature of projections (see [ARV04]):

Lemma 1 *For any vector* $v \in \mathbb{R}^n$ *and a unit vector* $u \in \mathbb{R}^n$ *chosen uniformly at random, we have*

$$\mathbf{E}[(v^T u)^2] \;=\; \frac{||v||^2}{n} \quad \textit{and} \quad \mathbf{Pr}\left[(v^T u)^2 \;>\; \frac{t||v||^2}{n}\right] \;\le\; e^{-t/4}$$

Using this lemma, we can prove that the first condition of failure of line 5 of the algorithm happens with low probability:

Lemma 2 *For any* i*,* $\mathbf{Pr}[x^T A_i x > 1] < 1/m^2 n$*.*

Proof: For convenience, we will work with $y = Vu$ instead of x. Since A_i is positive semidefinite, we can decompose it as $A = \sum_k a_k a_k^T$ for vectors $a_1, a_2, \ldots, a_n \in \mathbb{R}^n$. Then

$$y^T A_i y \;=\; \sum_k y^T a_k a_k^T y \;=\; \sum_k u^T V^T a_k a_k^T V u \;=\; \sum_k (a_k^T V u)^2$$

Invoking Lemma 1, we conclude that for any k,

$$\mathbf{Pr}\left[(a_k^T V u)^2 > \frac{8\ln(mn)||V^T a_k||^2}{n}\right] < \frac{1}{m^2 n^2}$$

and so by union bound on all k,

$$\mathbf{Pr}\left[\sum_k (a_k^T V u)^2 > \frac{8\ln(mn)\sum_k ||V^T a_k||^2}{n}\right] < \frac{1}{m^2 n}$$

Now $\sum_k ||V^T a_k||^2 = \sum_k a_k^T V^T V a_k = \sum_k a_k a_k^T \bullet X = A_i \bullet X \le 1$. Since $x = \left[\sqrt{\frac{n}{8\ln(mn)}}\right] y$, we get the required bound. □

Now we bound the probability of failure from the second condition of line 5 of the algorithm:

Lemma 3 $\mathbf{Pr}[x^T A x < \frac{1}{16\ln(mn)} A \bullet X] < 1 - 1/2n$*.*

Proof: As before, it is more convenient to work with $y = Vu$ rather than x. We will show the equivalent bound $\mathbf{Pr}[y^T A y < A \bullet X/2n] < 1 - 1/2n$. First, we note that $\mathbf{E}[y^T A y] = A \bullet X/n$: by Lemma 1,

$$\mathbf{E}[y_k y_\ell] \;=\; \frac{1}{2}\mathbf{E}[y_k^2 + y_\ell^2 - (y_k - y_\ell)^2] \;=\; \frac{1}{2}\left[\frac{||v_k||^2}{n} + \frac{||v_\ell||^2}{n} - \frac{||v_k - v_\ell||^2}{n}\right] \;=\; \frac{v_k^T v_\ell}{n}$$

Using the facts $A \bullet X = \sum_{k\ell} A_{k\ell} v_k^T v_\ell$; $y^T A y = \sum_{k\ell} A_{k\ell} y_k y_\ell$ and linearity of expectation, the claim follows.

Next, we claim that for any direction u and the corresponding y, $y^T A y \le A \bullet X$. We prove this showing that $\tilde{X} = yy^T$ is a feasible solution for (2) and hence $y^T A y = A \bullet \tilde{X} \le A \bullet X$. For this, we show that $A_i \bullet \tilde{X} \le 1$ for all i. As in Lemma 2, let $A_i = \sum_k a_k a_k^T$, and then

$$A_i \bullet \tilde{X} \;=\; y^T A_i y \;= \sum_k (a_k^T V u)^2 \;\le\; \sum_k ||V^T a_k||^2 \;=\; A_i \bullet X \;\le\; 1$$

The first inequality uses the fact that u is a unit vector, and the last equality follows as in Lemma 2.

Let $t = \mathbf{Pr}[y^T Ay < A \bullet X/2n]$. Then we have the following averaging argument:

$$t \cdot \frac{A \bullet X}{2n} + (1-t) \bullet A \bullet X \;\geq\; \mathbf{E}[y^T Ay] \;=\; \frac{A \bullet X}{n}$$

Simplifying, we get $t \leq 1 - 1/(2n-1) < 1 - 1/2n$. □

PROOF:[Theorem 1]
Using lemmas (2), (3) and the union bound, we conclude that the probability of failure is bounded by (assuming $m \geq 4$)

$$m \cdot \frac{1}{m^2 n} + 1 - \frac{1}{2n} \;\leq\; 1 - \frac{1}{4n}$$

□

References

[ABH+] S. Arora, E. Berger, E. Hazan, G. Kindler, and M. Safra. On non-approximability for quadratic programs. *To appear in FOCS 2005.*

[ARV04] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *36th ACM STOC*, pages 222–231, 2004.

[CW04] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending grothendieck's inequality. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, pages 54–60, Washington, DC, USA, 2004. IEEE Computer Society.

[Meg01] A. Megretski. Relaxation of quadratic programs in operator theory and system analysis. *Systems, Approximation, Singular Integral Operators, and Related Topics (Bordeaux, 2000)*, (3):365–392, 2001.

[Nes98] Y. Nesterov. Global quadratic optimization via conic relaxation. *Working paper CORE*, 1998.

[NRT99] A. Nemirovski, C. Roos, and T. Terlaky. On maximization of quadratic form over intersection of ellipsoids with common center. *Mathematical Programming*, 86(3):463–473, 1999.