2-2018

Optimization of Quadratic Forms and t-norm Forms on Interval Domain and Computational Complexity

Milan Hladik
Charles University, Faculty of Mathematics and Physics, Department of Applied Mathematics, Prague, milan.hladik@matfyz.cz

Michal Černý
University of Economics, Prague, cernym@vse.cz

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Comments:
Technical Report: UTEP-CS-18-14
To appear in *Proceedings of the World Conference on Soft Computing*, Baku, Azerbaijan, May 29-31, 2018.

Optimization of Quadratic Forms and t-norm Forms on Interval Domain and Computational Complexity

Milan Hladík
Charles University
Faculty of Mathematics and Physics
Department of Applied Mathematics
Malostranské nám. 25
11800, Prague, Czech Republic
milan.hladik@matfyz.cz

Michal Černý
Department of Econometrics
University of Economics, Prague
W. CHurchill's sq. 4
13067 Prague, Czech Republic
cernym@vse.cz

Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, Texas 79968, USA
vladik@utep.edu

***Abstract*—We consider the problem of maximization of a quadratic form over a box. We identify the NP-hardness boundary for sparse quadratic forms: the problem is polynomially solvable for $O(\log n)$ nonzero entries, but it is NP-hard if the number of nonzero entries is of the order n^ε for an arbitrarily small $\varepsilon > 0$. Then we inspect further polymonially solvable cases. We define a sunflower graph over the quadratic form and study efficiently solvable cases according to the shape of this graph (e.g. the case with small sunflower leaves or the case with a restricted number of negative entries). Finally, we define a generalized quadratic form, called t-norm form, where the quadratic terms are replaced by t-norms. We prove that the optimization problem remains NP-hard with an *arbitrary* Lipschitz continuous t-norm.**

I. Introduction

In this paper we elaborate the problems outlined in [4] in more details. In that work we studied processing imprecise data from multiple sources which interact together. The interaction among inputs $x_1, \dots, x_n$ is formalized by a function $f(x_1, \dots, x_n)$ which cannot be written in a separable form as $\sum_{i=1}^n f_i(x_i)$ (for some functions f_i). An example is a quadratic form $x^T Ax$ with nonzero off-diagonal entries, which is studied in this paper. Then we consider a more general form of pairwise interactions: formally, we replace the bilinear terms $x_i x_j$ $(i \neq j)$ from $x^T Ax$ by so-called t-norms (which can be regarded as generalizations of the "AND" logical connective).

The general question is: when the inputs $x_1, \dots, x_n$ are imprecise but are known to be in given compact intervals

$$\boldsymbol{x_1} = [\underline{x}_1, \overline{x}_1],\ \dots,\ \boldsymbol{x_n} = [\underline{x}_n, \overline{x}_n],$$

and we are given a function $f : \mathbb{R}^n \to \mathbb{R}$, can we find tight bounds for $f(x_1, \dots, x_n)$? Formally, denoting

$$\boldsymbol{x} = \boldsymbol{x_1} \times \cdots \times \boldsymbol{x_n}, \tag{1}$$

the problem reduces to the computation of $\sup_{x\in\mathbb{R}^n}\{f(x) \mid x \in \boldsymbol{x}\}$ and $\inf_{x\in\mathbb{R}^n}\{f(x) \mid x \in \boldsymbol{x}\}$. Here, the expression "to find the bounds" refers to computational complexity: We are to determine under which conditions the bounds can be evaluated in polynomial time and when the computation is NP-hard.

Recall that in general, finding the tight bounds for a general function f need not be recursive. This is why various classes of functions of interest in data processing need to be studied separately.

In this text, bold symbols—such as $\boldsymbol{x}$—refer to n-dimensional intervals of the form (1). The real n-vectors of lower and upper bounds are denoted by $\underline{x}$ and $\overline{x}$, respectively, and we write $\boldsymbol{x} = [\underline{x}, \overline{x}]$ for short.

Basics in computations complexity and interval computing can be found e.g. in [6].

II. Quadratic forms on interval domain

Consider a quadratic form $f \colon \mathbb{R}^n \to \mathbb{R}$

$$\begin{aligned} f(x) &= b^T x + x^T Ax \\ &= \sum_{i=1}^n b_i x_i + \sum_{i,j=1}^n a_{ij} x_i x_j \end{aligned}$$

restricted to a given interval domain $\boldsymbol{x} = [\underline{x}, \overline{x}]$. It is known that computing the range of f on $\boldsymbol{x}$, i.e.,

$$\begin{aligned} \underline{f} &:= \min\ f(x) \ \text{ subject to } \ x \in \boldsymbol{x}, \\ \overline{f} &:= \max\ f(x) \ \text{ subject to } \ x \in \boldsymbol{x} \end{aligned}$$

is an NP-hard problem. This is true even for A positive definite, in which case computing $\underline{f}$ is polynomial whereas computing $\overline{f}$ is NP-hard.

Assumption. For simplicity of exposition, we focus only on computation of $\overline{f}$ in the sequel. We will also assume for the remainder of the paper that $f(x)$ is convex (i.e., that A is positive semidefinite).

III. Sparse quadratic forms

Suppose that the matrix A is sparse, that is, most of the off-diagonal entries are zero. Becomes the problem of computing the range tractable?

In this section it is sufficient to fix

$$\boldsymbol{x} = [0, 1]^n.$$

Proposition 1. *The problem of computing $\overline{f}$ remains NP-hard even when the number of off-diagonal non-zeros in A is bounded by $\mathcal{O}(n^{1/k})$.*

Proof: Let $f(x) = b^T x + x^T Ax$ be a quadratic function on $\mathbb{R}^n$. Consider the quadratic form

$$g(x, y) := f(x) + \sum_{i=1}^{m}(2y_i - 1)^2.$$

Then the maximum of $g(x, y)$ on $[0, 1]^{n+m}$ is the same as the maximum of $f(x)$, shifted by the amount of m. That is,

$$\overline{g} = \overline{f} + m. \tag{2}$$

Putting $m := n^{2k}$ we get that the quadratic form $g(x, y)$ of dimension $d = n+m$ has $\mathcal{O}(d^{1/k})$, non-zero off-diagonal entries in the corresponding matrix. Since $f(x)$ was an arbitrary quadratic form, computing the range of $g(x, y)$ is NP-hard, too. ∎

Corollary (to the proof). Under the assumption of Proposition 1, *it is NP-hard to approximate $\overline{f}$ with a given (arbitrarily large) absolute error*. This follows from the fact that the maximum of a quadratic form is known to be NP-hard to approximate with an absolute error [1], and (2) does not change the absolute error.

On the other hand, approximating $\overline{f}$ with a relative error can be done efficiently via semidefinite ralaxation even for a nonconvex $f(x)$, see [5].

Proposition 2. *The problem becomes polynomial if the number of off-diagonal non-zeros in A is bounded by $\mathcal{O}(\log n)$.*

Proof: Denote

$$I := \{i = 1, \dots, n \mid \exists j \neq i : a_{ij} \neq 0\}.$$

Now, $f(x)$ can be expressed as

$$f(x) = \sum_{i \notin I}(b_i x_i + a_{ii}x_i^2) + \sum_{i,j \in I} a_{ij}x_i x_j,$$

and its maximum as

$$\overline{f} = \max_{x \in \boldsymbol{x}} \left(\sum_{i \notin I}(b_i x_i + a_{ii}x_i^2) \right) + \max_{x \in \boldsymbol{x}} \left(\sum_{i,j \in I} a_{ij}x_i x_j \right). \tag{3}$$

The first term in (3) is computed easily as

$$\max_{x \in \boldsymbol{x}} \left(\sum_{i \notin I}(b_i x_i + a_{ii}x_i^2) \right) = \sum_{i \notin I} \max_{x_i \in \boldsymbol{x}_i} (b_i x_i + a_{ii}x_i^2),$$

and maximizing a univariate quadratic function is a trivial task.

The second term in (3) requires maximizing a quadratic function on an interval domain in dimension $\mathcal{O}(\log n)$. Hence, by brute force, we find the maximum [2] in exponential time w.r.t. $\mathcal{O}(\log n)$, which is polynomial w.r.t. n. ∎

IV. Polynomial cases based on sunflower graphs

Without loss of generality assume that A is upper triangular and that $\boldsymbol{x} = [0, 1]^n$. Consider the graph $G = (V, E)$, where $V = \{x_1, \dots, x_n\}$ and $\{x_i, x_j\}$ is an edge of G if and only if $a_{ij} \neq 0$. So we are in fact maximizing a quadratic form $f(x)$ on the graph G (see Chapter 10 of [3]).

Let $D \subseteq V$ be a vertex cut such that the graph $G' = (V \setminus D, E')$ after removing the cut D consists of connected components of vertex size $\mathcal{O}(\log n)$. Suppose further that the size of the cut is $|D| = \mathcal{O}(\log n)$. (A graph with such cut is sometimes called *sunflower graph*, see Figure 1.) Then the cut is associated with $|D|$ variables. Hence we can process all $0/1$-assignments of these variables. There are at most $2^{|D|}$ such assignments. For every such assignments, we resolve the problem by brute force in each of the components. Therefore, the overall time complexity is

$$\begin{aligned} &2^{|D|}(T_1 + T_2 + \dots + T_k) \\ &\quad \leq 2^{\mathcal{O}(\log n)}(2^{\mathcal{O}(\log n)} + 2^{\mathcal{O}(\log n)} + \dots + 2^{\mathcal{O}(\log n)}) \\ &\quad \leq \text{poly}(n), \end{aligned}$$

where T_i is time complexity of maximization over ith component, $k \leq n$ is the number of components, and poly(n) is a polynomial in n.

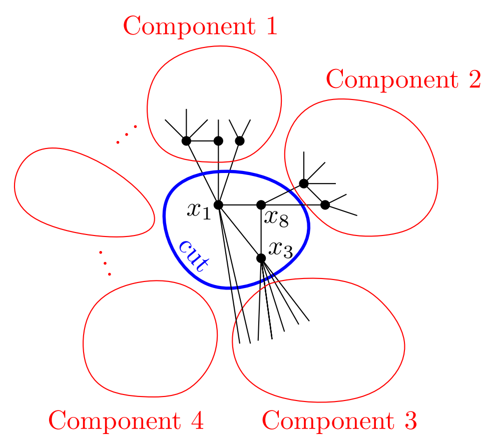

Fig. 1. A sunflower graph with a cut of size $\mathcal{O}(\log n)$ and components of size $\mathcal{O}(\log n)$.

A problem. How to find a suitable cut? This is an open challenging question. Notice that minimum cut splitting graph G into two components can be found efficiently by means of linear programming. Nevertheless, incorporating restrictions on size of the components seems a hard problem.

Special graphs

The above reasoning can be extended even to the components larger than $\mathcal{O}(\log n)$, but having a special structure. So, we will now discuss a few of special graphs possessing a suitable structure. For the sake of simplicity of exposition, we will illustrate it on the graph $G = (V, E)$.

Few negative coefficients: Provided that all coefficients are nonnegative, that is, $b_i \geq 0$ and $a_{ij} \geq 0$ for all $i, j \in \{1, \dots, n\}$, then the optimal solution is simply $x = (1, \dots, 1)^T$. If it is not the case, we can still effectively compute an optimal solution as long as the number of negative coefficients b_i and a_{ij} is small. Define a cut D to contain all variables incident with negative coefficients:

$$D := \{x_i;\, b_i < 0 \text{ or } a_{ij} < 0 \text{ for some } j\}.$$

If $|D| = \mathcal{O}(\log n)$, then we are done since applying the cut we obtain a subproblem with nonnegative coefficients and the 0/1-variables in D can be tested brute-force in time $2^{|D|} = poly(n)$.

Other special graphs

Assume now for simplicity in the remainder of this section that the domain of variables is $\boldsymbol{x} = [-1, 1]^n$. Further assume that $b_i = 0$ for every i.

Trees: If G is a tree, then maximizing the quadratic function on G is easy: Take an arbitrary vertex in $x_i \in G$ as a root, and distinguish two assignments $x_i = \pm 1$. For each assignment, the remaining variables associated with G have determined values. Sorting the vertices according to some tree search algorithm, we put $x_j := \operatorname{sgn}(a_{ij} x_i)$ when x_i precedes x_j.

Planar graphs: The above class can be extended to planar graphs with $\mathcal{O}(\log n)$ faces because by removing $\mathcal{O}(\log n)$ vertices we obtain a tree.

Bipartite graphs: Complete bipartite graphs $K_{m,n}$ and their subgraphs are also efficiently processed provided $a_{ij} \leq 0$ for $i \neq j$. The variables associated with the first set of vertices will be set as $x_i := 1$, and the others $x_i := -1$.

If the assumption $a_{ij} \leq 0$ for $i \neq j$ is not satisfied, then the bipartite graph is still efficiently processed as long as $m = \mathcal{O}(\log n)$, in which case the vertex cut D is the smaller of those two subsets.

Remark. For related results see [7].

V. T-NORM FORMS

Recall that a t-norm is a function $T\colon [0,1]^2 \to [0,1]$ satisfying:

- commutativity: $T(a,b) = T(b,a)$,
- monotonicity: $a \leq c,\ b \leq d \ \Rightarrow\ T(a,b) \leq T(c,d)$,
- associativity: $T(a,(T(b,c))) = T(T(a,b),c)$,
- 1 is identity element: $T(a,1) = a$.

From the definition, we immediately have

$$T(0,0) = T(0,1) = T(1,0) = 0, \quad T(1,1) = 1. \tag{4}$$

Given t-norms T_{ij}, the question is how easy is evaluation of the *t*-norm form

$$f_T(x) = \sum_{i=1}^{n} (b_i x_i + a_{ii} x_i^2) + \sum_{i \neq j} a_{ij} T_{ij}(x_i, x_j)$$

on a given interval domain $\boldsymbol{x}$.

Proposition 3. *Maximizing a t-norm form on $\boldsymbol{x} = [0,1]^n$ is NP-hard even if we choose and fix for every T_{ij} a Lipschitz continuous t-norm, that is,*

$$|T_{ij}(x) - T_{ij}(x')| \leq \alpha \cdot \|x - x'\|,$$

where α is a Lipschitz constant and $\|\cdot\|$ is any vector norm.

Proof: Let $f(x) = b^T x + x^T A x$ be a convex quadratic function on $\mathbb{R}^n$. Consider the t-norm form

$$f_T(x) := b^T x + \sum_{i=1}^{n} a_{ii} x_i^2 + \beta \sum_{i=1}^{n} (2x_i - 1)^2 + \sum_{i \neq j} a_{ij} T_{ij}(x_i, x_j).$$

By the Lipschitz continuity assumption, for sufficiently large β the function $f_T(x)$ is convex. Thus the maximum of $f_T(x)$ is attained in a vertex of $\boldsymbol{x}$. However, on a set of vertices $x \in \{0,1\}^n$,

$$f_T(x) = f(x) + \beta n$$

since $T_{ij}(x_i, x_j) = x_i x_j$ and $(2x_i - 1)^2 = 1$. This means that the maximum of $f_T(x)$ is the same as the maximum of $f(x)$, shifted by the amount of βn. Since maximizing $f(x)$ on $\boldsymbol{x}$ is NP-hard, maximizing t-norm forms on $\boldsymbol{x}$ is NP-hard, too. ∎

Remark 1. It is interesting that the proof does not require all the axioms of a *t*-norm. Basically, we used (4) only. Thus the statement holds true for any Lipschitz continuous functions T_{ij} satisfying (4).

Notice that the commonly used t-norms satisfy the assumption of the proposition:

- product t-norm $T(x,y) = xy$ (in this case, the t-norm form is a quadratic form),
- minimum t-norm $T(x,y) = \min\{x,y\}$,
- Łukasiewicz t-norm
$$T(x,y) = \max\{0, x + y - 1\},$$
- nilpotent minimum t-norm
$$T(x,y) = \begin{cases} \min\{x,y\} & \text{if } x + y > 1, \\ 0 & \text{otherwise,} \end{cases}$$
- Hamacher product t-norm
$$T(x,y) = \begin{cases} 0 & \text{if } x = y = 0, \\ \frac{xy}{x+y-xy} & \text{otherwise.} \end{cases}$$

On the other hand, the drastic t-norm defined as

$$T(x,y) = \begin{cases} \min\{x,y\} & \text{if } \max\{x,y\} = 1, \\ 0 & \text{otherwise} \end{cases}$$

does not satisfy the assumption.

ACKNOWLEDGMENTS

M. Hladík was supported by the Czech Science Foundation Grant P403-18-04735S. M. Černý was supported by the Czech Science Foundation Grant P402/12/G097. V. Kreinovich was supported in part by the National Science Foundation Grant HRD-1242122 (Cyber-ShARE Center of Excellence).

References

[1] M. Černý M and M. Hladík M. The complexity of computation and approximation of the t-ratio over one-dimensional interval data. *Computational Statistics & Data Analysis* **80**, 2014, 26–43.

[2] J.-A. Ferrez, K. Fukuda, and T. Liebling. Solving the fixed rank convex quadratic maximization in binary variables by a parallel zonotope construction algorithm. *European Journal of Operational Research* **166**, 2005, 35–50.

[3] B. Gärtner and J. Matoušek. *Approximation Algorithms and Semidefinite Programming.* Springer, 2012.

[4] M. Hladík, M. Černý, and V. Kreinovich, "When Is Data Processing under Interval and Fuzzy Uncertainty Feasible: What If Few Inputs Interact? Does Feasibility Depend on How We Describe Interaction?" *These Proceedings.*

[5] Nesterov Yu. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods & Software* **9** (1–3), 1998, 141–160.

[6] V. Kreinovich, A. Lakayev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.

[7] Yu G.-D. (2014). Quadratic forms on graphs with applications to minimizing the least eigenvalue of signless Laplacian over bicyclic graphs. *Electronic Journal of Linear Algebra* **27**, article 13.